Equitable Allocations of Indivisible Chores

Rupert Freeman
Microsoft Research
New York, NY, USA
rupert.freeman@microsoft.com

Rohit Vaish
Rensselaer Polytechnic Institute
Troy, NY, USA
vaishr2@rpi.edu

Sujoy Sikdar
University of Washington in St. Louis
St. Louis, MO, USA
sujoy@wustl.edu

Lirong Xia
Rensselaer Polytechnic Institute
Troy, NY, USA
xial@cs.rpi.edu

ABSTRACT

We study fair allocation of indivisible chores (i.e., items with non-positive value) among agents with additive valuations. An allocation is deemed fair if it is (approximately) equitable, which means that the disutilities of the agents are (approximately) equal. Our main theoretical contribution is to show that there always exists an allocation that is simultaneously equitable up to one chore (EQ1) and Pareto optimal (PO), and to provide a pseudopolynomial-time algorithm for computing such an allocation. In addition, we observe that the Leximin solution—which is known to satisfy a strong form of approximate equitability in the goods setting—fails to satisfy even EQ1 for chores. It does, however, satisfy a novel fairness notion that we call equitability up to any duplicated chore. Our experiments on synthetic as well as real-world data obtained from the Spliddit website reveal that the algorithms considered in our work satisfy approximate fairness and efficiency properties significantly more often than the algorithm currently deployed on Spliddit.

KEYWORDS

Fair Division; Indivisible Chores; Equitability

ACM Reference Format:


1 INTRODUCTION

Imagine a group of agents who must collectively complete a set of undesirable or costly tasks, also known as chores. For example, household chores such as cooking, cleaning, and maintenance need to be distributed among the members of the household. As another example, consider the allocation of global climate change responsibilities among the member nations in a treaty [44]. These responsibilities could entail producing more clean energy, reducing overall emissions, research and development, etc. In both of these cases, it is important that the allocation of chores is fair and that it takes advantage of heterogeneity in agents’ preferences. For instance, someone might prefer to cook than to clean, while someone else might have the opposite preference. Likewise, different countries might have competitive advantages in different areas.

Problems of this nature can be modeled mathematically as chore division problems, first introduced by Gardner [30]. Each agent incurs a non-positive utility, or cost (in terms of money, time, or general dissatisfaction), from completing each chore that she reports to a central mechanism. In this paper, our focus is on designing mechanisms to divide the chores among the agents equitably. An allocation of chores is equitable if all agents get exactly the same (dis)utility from their allocated chores. Other fairness properties can, of course, be considered too—for instance, envy-freeness dictates that no agent should prefer another agent’s assigned chores to her own. While this is not the main focus of our work, we do consider (approximate) envy-freeness in conjunction with (approximate) equitability.

Equitable allocations have been studied extensively in the context of allocating goods (i.e., items with non-negative value). When the goods are divisible (or, even more generally, in the cake-cutting setting), perfectly equitable allocations are guaranteed to exist [2, 25]. For indivisible goods, though, perfect equitability might not be possible, but approximate versions can still be achieved [28, 33].

At first glance, the problem of chore division appears similar to the goods division problem. However, there are subtle technical differences between the two settings. In the context of (approximate) envy-freeness, this contrast has been noted in several works [10, 11, 17, 39]. To take one example, it is known that an allocation of goods that is both envy-free up to one good and Pareto optimal can be found by allocating goods so that the product of the agents’ utilities—the Nash social welfare—is maximized [20]. However, maximizing the product of utilities is not sensible when valuations are negative, and no analogous procedure is known for the case of chores.

In this paper, we demonstrate a similar set of differences between the goods and chores settings in the context of equitability. Our focus is on equitability up to one/any chore (EQ1/EQX) which requires that pairwise violations of equitability can be eliminated by removing some/any chore from the bundle of the less happy agent.

For goods division, Freeman et al. [28] showed that equitability up to any good and Pareto optimality are achieved simultaneously by the Leximin algorithm. However, we show that in the chores setting, Leximin does not even guarantee equitability up to one chore.

1The Leximin algorithm maximizes the utility of the least well-off agent, and subject to that maximizes the utility of the second-least, and so on.
We therefore consider it to be the best choice for use in practice, we show that

