Artificial Intelligence 347 (2025) 104386

Contents lists available at ScienceDirect

Artificial Intelligence

journal homepage: www.elsevier.com/locate/artint

Weighted EF1 allocations for indivisible chores ☆

Xiaowei Wu *, Cong Zhang, Shengwei Zhou

IOTSC, University of Macau, Taipa, 999078, Macau SAR, China

A B S T R A C T

We study how to fairly allocate a set of indivisible chores to a group of agents, where each agent i has a non-negative weight w_i that represents her obligation for undertaking the chores. We consider the fairness notion of *weighted envy-freeness up to one item* (WEF1) and propose an efficient picking sequence algorithm for computing WEF1 allocations. Our analysis is based on a natural and powerful continuous interpretation for the picking sequence algorithms in the weighted setting, which might be of independent interest. Using this interpretation, we establish the necessary and sufficient conditions under which picking sequence algorithms can guarantee other fairness notions in the weighted setting. We also study the best-of-both-worlds setting and propose a lottery that guarantees ex-ante WEF and ex-post WEF(1, 1). Then we study the existence of fair and efficient allocations and propose efficient algorithms for computing WEF1 and PO allocations for bi-valued instances. Our result generalizes that of Garg et al. (AAAI 2022) and Ebadian et al. (AAMAS 2022) to the weighted setting. Our work also studies the price of fairness for WEF1, and the implications of WEF1 to other fairness notions.

1. Introduction

As a classic problem that can be traced back to 1948 [1], fair allocation has received much attention in the past decades, in the fields of computer science, economics, and mathematics. While the traditional study of fair allocation focused on divisible items [2–4], there is an increasing attention to the fair allocation of indivisible items in recent years. In this problem, our goal is to allocate a set M of m indivisible items to a set N of n agents, where agents may have different valuation functions on the items. An allocation is defined as an n-partition $\mathbf{X} = (X_1, X_2, \dots, X_n)$ of the items, where $X_i \cap X_j = \emptyset$ for all $i \neq j$ and $\cup_{i\in N} X_i = M$. We say that X_i is the bundle assigned to agent $i \in N$. Depending on whether the agents have positive or negative values on the items, there are two lines of research, one for the allocation of goods and the other for chores. In this work, we focus on the allocation of chores, in which the agents have negative values on the items. For convenience of notation, we assume that each agent $i \in N$ has a cost function c_i that assigns a non-negative cost to each bundle of items. In this work, we assume that all cost functions are additive.

Two of the most well studied fairness notions are *envy-freeness* (EF) [2] and *proportionality* (PROP) [1]. An allocation is PROP if every agent receives a bundle with cost at most her proportional cost of all items, i.e., $c_i(X_i) \le \frac{1}{n} \cdot c_i(M)$ for all $i \in N$. An allocation is EF if under the allocation no agent wants to exchange her bundle of items with some other agent to decrease her cost, i.e., $c_i(X_i) \le c_i(X_j)$ for all $i \neq j$. Observe that every EF allocation is PROP. For the allocation of divisible items, EF allocations and PROP allocations are guaranteed to exist [3,4]. However, PROP allocations (and thus EF allocations) are not guaranteed to exist when items are indivisible, e.g., consider the allocation of a single item to two agents having non-zero cost on the item. Therefore, researchers have proposed several relaxations of these fairness notions. *Envy-freeness up to one item* (EF1) [5,6] is one of the most well-studied relaxations of envy-freeness. An allocation is said to be EF1 if the envy between any two agents can be eliminated after removing one chore from the bundle of the envious agent or removing one good from the envied agent. It has been shown that EF1 allocations

☆ Part of the results appeared in the 24th ACM Conference on Economics and Computation (EC 2023). The authors are ordered alphabetically. The project is funded by the Science and Technology Development Fund, Macau SAR (file no. 0147/2024/RIA2, 0014/2022/AFJ, 0085/2022/A, and 001/2024/SKL).

