Fair and Efficient Chore Allocation: Existence and Computation

Aniket Murhekar
University of Illinois at Urbana-Champaign
aniket2@illinois.edu

Abstract
We investigate the existence and computation of fair and efficient allocations of indivisible chores to agents with additive preferences. We consider the popular envy-based fairness notions of envy-freeness up to one chore (EF1) and the efficiency notion of Pareto-optimality (PO).

The existence of an allocation of chores that is simultaneously EF1 and PO is regarded a major open problem in discrete fair division. We show that an EF1 and PO allocation can be computed in polynomial time for certain structured instances. These results comprise the first non-trivial positive results for the problem and reveal insights towards settling the problem in its full generality.

1 Introduction
The question of fairly allocating a set of indivisible items to agents with additive preferences has been widely studied across various disciplines such as computer science, AI, economics, and social choice. Among various definitions of fairness, the central one is envy-freeness (EF) [Foley, 1967]. An allocation is EF if every agent prefers the items allocated to her at least as much as those allocated to any other agent. The simple example of allocating one indivisible task to agents. The single agent is strictly better off. While fairness is desirable, a fair allocation can have poor overall efficiency. Thus, obtaining allocations that achieve fairness in conjunction with efficiency is the ideal goal. Indeed, the existence of allocations that are both EF1 and PO is an important problem in discrete fair division. Note that merely checking if an allocation is PO is coNP-hard [de Keijzer et al., 2009], highlighting the challenge of this problem.

EF1 and PO for goods. For goods (items that provide value to agents), [Caragiannis et al., 2016] showed that an EF1 and PO allocation always exists: they showed that the allocation with the maximum Nash welfare, i.e., the geometric mean of the utilities of the agents, is both EF1 and PO. However, their result does not lead to fast computation, since the Nash welfare is NP-hard to compute, even approximately [Lee, 2015]. [Barman et al., 2018] and later [Garg and Murhekar, 2021] used another approach of using competitive equilibria (CE) to design pseudo-polynomial time algorithms for computing an EF1 and PO allocation of goods. In a CE allocation, agents have a fictitious amount of money, goods are assigned prices, and each agent is only allocated goods that have the maximum value to price ratio. The latter property ensures that the allocation is fractionally PO (fPO), which is an efficiency property stronger than PO.

EF1 and PO for chores. In contrast to goods, the problem is significantly harder for chores (items that impose a cost to agents). Indeed, the existence of an EF1 and PO allocation of chores is a major open question, let alone its computation. Formally, an instance of the problem for chores is a tuple \((N, M, D)\), where \(N = [n]\) is a set of \(n\) agents, \(M = [m]\) is a set of \(m\) indivisible chores, and \(D = \{d_i\}_{i \in N}\) is a list with \(d_i : 2^M \rightarrow \mathbb{R}_{\geq 0}\) denoting agent \(i\)'s disutility function over the chores. Let \(d_i(j)\) denote the disutility of chore \(j\) for agent \(i\). Since disutility functions are additive, we have \(d_i(S) = \sum_{j \in S} d_i(j)\) for every \(i \in N\) and \(S \subseteq M\). An allocation \(\mathbf{x} = (x_1, x_2, \ldots, x_n)\) is a partition of the chores into \(n\) bundles, where agent \(i\) receives bundle \(x_i \subseteq M\) and gets disutility \(d_i(x_i)\). An allocation \(\mathbf{x}\) is said to be:

- **EF1** if for all \(i, h \in N, \exists j \in x_i\) s.t. \(d_i(x_i \setminus \{j\}) \leq d_i(x_h)\).
- **PO** if there is no allocation \(\mathbf{y}\) that dominates \(\mathbf{x}\). An allocation \(\mathbf{y}\) dominates an allocation \(\mathbf{x}\) if for all \(i \in N\), \(d_i(y_i) \leq d_i(x_i)\), and there exists \(h \in N\) such that \(d_h(y_h) < d_h(x_h)\).

Main Question. Does every chore allocation instance \((N, M, D)\) admit an allocation that is both EF1 and PO?

Preliminary approaches of extending techniques from goods to chores reveal that the settings of goods and chores are only superficially similar. First, it is unknown whether a welfare function like Nash welfare which ensures EF1 and PO exists for chores. Second, it is not known whether the CE approach terminates for all instances. This motivates us...
to consider instances with special structure, which can potentially point us to the ‘source of hardness’ for this problem.

2 Contributions

We show that an EF1 and PO allocation of chores exists and can be computed in polynomial time for:
1. Bivalued instances [Garg et al., 2022]. In a bivalued instance, there exists \( a, b \in \mathbb{R}_{\geq 0} \) s.t. \( d_i(j) \in \{ a, b \} \) for all \( i \in N, j \in M \).
2. Three agents [Garg et al., 2023]. Note each agent can have a different disutility function.
3. Three types of agents [Garg et al., 2024], where agents of the same type have the same disutility function.
4. Two types of chores [Garg et al., 2024; Aziz et al., 2023].

Techniques. We design algorithms which use the CE framework. In a CE of chores, agents aim to earn a money by doing chores in exchange for payment. Each agent is only assigned chores that minimize the disutility to payment ratio, which ensures the allocation is PO and hence PO. Our algorithms move in the space of PO allocations by maintaining a CE allocation. To obtain EF1, we perform chore transfers from one agent to another while performing chore payment updates (either payment raises or drops) to maintain a competitive allocation. These steps are carefully designed depending on the specific structured instance in consideration. Likewise, the potential function arguments developed to prove the termination of our algorithms are involved and invoke the specialized structure of the instance.

3 Future Directions

In light of our results, the following are important questions for future investigation.
1. The existence of EF1 and PO allocations for \( n = 4 \) agents. The algorithm for \( n = 3 \) agents has the property that one agent \( i \) only loses chores in the course of the algorithm. This property is crucial, and it is unclear if such a property can be maintained for \( n = 4 \) agents.
2. The existence of EF1 and PO allocations for 2-ary instances, where \( d_i(j) \in \{ a_i, b_i \} \). This class generalizes bivalued instances.
3. The existence of an allocation that is approximately \( \alpha \)-EF\( k \) and PO, i.e., \( \alpha \)-EF\( k \)+PO, for \( \alpha \geq 1 \). In an \( \alpha \)-EF\( k \) allocation, no agent envies another up to a factor of \( \alpha \) after removing \( k \) chores. To the best of our knowledge, no results are known for any constants \( \alpha, k \).
4. The (non-)existence of weighted EF1 (wEF1) and PO allocations in the case of asymmetric agents with entitlements. Our recent work [Garg et al., 2024] shows that a wEF1 and PO allocation exists and can be computed for structured instances.

References