<table>
<thead>
<tr>
<th>Without PO</th>
<th>EQ1</th>
<th>EQX</th>
<th>DEQ1</th>
<th>DEQX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Existence</td>
<td>Always exists</td>
<td>Always exists</td>
<td>Always exists</td>
<td>Always exists</td>
</tr>
<tr>
<td>(Proposition 3.2)</td>
<td>(Proposition 3.2)</td>
<td>(Proposition 3.7)</td>
<td>(Proposition 3.7)</td>
<td>(Proposition 3.7)</td>
</tr>
<tr>
<td>Computation</td>
<td>Poly time</td>
<td>Poly time</td>
<td>Poly time</td>
<td>?</td>
</tr>
<tr>
<td>(Proposition 3.2)</td>
<td>(Proposition 3.2)</td>
<td>(Proposition 3.8)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>With PO</th>
<th>EQ1</th>
<th>EQX</th>
<th>DEQ1</th>
<th>DEQX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Existence</td>
<td>Always exists</td>
<td>Might not exist</td>
<td>Always exists</td>
<td>Always exists</td>
</tr>
<tr>
<td>(Theorem 3.4)</td>
<td>(Example 3.1)</td>
<td>(Proposition 3.7)</td>
<td>(Proposition 3.7)</td>
<td>(Proposition 3.7)</td>
</tr>
<tr>
<td>Computation</td>
<td>Pseudopoly time</td>
<td>Strongly NP-hard</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>(Theorem 3.4)</td>
<td></td>
<td>(Theorem 3.3)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Summary of our theoretical results. Each cell contains the existence/computation results for various combinations of fairness and efficiency properties. Open questions are marked with '?'.
We will assume throughout, without loss of generality, that for each chore \( j \in [m] \), there exists some agent \( i \in [n] \) with a non-zero valuation for it (i.e., \( v_{i,j} < 0 \)), and for each agent \( i \in [n] \), there exists a chore \( j \in [m] \) that it has non-zero value for.

**Allocation.** An allocation \( A := (A_1, \ldots, A_n) \) is an \( n \)-partition of the set of chores \([m]\), where \( A_i \subseteq [m] \) is the bundle allocated to the agent \( i \) (note that \( A_i \) can be empty). Given an allocation \( A \), the utility of agent \( i \in [n] \) for the bundle \( A_i \) is \( v_i(A_i) = \sum_{j \in A_i} v_{i,j} \).

**Equitability.** An allocation \( A \) is said to be (a) **equitable (EQ)** if for every pair of agents \( i, k \in [n] \), we have \( v_i(A_i) \geq v_k(A_k) \); (b) **equitable up to one chore (EQ1)** if for every pair of agents \( i, k \in [n] \) such that \( A_i \neq \emptyset \), there exists a chore \( j \in A_i \) such that \( v_i(A_i \setminus \{j\}) \geq v_k(A_k) \), and (c) **equitable up to any chore (EQX)** if for every pair of agents \( i, k \in [n] \) such that \( A_i \neq \emptyset \) and for every chore \( j \in A_i \) such that \( v_{i,j} < 0 \), we have \( v_i(A_i \setminus \{j\}) \geq v_k(A_k) \). These notions have been previously studied for goods by Gourvès et al. [33] and Freeman et al. [28].

**Envy-freeness.** An allocation \( A \) is said to be (a) **envy-free (EF)** if for every pair of agents \( i, k \in [n] \), we have \( v_i(A_i) \geq v_k(A_k) \); (b) **envy-free up to one chore (EF1)** if for every pair of agents \( i, k \in [n] \) such that \( A_i \neq \emptyset \), there exists a chore \( j \in A_i \) such that \( v_i(A_i \setminus \{j\}) \geq v_k(A_k) \), and (c) **envy-free up to any chore (EFX)** if for every pair of agents \( i, k \in [n] \) such that \( A_i \neq \emptyset \) and for every chore \( j \in A_i \) such that \( v_{i,j} < 0 \), we have \( v_i(A_i \setminus \{j\}) \geq v_k(A_k) \). The notions of EF, EF1, and EFX were proposed in the context of goods allocation by Foley [27], Budish [18], and Caragiannis et al. [20], respectively.\(^3\)

It is easy to see that envy-freeness and equitability (and their corresponding relaxations) become equivalent when the valuations are identical, i.e., for every \( j \in [m] \), \( v_{i,j} = v_{k,j} \) for all \( i, k \in [n] \).

**Proposition 2.1.** When agents have identical valuations, an allocation satisfies EF/EFX/EF1 if and only if it satisfies EQ/eqx/eq1.

**Pareto optimality.** An allocation \( A \) is Pareto dominated by allocation \( B \) if \( v_k(B_k) \geq v_k(A_k) \) for every agent \( k \in [n] \) with at least one of the inequalities being strict. A Pareto optimal (PO) allocation is one that is not Pareto dominated by any other allocation.

**Leximin-optimal allocation.** A Leximin-optimal (or Leximin) allocation is one that maximizes the minimum utility that any agent achieves, subject to which the second minimum utility is maximized, and so on. The utilities induced by a Leximin allocation are unique, although there may exist more than one such allocation [25].

### 3 THEORETICAL RESULTS

This section presents our theoretical contributions. We will first consider equitability and its relaxations (Section 3.1), followed by combining these notions with Pareto optimality (Section 3.2), and subsequently also considering envy-freeness (Section 3.3). Finally, we will discuss a novel approximation of equitability called **equitability up to a duplicated chore** (Section 3.4).

\(^3\)An earlier work by Lipton et al. [15] studied a weaker approximation of envy-freeness for goods, but their algorithm is known to compute an EF1 allocation.

#### 3.1 Equitability and its Relaxations

As discussed previously, an equitable (EQ) allocation is not guaranteed to exist when allocating indivisible chores. In addition, the computational problem of determining whether a given instance has an equitable allocation turns out to be NP-complete even for identical valuations (Proposition 3.1). The proof uses a standard reduction from 3-PARTITION and is therefore omitted.