* Corresponding author.
E-mail addresses: xiaoweiwu@um.edu.mo (X. Wu), yc27429@um.edu.umo (C. Zhang), yc17423@um.edu.mo (S. Zhou).

https://doi.org/10.1016/j.artint.2025.104386
Received 19 February 2024; Received in revised form 22 April 2025; Accepted 8 June 2025

Available online 16 June 2025

always exist for the cases of goods [5], chores, and even mixed items [7,8]. Another relaxation is the *envy-freeness up to any item* (EFX) proposed by Caragiannis et al. [9], which is stronger than EF1 and requires that envy can be eliminated after removing any chore from the envious agent or removing any good from the envied agent. However, unlike EF1, EFX allocations are guaranteed to exist only for some very special cases [10–15]. Whether EFX allocations exist in general remains one of the biggest open problems. Similarly, *proportionality up to one item* (PROP1) [16] and *proportionality up to any item* (PROPX) [17] are two well-known relaxations of proportionality. PROP1 allocations have been proven to exist, for both goods [16] and chores [17]. PROPX allocations might not exist for goods [17] but can be found in polynomial time for chores [12]. For a more detailed review of the fair allocation problem, please refer to surveys by Amanatidis et al. [18] and Aziz et al. [19].

Weighted setting. While the traditional fair allocation problem focuses on the case where agents have equal obligations, in the real world, it often happens that agents are not equally obliged. For example, a person in a leadership position is naturally expected to undertake more responsibilities for finishing the set of tasks. To model these applications, the *weighted* (or *asymmetric*) setting is proposed [20,12]. In the weighted setting, each agent $i \in N$ has a weight $w_i > 0$ that represents the obligation of agent i on the chores, and we have $\sum_{i\in N} w_i = 1$. Note that in the unweighted case, we have $w_i = 1/n$ for all $i \in N$. Chakraborty et al. [20] introduced the *weighted envy-freeness up to one item* (WEF1) for the allocation of goods and showed that WEF1 allocations always exist and can be computed in polynomial time. Aziz et al. [12] considered the allocation of chores and proposed an algorithm that computes WEF1 allocations for the *identical ordering* (IDO) instance and WPROPX allocations for general instances. The existence of WEF1 allocations for general instances remains unknown (until the present work). Due to the setting of chores being less understood than goods, it is interesting to investigate the existence and computation of WEF1 for chores [20] and has been proposed as an open problem [12,19]. For a more comprehensive review of weighted allocations, please refer to the recent survey by Suksompong [21].

Best-of-both-worlds. The research mentioned above mainly focuses on deterministic allocations that satisfy some fairness. On the other hand, guaranteeing envy-freeness or proportionality using randomized allocations can be trivially done by allocating all items to an agent selected uniformly at random, which however gives highly unfair allocation for every realization. Hence recently, a line of literature [22–25] focused on the results that guarantee fairness for both the allocation in expectation (ex-ante) and the allocation that is realized (ex-post), referred to as *best-of-both-worlds* (BoBW) results. Among these, one important result achieved by Aziz [22] and Freeman et al. [23] is the ex-ante EF and ex-post EF1 lottery for the allocation of goods. Recently, Aziz et al. [26] and Hoefer et al. [27] both considered the BoBW results for allocating goods to agents with general weights.

Efficiency. Besides fairness, efficiency is another important measurement of the quality of allocations. Unfortunately, efficiency and fairness are often competing with each other, e.g., many of the fair allocations give very bad efficiency guarantees. Therefore, the existence of fair and efficient allocations has recently drawn significant attention. Popular efficiency measurements include the social cost $\sum_{i\in N} c_i(X_i)$ and Pareto optimality. An allocation is said to be *Pareto optimal* (PO) if there does not exist another allocation that can decrease the cost of some agent without increasing the costs of other agents. For the allocation of goods, Caragiannis et al. [9] showed that the allocation maximizing *Nash social welfare* is EF1 and PO for unweighted agents. For the allocation of chores, the existence of EF1 and PO allocations is still a major open problem. Allocations that are EF1 and PO are known to exist only for some special cases, e.g., two agents [8], three agents [28], *bi-valued* instances [29,30], and two types of chores [31]. Recently, Garg et al. [32] showed the existence of WEF1 and PO allocations for the cases of three types of agents and two types of chores. Their methods extend our continuous interpretation for the picking sequence algorithms to structured instances while maintaining a competitive equilibrium throughout the algorithm. Besides, the *price of fairness* (PoF) that measures the loss in social welfare/cost due to the fairness constraints has also received increasing attention [33–37].