**Proposition 3.1.** Determining whether a given fair division instance admits an equitable (EQ) allocation is strongly NP-complete even for identical valuations.

The negative results regarding the existence and computation of exact equitability are in complete contrast with those of its relaxations. Indeed, when allocating indivisible chores, there always exists an allocation that is equitable up to any chore (EQX). Furthermore, such an allocation can be computed in polynomial time via a simple greedy procedure (Proposition 3.2). This algorithm is a straightforward adaptation of the algorithm of Gourvès et al. [33] for computing EQX allocations of goods.

**Proposition 3.2.** An EQX allocation of chores always exists and can be computed in polynomial time.

**Proof.** (Sketch) Our algorithm iteratively assigns the chores to the agents according to the following assignment rule: At each step, the happiest agent (i.e., one whose utility is closest to zero) is asked to select a chore from the set of available chores that it dislikes the most (i.e., the chore that gives it the most negative utility). It is easy to see that the chore assigned most recently to any agent is also its favorite (or least disliked) chore in its bundle. Thus, if an allocation is EQX before assigning a chore, then it continues to be EQX after it. The claim now follows by induction, since an empty allocation is EQX to begin with.

The positive result in Proposition 3.2 offers an interesting comparison between envy-freeness and equitability: It is not known whether an EFX allocation is even guaranteed to exist for chores, but an EQX allocation can always be computed in polynomial time.

#### 3.2 Equitability and Pareto Optimality

We will now consider equitability together with Pareto optimality. From Proposition 3.1, it is easy to see that checking the existence of an equitable and Pareto optimal (EQ+PO) allocation is strongly NP-hard (since every allocation is Pareto optimal under identical valuations). Therefore, we will strive for achieving Pareto optimality alongside approximate equitability, specifically EQ1 and EQX.

We will start by considering equitability up to any chore (EQX) and Pareto optimality. For goods allocation, Freeman et al. [28] have shown that equitability up to any good and Pareto optimality can be simultaneously achieved using the Leximin allocation.\(^4\) By contrast, as we show in Example 3.1, there might not exist an allocation that is equitable up to any chore and Pareto optimal, even when there are only two agents.

**Example 3.1 (Non-existence of EQX+PO).** Consider an instance with three chores \( c_1, c_2, c_3 \) and two agents \( a_1, a_2 \) with strictly negative (and normalized) valuations as shown below:

\[^4\]This result requires the valuations to be strictly positive.
Of the eight possible allocations in the above instance, the two allocations that assign all chores to a single agent, namely \( A^1 := \{c_1, c_2, c_3\} \) and \( A^2 := \{\emptyset\} \) violate EQX and can be immediately ruled out. Any other allocation must assign exactly one chore to one agent and two to the other. Of the three allocations in which \( a_1 \) is assigned exactly one chore, namely \( A^3 := \{\{c_1, c_2\}, \{c_3\}\} \), \( A^4 := \{\{c_2, c_3\}, \{c_1\}\} \), and \( A^5 := \{\{c_3\}, \{c_1, c_2\}\} \), none satisfies EQX. Therefore, these allocations can be ruled out as well.

This leaves us with the three allocations in which \( a_2 \) is assigned exactly one chore, namely \( A^6 := \{\{c_1, c_2\}, \{c_3\}\} \), \( A^7 := \{\{c_2, c_3\}, \{c_1\}\} \), and \( A^8 := \{\{c_1, c_3\}, \{c_2\}\} \). Among these, only \( A^7 \) satisfies EQX. However, \( A^7 \) is Pareto dominated by the allocation \( A^2 \); indeed, \( v_2(A^7) = -100 < v_2(A^2) = -2 \) and \( v_2(A^2) = -97 < v_2(A^8) = -5 \). Therefore, the above instance does not admit an EQX + PO allocation.

To make matters worse, determining whether a given instance admits an EQX and PO allocation turns out to be strongly NP-hard.

**Theorem 3.3 (Strong NP-hardness of EQX+PO).** Determining whether a given fair division instance admits an allocation that is simultaneously equitable up to any chore (EQX) and Pareto optimal (PO) is strongly NP-hard, even for strictly negative and normalized valuations.

Proof. We will show a reduction from 3-Partition, which is known to be strongly NP-hard [31, Theorem 4.4]. An instance of 3-Partition consists of a set \( X = \{b_1, \ldots, b_{3r}\} \) of \( 3r \) positive integers where \( r \in \mathbb{N} \) and the goal is to find a partition of \( X \) into \( r \) subsets \( X^1, \ldots, X^r \) such that the sum of numbers in each subset is equal to \( B := \frac{1}{r} \sum_{i \in X} b_i \).5 We will assume, without loss of generality, that for every \( i \in [3r], b_1 \) is even and \( b_i \geq 2 \). As a result, we can also assume, without loss of generality, that \( B \) is even.

We will construct a fair division instance with \( r + 1 \) agents and \( 4r + 2 \) chores (see Table 2). The set of agents consists of \( r \) main agents \( a_1, \ldots, a_r \) and a dummy agent \( d \). The set of chores consists of \( 3r \) main chores \( c_1, \ldots, c_{3r} \), \( r \) signature chores \( S_1, \ldots, S_r \), and two dummy chores \( D_1, D_2 \). The valuations of the agents are specified as follows: For every \( i \in [r] \) and \( j \in [3r] \), agent \( a_j \) values the main chore \( c_j \) at \( -b_j \), the signature chore \( S_i \) at \( -1 \), and all other chores at a large negative number \( -L \), where \( -L < -rB < -1 \). The dummy agent \( d \) values the dummy chores \( D_1 \) and \( D_2 \) at \( -1 \) and \( -B \), respectively, and all other chores at a large negative number \( -L \). In the interest of having normalized valuations, we can set \( L := \frac{(r-1)B+r+1}{4r} \). It is easy to show using standard calculus that \( -L' < -B \) for all \( r \geq 3 \). Since the condition \( r \geq 3 \) holds without loss of generality, we will assume throughout that \( -L' < -B \).

We will now argue the equivalence of solutions.

\( \Rightarrow \) Let \( X^1, \ldots, X^r \) be a solution of 3-Partition. Then, the desired allocation \( A \) can be constructed as follows: For every \( i \in [r] \), the main agent \( a_i \) gets the signature chore \( S_i \) as well as the chores corresponding to the numbers in \( X^i \). The dummy agent gets the two dummy chores. The allocation \( A \) is Pareto optimal because each chore is assigned to an agent that has the highest valuation for it (thus, \( A \) maximizes social welfare). Also, each agent’s valuation in \( A \) is \( -B - 1 \), implying that \( A \) is equitable, and hence also EQX.

\( \Leftarrow \) Now suppose that there exists an EQX and Pareto optimal allocation \( A \). Below, we will make a series of observations about \( A \) that will help us infer a solution of 3-Partition using \( A \).

**Claim 1.** No agent gets an empty bundle in \( A \).

Proof. (of Claim 1) If an agent gets an empty bundle, then some other agent will get four or more chores (as more than 4r chores will need to be allocated among \( r \) other agents). Since all valuations are strictly negative, this results in a violation of EQX.

**Claim 2.** Each main agent \( a_i \) gets its signature chore \( S_i \) in \( A \).

Proof. (of Claim 2) From Claim 1, we know that \( a_i \) owns at least one chore in \( A \). Fix any chore \( j \in A_i \). Suppose \( S_i \) is assigned to an agent \( k \) in \( A \). Notice that the valuation of agent \( k \) for \( S_i \) is \( -L' \) (depending of whether \( k \) is a main or a dummy agent). This is also the smallest valuation that agent \( k \) has for any chore (recall that \( -L < -rB - 1 \) and \( -L' < -B \)). Furthermore, since \( -b_1 \geq -2 \) for every \( i \in [3r] \), \( S_i \) is the unique favorite chore of agent \( a_i \). Therefore, after exchanging the chores \( j \) and \( S_i \), the valuation of agent \( k \) cannot decrease (due to additivity), and the valuation of agent \( a_i \) necessarily increases. Thus, the new allocation is a Pareto improvement over \( A \), which is a contradiction.

**Claim 3.** The chore \( D_1 \) is assigned to the dummy agent \( d \) in \( A \).

Proof. (of Claim 3) By an argument similar to that in the proof of Claim 2, we can show that if \( D_1 \) is not assigned to \( d \), then a Pareto improving swap between \( d \) and the owner of \( D_1 \) is possible.

**Claim 4.** The chore \( D_2 \) is assigned to the dummy agent \( d \) in \( A \).

Proof. (of Claim 4) Suppose, for contradiction, that \( D_2 \) is assigned to main agent \( a_i \) in \( A \). From Claim 2, we know that \( a_i \) also assigned its signature chore \( S_i \). Since \( S_i \) is the favorite chore of \( a_i \), the EQX condition requires that for every other main agent \( a_k \),

\[
v_k(A_k) \leq v_k(A_k \setminus \{S_i\}) \leq v_k(D_2) = -L.
\]

Even if \( a_i \) is assigned all the remaining chores whose assignment has not been finalized yet (this includes the \( 3r \) main chores), its valuation will still only be \( -rB - 1 > -L \). This would imply a violation of EQX condition between \( a_i \) and another agent, which is a contradiction.

From Claims 3 and 4, we know that \( D_1, D_2 \in A_d \). Therefore, by EQX condition, the following must hold for every main agent \( a_i \):

\[
v_i(A_i) \leq v_d(A_d \setminus \{D_1\}) \leq v_d(D_2) = -B.
\]
From Claim 2, we know that $a_i$ gets its signature chore $S_i$. Thus, the valuation of $a_i$ for the remaining items in its bundle must be

$$v_i(A_i \setminus \{S_i\}) \leq -B + 1.$$  \hfill (1)

Since the assignment of all signature and dummy chores has been fixed, the set $A_i \setminus \{S_i\}$ can only have main chores. By assumption, main agents have even-valued valuations for main chores. By additivity of valuations, the quantity $v_i(A_i \setminus \{S_i\})$ must also be even. Also, $-B$ is even, so $-B + 1$ must be odd, and therefore the inequality in Equation (1) must be strict. Thus, $v_i(A_i \setminus \{S_i\}) \leq -B$.