1.1. Our contribution

In this paper, we consider the existence, computation, and efficiency of WEF1 allocations for indivisible chores. We first show that WEF1 allocations always exist for chores by proposing a polynomial-time algorithm based on the weighted picking sequence protocols.

Result 1 *(Theorem 3.2).* For the allocation of chores to weighted agents, there exists a polynomial time algorithm that computes WEF1 allocations.

Similar to existing works [20,12,26,38], our algorithm uses a weighted picking sequence to decide the order following which agents pick their favorite items. Thus, our algorithm works under the ordinal setting in which we only know the ranking of each agent over the items (instead of the actual costs). Our analysis is based on a natural continuous interpretation of the picking sequence algorithm in the weighted setting, which was first used by Aziz et al. [12] to show the existence of WEF1 allocations for IDO instances. Moreover, using the continuous interpretation, we reproduce the proof of Chakraborty et al. [20] for the existence of WEF1 allocations for goods. We also establish the necessary and sufficient conditions under which the picking sequences can guarantee other weighted fairness notions, e.g., WEF(x, y), where weighted envy-freeness is obtained by deleting x chores from the bundle of the envious agent and adding y to the envied agent [38].

We further consider the best-of-both-worlds results for the weighted envy-freeness for chores. Analogous to the recent works on the allocation of goods to weighted agents [26,27], we propose a lottery that guarantees ex-ante WEF and ex-post WEF$(1,1)$. We show that the result is tight by providing a hardness for which ex-ante WEF is incompatible with WEF(x,y) for any $x+y<2$.

Result 2 *(Theorem 4.13)*. For the allocation of chores to weighted agents, there exists a lottery that guarantees ex-ante WEF and ex-post WEF$(1,1)$.

We also consider allocations that are fair and efficient. We consider *bi-valued* instances, in which there are two values $a,b>0$ and $a\neq b$ such that $c_i(e)\in\{a,b\}$ for all $i\in N$ and $e\in M$, and propose a polynomial time algorithm that computes WEF1 and PO allocations.

Result 3 *(Theorem 5.2)*. For the allocation of chores to weighted agents, there exists a polynomial time algorithm that computes WEF1 and PO allocations for bi-valued instances.

Bi-valued instances are considered as an important special case of the fair allocation problem and have been extensively studied for both goods [39,40] and chores [30,29,13]. Our result generalizes the results of Garg et al. [29] and Ebadian et al. [30] to the weighted setting. Besides bi-valued instances, we also show that WEF1 and PO allocations exist for two agents.

Finally, we consider the *price of fairness* (PoF) that measures the loss in efficiency due to the fairness constraints. In particular, we characterize the price of WEF1, which is the ratio between the minimum social cost of WEF1 allocations and that of the unconstrained allocations. For the unweighted case, it has been shown that the price of EF1 is unbounded for three or more agents, and is $\frac{5}{4}$ for two agents [37]. We generalize the result of Sun et al. [37] to the weighted setting by showing that the price of WEF1 is $\frac{4+\alpha}{4}$ for two agents, where $\alpha=\frac{\max\{w_1,w_2\}}{\min\{w_1,w_2\}}$ is the ratio between the weights of agents.

Result 4 *(Theorem 6.3)*. For the allocation of chores to weighted agents, the price of WEF1 is unbounded for three or more agents, and is $\frac{4+\alpha}{4}$ for two agents where $\alpha=\frac{\max\{w_1,w_2\}}{\min\{w_1,w_2\}}$.