We can now infer a solution of 3-PARTITION as follows: For every $i \in [r]$, the set $X^i$ contains those numbers whose corresponding chores are included in $A_i \setminus \{S_i\}$. Since $v_i(A_i \setminus \{S_i\}) \leq -B$, it follows that all main chores must be assigned among the main agents, implying that $X^1, \ldots, X^r$ constitute a valid partition of $X$. Furthermore, the sum of numbers in the set $X^i$ cannot exceed $B$, or otherwise the sum of numbers in some other set $X^k$ will be strictly less than $B$, which would violate the above inequality. Hence, $X^1, \ldots, X^r$ is a valid solution of 3-PARTITION, as desired.

The negative results concerning the existence and computation of EQX+PO lead us to consider a weaker relaxation of equitability, namely equitability up to one chore (EQ1). A natural starting point in studying the existence of EQ1+PO allocations is the Leximin solution, as it yields strong positive results for the goods setting [28]. Unfortunately, as Example 3.2 shows, Leximin sometimes fails to satisfy EQ1 (as well as EF1) for chores.

**Example 3.2 (Leximin fails EQ1 and EF1).** Consider the following instance with four chores and three agents with normalized and strictly negative valuations:

<table>
<thead>
<tr>
<th></th>
<th>$c_1$</th>
<th>$c_2$</th>
<th>$c_3$</th>
<th>$c_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_1$</td>
<td>$-1$</td>
<td>$-5$</td>
<td>$-5$</td>
<td>$-5$</td>
</tr>
<tr>
<td>$a_2$</td>
<td>$-1$</td>
<td>$-2$</td>
<td>$-2$</td>
<td>$-1$</td>
</tr>
<tr>
<td>$a_3$</td>
<td>$-6$</td>
<td>$-5$</td>
<td>$-3$</td>
<td>$-2$</td>
</tr>
</tbody>
</table>

We will show that the allocation $A$ given by $A_1 = \{c_1\}$, $A_2 = \{c_2, c_3\}$, and $A_3 = \{c_4\}$ is Leximin-optimal. Suppose, for contradiction, that another allocation $B$ is a Leximin improvement over $A$. The utility profile induced by $A$ is $(-1, -4, -2)$, and therefore, for any chore $j$ and agent $i$ such that $j \in B_i$, we must have that $v_{i,j} \geq -4$. The chore $c_4$ is valued at less than $-4$ by both $a_1$ and $a_2$, so we must have $c_4 \in B_3$. Similarly, we can fix $c_2 \in B_2$. This, in turn, forces us to fix $c_3 \in B_2$, since otherwise if $c_i \in B_2$, then the utility of $a_3$ will be $-5 < -4$, which would violate the Leximin improvement assumption. By a similar argument, we have $c_1 \in B_1$. This, however, implies that $A$ and $B$ are identical, which is a contradiction. Therefore, $A$ must be Leximin. Notice that $A$ violates EQ1 and EF1 for the pair $(a_1, a_2)$. \hfill \Box

Another natural approach to show the existence of EQ1+PO allocations could be to use the relax-and-round framework. Specifically, one could start from an egalitarian-equivalent solution [38] (i.e., a fractional allocation that is perfectly equitable and minimizes the agents’ disutilites), and use a rounding algorithm to achieve EQ1. However, there is a simple example where this approach fails.\footnote{Consider an instance with three chores and three agents. Agents 1 and 2 value the first chore at $-4$ and the other two chores at $-\infty$ (or a suitably large negative value). Agent 3 values the first chore at $-\infty$ and the other two chores at $-1$ each. An egalitarian-equivalent solution divides the first chore equally between agents 1 and 2, and assigns the remaining two chores to agent 3. Any rounding of this fractional allocation violates EQ1 with respect to agent 3 and whoever of agents 1 or 2 gets an empty bundle.}

The failure of Leximin and the relax-and-round framework in achieving EQ1 prompts us to consider a different approach for studying approximately fair and Pareto optimal allocations. This approach, which is based on Fisher markets [13], has been successfully used in the goods model to provide an algorithmic framework for computing EF1+PO [9] and EQ1+PO [28] allocations.\footnote{Similar techniques have also been used in developing approximation algorithms for Nash social welfare objective for budget-additive and multi-item concave utilities [23].} Note that the existence of such allocations was established by means of computationally intractable methods, namely the Maximum Nash Welfare and Leximin solutions [20, 28].

Briefly, the idea is to start with an allocation that is an equilibrium of some Fisher market. By the first welfare theorem [36], such an allocation is guaranteed to be Pareto optimal. By using a combination of local search and price change steps, our algorithm converges to an approximately equitable equilibrium, which gives an approximately equitable and Pareto optimal allocation. It is worth noting that while the existing Fisher market based approaches use price-rise [9, 28], our algorithm instead uses price-drop as the natural option for negative valuations.

Our main result in this section (Theorem 3.4) establishes the existence of EQ1 and PO allocations using the markets framework.

**Theorem 3.4 (Algorithm for EQ1+PO).** Given any chores instance with additive and integral valuations, an allocation that is equitable up to one chore (EQ1) and Pareto optimal (PO) always exists and can be computed in $O(\text{poly}(m, n, |v_{\text{min}}|))$ time, where $v_{\text{min}} = \min_{i,j} v_{i,j}$.

In particular, when the valuations are polynomially bounded (i.e., for every $i \in [n]$ and $j \in [m]$, $v_{i,j} \leq \text{poly}(m, n)$), our algorithm computes an EQ1 and PO allocation in polynomial time. Whether an EQ1+PO allocation can be computed in polynomial time without this assumption is an interesting avenue for future research.\footnote{Interestingly, similar questions concerning the computation of EF1+PO or EQ1+PO allocations are also open in the goods setting [9, 28].}

The proof of Theorem 3.4 is deferred to the full version of the paper [29]. Here, we will provide an informal overview of the algorithm by demonstrating its execution on the instance in Example 3.2 where Leximin fails to satisfy EQ1.

**Example 3.3.** Consider once again the instance in Example 3.2. Our algorithm in Theorem 3.4 works in three phases. In Phase 1, the algorithm creates an equilibrium allocation by assigning each chore to an agent that has the highest valuation for it and setting its price to be (the absolute value of) the owner’s valuation; see Figure 1a. This ensures that the allocation satisfies the maximum bang-per-buck or MBB property (i.e., each agent’s bundle consists only of items with the highest valuation-to-price ratio for that agent). The MBB property guarantees that the allocation at hand is an equilibrium of some Fisher market, and therefore Pareto optimal.

The allocation constructed in Phase 1 is not EQ1 as $a_2$ gets three negatively valued chores and $a_1$ gets none. So, the algorithm switches to Phase 2, where it uses local search to address the equitability violations. Specifically, if there is an EQ1 violation, then there must be one involving the ‘happiest’ agent, i.e., agent with the highest utility (shaded in green in Figure 1a). The algorithm now proceeds to
We will now consider all three notions—equitability, envy-freeness, and Pareto optimality— together. It turns out that the existence result for EQ1+PO allocations does not hold up when we also require EF1 (Proposition 3.5).

**Proposition 3.5 (Non-existence of EQ1+EF1+PO).** There exists an instance with normalized and strictly negative valuations in which no allocation is simultaneously equitable up to one chore (EQ1), envy-free up to one chore (EF1), and Pareto optimal (PO).

**Proof.** Consider the following instance with eight chores and four agents with normalized and strictly negative valuations:

<table>
<thead>
<tr>
<th>chore</th>
<th>c1</th>
<th>c2</th>
<th>c3</th>
<th>c4</th>
<th>c5</th>
<th>c6</th>
<th>c7</th>
<th>c8</th>
</tr>
</thead>
<tbody>
<tr>
<td>a1</td>
<td>-10</td>
<td>-10</td>
<td>-10</td>
<td>-10</td>
<td>-10</td>
<td>-10</td>
<td>-10</td>
<td>-10</td>
</tr>
<tr>
<td>a2</td>
<td>-10</td>
<td>-10</td>
<td>-10</td>
<td>-10</td>
<td>-10</td>
<td>-10</td>
<td>-10</td>
<td>-10</td>
</tr>
<tr>
<td>a3</td>
<td>-73</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>a4</td>
<td>-73</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
</tbody>
</table>

Suppose, for contradiction, that there exists an allocation $A$ that is EQ1, EF1, and PO. Then, we claim that $a_1$ gets exactly one chore in $A$. Indeed, $a_1$ cannot get three or more chores in $A$, since that would result in some other agent getting at most one chore, creating an EF1 violation with respect to $a_1$. If $a_1$ gets exactly two chores, then either $a_3$ or $a_4$ will create an EQ1 violation with respect to $a_1$. This is because one of $a_3$ and $a_4$ will necessarily miss out on $c_1$ and therefore have a utility of at least $-7$ from the remaining chores. Finally, if $a_1$ does not get any chore, then one of the other agents will get at least three chores. Because of strictly negative valuations, this will create an EQ1 violation with $a_1$. Therefore, $a_1$ gets exactly one chore in $A$. By a similar argument, $a_2$ must also get exactly one chore.

Therefore, a total of six chores are assigned between $a_3$ and $a_4$. Assume, without loss of generality, that $a_3$ gets at least three chores. Then, whoever of $a_3$ or $a_4$ misses out on $c_1$ will create an EF1 violation with respect to $a_3$, giving us the desired contradiction.