1.2. Other related works

In addition to PROP1 and PROPX, MMS [41] and APS [42] are two other popular relaxations of PROP. For agents with general weights, the weighted version of MMS is studied in [43,44], and the *AnyPrice Share* (APS) fairness is proposed by Babaioff et al. [42]. They showed that there always exist $(3/5)$-approximation of APS allocations for goods and 2-approximation of APS allocations for chores. The approximate ratio for chores has been improved to 1.733 by Feige and Huang [45]. Regarding efficiency, it has been shown that WPROP1 and PO allocations always exist for chores [46], and the mixture of goods and chores [17]. Whether WPROPX and PO allocations always exist remains unknown. Besides WEF1, a *weak* weighted version of EF1 (WWEF1) is introduced by Chakraborty et al. [20]. When agents are unweighted, both WEF1 and WWEF1 reduce to EF1. They showed that for the allocation of goods, in the weighted setting, maximizing the *weighted Nash social welfare* (WNSW) fails to satisfy WEF1, but guarantees WWEF1. Chakraborty et al. [38] further introduced WEF(x,y) that generalizes WEF1, WWEF1; and WPROP(x,y) that generalizes WPROP1. Recently, Aziz et al. [26] and Hoefer et al. [27] studied the *weighted envy-freeness up to one transfer* (WEF1T), which is equivalent to WEF$(1,1)$, and proposed randomized algorithms for the allocation of goods that guarantee ex-ante WEF and ex-post WEF1T (WEF$(1,1)$).

Remark. Independent and concurrent to our work, Springer et al. [47] also investigated the existence and computation of WEF1 allocations for chores (which they call 1WEF allocations). In their algorithm, the agents pick items following the original picking sequence (with a different initialization) until n items remain. Then they give one item to each agent in the last round and show that the resulting allocation is WEF1. Different from our analysis, their proof is based on a discretized perspective and mathematical induction.

2. Preliminaries

We consider how to fairly allocate a set of m indivisible items (chores) M to a group of n agents N, where each agent $i\in N$ has a weight $w_i>0$ and $\sum_{i\in N} w_i=1$. When $w_i=1/n$ for all $i\in N$, we call the instance *unweighted*. We call a subset of items, e.g. $X\subseteq M$, a *bundle*. Each agent $i\in N$ has an additive cost function $c_i: 2^M\to\mathbb{R}^+\cup\{0\}$ that assigns a cost to every bundle of items. For convenience we use $c_i(e)$ to denote $c_i(\{e\})$, the cost of agent $i\in N$ on item $e\in M$, and thus $c_i(X)=\sum_{e\in X} c_i(e)$ for all $X\subseteq M$. We use $\mathbf{w}=(w_1,\dots,w_n)$ and $\mathbf{c}=(c_1,\dots,c_n)$ to denote the weights and cost functions of agents, respectively. For ease of notation we use $X+e$ and $X-e$ to denote $X\cup\{e\}$ and $X\setminus\{e\}$, respectively, for any $X\subseteq M$ and $e\in M$. An allocation $\mathbf{X}=(X_1,\dots,X_n)$ is an n-partition of the items M such that $X_i\cap X_j=\emptyset$ for all $i\neq j$ and $\cup_{i\in N} X_i=M$, where agent i receives bundle X_i. Given an instance $I=\langle N,M,\mathbf{w},\mathbf{c}\rangle$, our goal is to find an allocation $\mathbf{X}$ that is *fair* to all agents. We first introduce the *weighted envy-freeness* (WEF) for the allocation of chores.

Table 1
Instance showing that the weighted picking sequence protocol fails for the allocation of chores.

	e_1	e_2	e_3	e_4
agent 1	ϵ	ϵ	$0.5-\epsilon$	$0.5-\epsilon$
agent 2	ϵ	ϵ	$0.5-\epsilon$	$0.5-\epsilon$

Definition 2.1 *(WEF).* An allocation $\mathbf{X}$ is *weighted envy-free* (WEF) if for any $i, j \in N$,

$$\frac{c_i(X_i)}{w_i} \le \frac{c_i(X_j)}{w_j}.$$

Note that when the instance is unweighted, the notion of WEF coincides with the envy-freeness (EF) notion. Hence, WEF allocations are not guaranteed to exist. In the following, we study the weighted envy-freeness up to one item (WEF1), a relaxation of WEF.

Definition 2.2 *(WEF1).* An allocation $\mathbf{X}$ is *weighted envy-free up to one item* (WEF1) if for any agents $i, j \in N$, either $X_i = \emptyset$, or there exists an item $e \in X_i$ such that

$$\frac{c_i(X_i - e)}{w_i} \le \frac{c_i(X_j)}{w_j}.$$

Finally, we define the Pareto optimality (PO) that evaluates the efficiency of allocations.

Definition 2.3 *(PO).* An allocation $\mathbf{X}'$ *Pareto dominates* another allocation $\mathbf{X}$ if $c_i(X'_i) \le c_i(X_i)$ for all $i \in N$ and the inequality is strict for at least one agent. An allocation $\mathbf{X}$ is said to be *Pareto optimal* (PO) if $\mathbf{X}$ is not dominated by any other allocation.

In Section 7, we provide some proofs and examples to show the connection between WEF1 and other fairness notions. In contrast to the allocations of goods where WEF1 fails to imply WPROP1 [20], we show that every WEF1 allocation is WPROP1 for the allocation of chores. On the other hand, we show that WPROP1 allocations fail to guarantee (any approximation of) WEF1. We also show that for the allocation of chores, WEF1 gives a $(2 - \min_{i \in N}\{w_i\})$-approximation of APS (a fairness criterion introduced by Babaioff et al. [42] for weighted agents), and the approximation ratio is tight.

3. Weighted EF1 for chores

In this section, we present a polynomial-time algorithm for computing WEF1 allocations. In the unweighted setting, EF1 allocations for chores can be computed by either the *envy-cycle elimination algorithm* [5,7] or the *round-robin algorithm* [8]. Extending the first algorithm to the weighted setting fails for the case of goods, as shown by Chakraborty et al. [20]. The main reason is that swapping bundles might not help in resolving envy cycles when agents involved have different weights. For the same reason, the envy-cycle elimination cannot be extended straightforwardly to the weighted case for the allocation of chores. In contrast, the round-robin algorithm has been extended successfully to the weighted setting and becomes the *weighted picking sequence protocol* [20], which computes WEF1 allocations for goods. Similar to round-robin, in the weighted picking sequence protocol, the agents take turns picking their favorite unallocated items. In each round, the agent i with minimum $|X_i|/w_i$ is chosen to pick an item (breaking ties by agent index), where X_i is the set of items agent i receives at the moment. Note that for the unweighted case, the algorithm degenerates to the round-robin algorithm. Unfortunately, via the following simple example, we show that the algorithm fails to compute WEF1 allocations for chores, even for two agents.

Example 3.1. Consider an instance with $n = 2$ agents, with weights $w_1 = 0.3$ and $w_2 = 0.7$, and $m = 4$ items. The two agents have the same cost function, and the costs are shown in Table 1, where $\epsilon > 0$ is arbitrarily small. Running the weighted picking sequence (regardless of the tie-breaking rules), we obtain allocations in which agent 1 receives one of e_1, e_2 while agent 2 receives all other items. The returned allocations are not WEF1 because no matter which item e from X_2 is removed, $c_2(X_2 - e)/w_2$ is still much larger than $c_2(X_1)/w_1$.

The algorithm fails because agent 2 picks more items (compared to that of agent 1), and this happens *after* agent 1 picks her favorite item. This is not a problem in the case of goods because the items are allocated from the most valuable to the least valuable, but it will cause severe envy for the allocation of chores. Interestingly, we observe that if we allow agent 2 to pick two items *before* agent 1, then the resulting allocation will be WEF1. This motivates the design of our algorithm, the *Reversed Weighted Picking Sequence* (RWPS) algorithm, which generates a sequence of agents that is the same as in the weighted picking sequence protocol, but then reverses the sequence to decide the picking sequence of agents. While the algorithms are very similar, as we will show in the following section, our analysis is quite different from that of Chakraborty et al. [20], and is arguably simpler.