Turning to the computational question, we notice that the allocation constructed in the proof of Theorem 3.3 is envy-free. Therefore, checking the existence of an EQX+PO+EF/EFX/EF1 allocation is also strongly NP-hard. We note that the analogous problem in the goods setting is also known to be computationally hard [28].

**Corollary 3.6 (Hardness of EQX+PO+EF/EFX/EF1).** Determining whether a given fair division instance admits an allocation that is simultaneously $X + Y + PO$, where $X$ refers to equitable up to any chore (EQX), and $Y$ refers to either envy-free (EF), envy-free up to any chore (EFX), or envy-free up to one chore (EF1), is strongly NP-hard, even for normalized valuations.

### 3.4 Equitability up to a Duplicated Chore

In this section, we will explore a slightly different version of approximate equitability for chores wherein instead of removing a

![Figure 1: Executing the EQ1+PO algorithm from Theorem 3.4 on the instance in Example 3.2. The solid (respectively, dashed) edges denote items that are allocated to (respectively, in the MBB set of) an agent. The edge labels denote the valuations, and the numbers next to the agent and chore nodes denote the utilities and the prices, respectively.](image-url)
chores can be computed in polynomial time via a greedy algorithm. The proof of this result is deferred to the full version [29].

Proposition 3.7 (Existence of DEQX+PO). Given any fair division instance with additive valuations, an allocation that is equitable up to any duplicated chore (DEQX) and Pareto optimal (PO) always exists.

Proof. (Sketch) We will show that any Leximin-optimal allocation, say A, satisfies DEQX (Pareto optimality is easy to verify). Suppose, for contradiction, that there exist agents i, k ∈ [n] with Ai ≠ ∅ and some chore j ∈ Aj such that vi,j < 0 and vi(Ai) < vi(Ak ∪ {j}). Let B be an allocation derived from A by transferring the chore j from agent i to agent k. That is, Bk := Ai \ {j}, Bk := Ak ∪ {j} and Bk := Ak for all h ∈ [n] \ {i, k}. Since DEQX is violated with respect to chore j, we have that vi,j < 0, and therefore vi(Bk) = vi(Ai) − vi,j > vi(Ak). Furthermore, vi(Bk) = vi(Ak ∪ {j}) > vi(Aj) by the DEQX violation condition. The utility of any other agent is unchanged. Therefore, B is a ‘Leximin improvement’ over A, which is a contradiction. □

Thus, Proposition 3.7 shows that the duplicate version of approximate equitability (DEQX) compares favorably against the standard version (EQX) in the sense that a DEQX+PO allocation is guaranteed to exist whereas an EQX+PO allocation might not exist even with two agents and strictly negative valuations (Example 3.1).

On the computational side, we find that a DEQ1 allocation of chores can be computed in polynomial time via a greedy algorithm. The proof of this result is deferred to the full version [29].

Proposition 3.8. A DEQ1 allocation of chores always exists and can be computed in polynomial time.

Unfortunately, the greedy algorithm in Proposition 3.8 does not guarantee a DEQX allocation. This stands in contrast to the situation for EQX, which is easily achieved by a greedy procedure. Settling the complexity of computing DEQX allocations is an interesting question for future work.

The complexity of computing an allocation that satisfies either DEQ1+PO or the stronger DEQX+PO also remains open. For DEQ1+PO, a natural approach would be to apply the market techniques used in Theorem 3.4, but that would require care as DEQ1 lacks the following “monotonicity” property that EQ1 has: If an allocation is not EQ1, then without loss of generality, there exists a violation with respect to the happiest agent. The same is not true for violations of DEQ1, which makes the analysis less obvious.

In the full version of the paper [29], we explore a variant of DEQX, denoted as DEQX0, in which the vi,j < 0 condition is not imposed on the duplicated chore j. With this modification, we show that computing an allocation satisfying DEQX0+PO is NP-hard, as well as an equivalent result for the analogous notion of EQX0.

 Remark 3.2 (A tractable special case: binary valuations). An instance is said to have binary valuations if for every agent i ∈ [n] and every chore j ∈ [m], we have vi,j ∈ {−1, 0}. For this restricted setting, there is a simple polynomial-time algorithm that gives an EQX+DEQX+EFX+PO allocation, as follows: If a chore is valued at 0 by one or more agents, then it is arbitrarily assigned to an agent that values it at 0. The remaining chores, which are valued at −1 by every agent, are assigned in a round-robin fashion.

4 EXPERIMENTS

In this section, we will compare various algorithms in terms of how frequently they satisfy different combinations of fairness and efficiency properties on synthetic as well as real-world datasets.

For synthetic data, we follow the setup of Freeman et al. [28] for goods by fixing n = 5 agents, m = 20 chores, and generating 1000 instances with the (negation of) the valuations drawn from Dirichlet distribution. Additional pre-processing is required to ensure that the valuations are integral and normalized; the details are deferred to the full version of the paper [29]. Recall that integral valuations are required for Theorem 3.4. None of our results require normalization, but it is a natural condition to impose in practice.