3.1. Reversed weighted picking sequence

In this section, we propose the Reversed Weighted Picking Sequence (RWPS) Algorithm (see Algorithm 1 for the details) that computes WEF1 allocations for the allocation of chores. In the unweighted case, a reversed picking sequence algorithm for chores appeared in [48,8].

The algorithm. We maintain a variable s_i for each agent $i \in N$, where initially $s_i = 0$. We call s_i the *size* of agent i (its meaning will be clear later). In the first phase, we generate a length-m sequence of agents σ as follows: in round-t, where $t = 1, 2, \dots, m$, we let $\sigma(t)$ be the agent with minimum s_i (breaking ties by picking the agent with minimum index), and increase s_i by $1/w_i$. In the second phase, we let agents pick items in the order of $(\sigma(m), \sigma(m-1), \dots, \sigma(1))$. In each agent $i = \sigma(t)$'s turn, she picks her favorite unallocated item, i.e., the item with minimum cost among those that are unallocated, under her own cost function. We call $(\sigma(1), \sigma(2), \cdots, \sigma(m))$ the *forward sequence* and $(\sigma(m), \sigma(m-1), \cdots, \sigma(1))$ the *reversed sequence* or *picking sequence*. Since the sequence has length m, in the final allocation, all items are allocated.

Algorithm 1: Reversed Weight Picking Sequence Algorithm.

Input: An instance $\langle M, N, \mathbf{w}, \mathbf{c}\rangle$ with additive cost valuations.
initialize $X_i \leftarrow \emptyset$ and $s_i \leftarrow 0$ for all $i \in N$, and $P \leftarrow M$;
for $t = 1, 2, \dots, m$ **do**
  let $i^* \leftarrow \arg\min_{i \in N}\{s_i\}$, breaking ties by agent index;
  set $\sigma(t) \leftarrow i^*$, and update $s_{i^*} \leftarrow s_{i^*} + 1/w_{i^*}$;
for $t = m, m-1, \cdots, 1$ **do**
  let $i \leftarrow \sigma(t)$ and $e^* \leftarrow \arg\min_{e \in P}\{c_i(e)\}$, breaking ties by item index;
  update $X_i \leftarrow X_i + e^*$ and $P \leftarrow P - e^*$;
Output: $\mathbf{X} = \{X_1, X_2, \cdots, X_n\}$.

In this section, we prove the following main result.

Theorem 3.2. *For the allocation of chores, the reversed weighted picking sequence algorithm (Algorithm 1) computes WEF1 allocations in polynomial time.*

Note that the only difference between our algorithm and that of Chakraborty et al. [20] is that they use the forward sequence as the picking sequence, while we use the backward one. For the allocation of goods, they show that the partial allocation up to round-t, for any $t = 1, 2, \dots, m$, is always WEF1 and their proof is based on mathematical induction. However, in the case of chores, we cannot guarantee that the partial allocation is always WEF1, e.g., consider the instance in Example 3.1 (in which agent 2 picks two items at the beginning of the second phase). Therefore, instead of using mathematical induction, we take a continuous perspective on the picking sequence, which was first used to analyze the algorithm that computes WEF1 allocations for chores for IDO instances [12].

Continuous perspective. Suppose that in the first phase, when we decide the forward sequence, in a round t when agent i is chosen, the variable s_i continuously increases at a rate of $1/w_i$ for one unit of time, and variable s_j does not change, for all $j \neq i$. Therefore, we can imagine that $s_i : [0, m] \to [0, \infty)$ is a non-decreasing continuous function, where $s_i(t)$ denotes the value of variable s_i at time $t \in [0, m]$. Similarly, in the second phase when we let agents pick items following the reserved sequence, we assume that in round $t = m, m-1, \dots, 1$, agent $i = \sigma(t)$ *consumes* the item she picks continuously at a rate of 1, in time interval $(t-1, t]$. Note that round m is in fact the first round in the second phase, and round 1 is the last round. To avoid confusion, we will only use the round index t to refer to a round, without saying "the first round" or "the last round".