The real-world dataset consists of 2613 instances obtained from the Spliddit website [32], with the number of agents ranging from 2 to 15, and the number of distinct chores ranging from 3 to 1100. Unlike the goods case, the “task division” segment of Spliddit allows distinct items to have multiple copies. Furthermore, instead of directly eliciting additive valuations (as is the case for goods), the website asks the users to specify their preferences in the form of multipliers; that is, given two chores c1 and c2, how many times would a user be willing to complete c1 instead of completing c2 once. As a result, the elicited valuations might not be integral. These design features force us to make a number of pre-processing decisions; in particular, in order to ensure integrality of valuations and remain as faithful as possible to the Spliddit instances, we have to give up on normalization.

We consider the following four algorithms: (1) The greedy algorithm from Proposition 3.2, (2) the Leximin solution, (3) the market-based algorithm Alg-EQ1+PO from Theorem 3.4, and (4) an algorithm currently deployed on the Spliddit website for dividing chores. The latter is a randomized algorithm that computes an ex ante equitable lottery over integral allocations; see the full version of the paper for details [29].

Figure 2 presents our experimental results. For each property combination (X-axis), the plots show the % of instances (Y-axis) for which each algorithm achieves those properties. The rightmost set of bars present a comparison of the running times. For the Spliddit algorithm, we plot the average values obtained from 100 runs, and the error bars show one standard deviation around the mean.

Starting with exact equitability, we observe that a very small fraction of instances (< 20% in Spliddit and none in Synthetic) admit EQ and EQ+PO allocations, as one might expect. For the approximate notions, the greedy algorithm finds EQX allocations on all

9 http://www.spliddit.org/apps/tasks
10 E.g., doing laundry 2.5 times could be equivalent to washing dishes once.
11 An equitable (EQ) and Pareto optimal allocation (PO), whenever it exists, is provably achieved by the Leximin algorithm.
instances as advertised (Proposition 3.2), but its performance drops off sharply when PO is also required; in particular, for Synthetic data, the greedy outcome is always Pareto dominated.

Leximin performs remarkably well across the board. In addition to satisfying DEQX+PO on all instances (Proposition 3.7), it also satisfies EQX and EFX on more than 80% of the instances in both datasets. Unfortunately, it is also the slowest of all algorithms, with an average runtime of ∼140 seconds on Synthetic dataset, compared to <1 second runtime of the fastest (greedy) algorithm.

The market-based algorithm Alg-EQ1+PO computes EQ1+PO allocations as expected (Theorem 3.4), and somewhat surprisingly, also satisfies DEQ1 (and EF1). However, its performance drops off when stronger approximations of EQX/DEQX are required.

The Spliddit algorithm is consistently (and often, significantly) outperformed by Leximin and Alg-EQ1+PO, even on the Spliddit dataset. The reason is that the Spliddit algorithm is perfectly equitable ex ante but not necessarily EQ1 ex post. As a result, it is better suited for ensuring fairness over time, say, when the same set of chores are repeatedly divided among the same agents, as noted on the Spliddit website.

In summary, Leximin emerges as the algorithm of choice in terms of simultaneously achieving approximate fairness and economic efficiency. We find it intriguing that the same algorithm was also a clear winner in the experimental analysis of Freeman et al. [28] to achieve EQ1+PO in the goods setting works for chores, it may be the case that no allocation satisfying EQX+PO exists in the chores setting. In response to this possible nonexistence, we have defined two new notions of relaxed equitability, DEQ1 and DEQX, that address equitability violations by adding chores to bundles rather than removing them. A number of open questions remain regarding the computation of allocations that satisfy these notions (with or without Pareto optimality). It may also be an interesting topic for future work to consider similar relaxations of envy-freeness in the chores setting.

In our experimental analysis, we have considered four different algorithms for chore division on both a real-world dataset gathered from the Spliddit website as well as a synthetic dataset. Our experiments present a compelling case that, in practice, Leximin is the best known algorithm for one-shot allocation of indivisible chores. This is true not only with respect to (relaxed) equitability, but also (relaxed) envy-freeness and Pareto optimality.

ACKNOWLEDGMENTS

We are grateful to the anonymous reviewers for their helpful comments, and to Ariel Procaccia and Nisarg Shah for sharing with us the data from Spliddit. LX acknowledges NSF #1453542 and #1716333 for support.

5 DISCUSSION

We studied equitable allocations of indivisible chores in conjunction with other well-known notions of fairness (envy-freeness) and economic efficiency (Pareto optimality), and provided a number of existential and computational results. Our results reveal some interesting points of difference between the goods and chores settings. While a modification of the market approach used by Freeman et al. [28] to achieve EQ1+PO in the goods setting works for chores, it may be the case that no allocation satisfying EQX+PO exists in the chores setting. In response to this possible nonexistence, we have defined two new notions of relaxed equitability, DEQ1 and DEQX, that address equitability violations by adding chores to bundles rather than removing them. A number of open questions remain regarding the computation of allocations that satisfy these notions (with or without Pareto optimality). It may also be an interesting topic for future work to consider similar relaxations of envy-freeness in the chores setting.
REFERENCES