Example 3.3. Consider the following instance with $n = 2$ agents and $m = 5$ items. Let $w_1 = 0.4$ and $w_2 = 0.6$. After executing Algorithm 1, the forward sequence is $(1, 2, 2, 1, 2)$. We can plot the functions s_1 and s_2 with a continuous domain as follows, where the time interval $(t-1, t]$ corresponds to round t in Algorithm 1. At time $t = 0, 1, 2, 3, 4$, the agent with smaller $s_i(t)$ will grow at a rate of $1/w_i$ until time $t+1$. When s_1 increases, its rate is given by $1/w_1 = 5/2$; when s_2 increases, its rate is given by $1/w_2 = 5/3$. Suppose that after the second phase, $X_1 = \{e_1, e_2\}$, where e_1 is chosen by agent 1 in round $t = 1$ and e_2 is chosen in round $t = 4$; $X_2 = \{e'_1, e'_2, e'_3\}$, where e'_1, e'_2 and e'_3 are chosen by agent 2 in rounds $t = 2$, $t = 3$ and $t = 5$, respectively. Note that since the picking sequence is the reversed sequence, we have $c_1(e_1) \ge c_1(e_2)$ and $c_2(e'_1) \ge c_2(e'_2) \ge c_2(e'_3)$. We can equivalently represent the allocation with continuous time domain as shown in Fig. 2, where each rectangle represents an item, has width 1 and height $1/w_i$, if the item is chosen by agent i.

Under the continuous perspective, we prove Theorem 3.2.

Proof of Theorem 3.2. Fixing any two agents $i, j \in N$, we show that agent i is WEF1 towards j. Suppose that in the final allocation $X_i = \{e_1, e_2, \dots, e_k\}$, where the items are sorted in increasing order of the index of rounds in which they are chosen. Therefore,

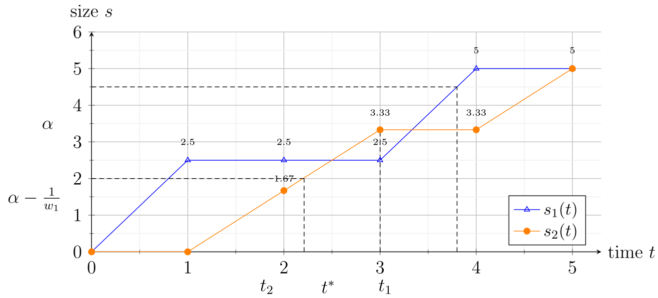

Fig. 1. The size functions of the two agents, where $s_1(t_1) = \alpha$, $t^* = \lfloor t_1 \rfloor$ and t_2 satisfies $s_2(t_2) = \alpha - 1/w_1$.

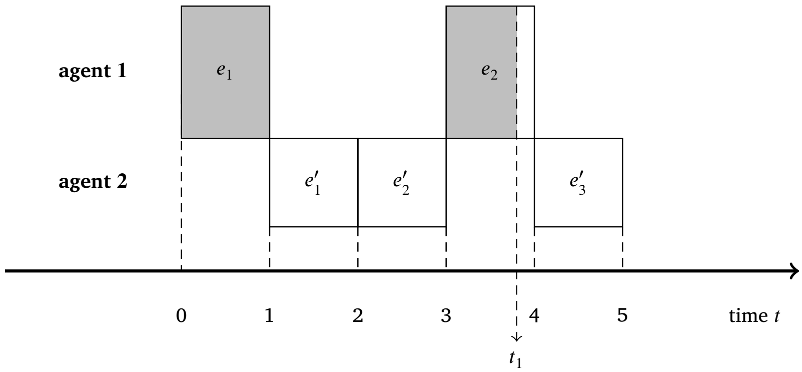

Fig. 2. Illustration of the continuous perspective of the allocation, where the size of the shadow area is $s_1(t_1)$.

$$c_i(e_1) \geq c_i(e_2) \geq \cdots \geq c_i(e_k).$$

Similarly we define $X_j = \{e'_1, e'_2, \ldots, e'_{k'}\}$, which implies

$$c_j(e'_1) \geq c_j(e'_2) \geq \cdots \geq c_j(e'_{k'}).$$

Note that $s_i(m) = \frac{k}{w_i}$ and $s_j(m) = \frac{k'}{w_j}$. In the following, we show that

$$\frac{c_i(X_i - e_1)}{w_i} \leq \frac{c_i(X_j)}{w_j}.$$

Now we take a continuous perspective and observe how $\frac{c_i(X_i)}{w_i}$ changes when t increases continuously from 0 to m. Recall that $s_i(t)$ represents the size of X_i at time t divided by w_i, and $s_i(m) = k/w_i$. When $s_i(t)$ increases from $(z-1)/w_i$ to z/w_i, where $z \in [k]$, item e_z is being consumed by agent i continuously. Let $\rho : (0, k/w_i] \to \mathbb{R}^+$ be a continuous function such that $\rho(\alpha)$ represents the cost of the item agent i is consuming when $s_i(t)$ reaches α. In particular, we have

$$\rho(\alpha) = c_i(e_z), \quad \text{for } \alpha \in \left(\frac{z-1}{w_i}, \frac{z}{w_i}\right], \text{ where } z \in \{1, 2, \cdots, k\}.$$

By definition, ρ is a non-increasing function.

Similarly, we define $\rho' : (0, k'/w_j] \to \mathbb{R}^+$ be a continuous function such that $\rho'(\alpha)$ represents the cost of the item agent j is consuming, under the cost function of agent i, when $s_j(t)$ reaches α. Hence we have

$$\rho'(\alpha) = c_i(e'_z), \quad \text{for } \alpha \in \left(\frac{z-1}{w_j}, \frac{z}{w_j}\right], \text{ where } z \in \{1, 2, \cdots, k'\}.$$

By definition of ρ and ρ', we have

$$\frac{c_i(X_i - e_1)}{w_i} = \int_{\frac{1}{w_i}}^{\frac{k}{w_i}} \rho(\alpha) d\alpha, \quad \text{and} \quad \frac{c_i(X_j)}{w_j} = \int_{0}^{\frac{k'}{w_j}} \rho'(\alpha) d\alpha.$$

Next we establish two useful lemmas to show that $\frac{c_i(X_i - e_1)}{w_i} \leq \frac{c_i(X_j)}{w_j}$.

Lemma 3.4. *We have* $\frac{k-1}{w_i} \le \frac{k'}{w_j}$.

Proof. Recall that each round, when $s_i(t)$ increases, its value grows by an amount of $1/w_i$. Consider the moment in time t' when $s_i(t)$ starts to grow from $(k-1)/w_i$, i.e., $t' = \max\{t : s_i(t) = (k-1)/w_i\}$. At time t', since $s_i(t')$ is chosen to grow, we must have that $s_i(t') \le s_j(t')$. Therefore we have

$$k'/w_j = s_j(m) \ge s_j(t') \ge s_i(t') = (k-1)/w_i,$$

as claimed. □

Lemma 3.5. *For all* $\alpha \in \left(\frac{1}{w_i}, \frac{k}{w_i}\right)$, *we have* $\rho(\alpha) \le \rho'(\alpha - \frac{1}{w_i})$.

Proof. Fix any α and suppose that $\rho(\alpha) = c_i(e_z)$, i.e., $\alpha \in \left(\frac{z-1}{w_i}, \frac{z}{w_i}\right]$. Let t_1 be the minimum[1] such that $s_i(t_1) = \alpha$. Let t^* be the maximum integer that is smaller than t_1. By definition, from time t^* to t^*+1, $s_i(t)$ grows from $(z-1)/w_i$ to z/w_i and $t_1 \in (t^*, t^*+1]$. Let t_2 be minimum time for which $s_j(t_2) = \alpha - 1/w_i$. By definition we have $\rho'(\alpha - 1/w_i) = c_i(e'_x)$, where e'_x is the item agent j is consuming at time t_2. See Fig. 1 for an example illustrating the definition of t_1, t^* and t_2, with $i = 1$ and $j = 2$.

Since at time t^*, s_i is chosen to grow, we have

$$s_i(t^*) = (z-1)/w_i \le s_j(t^*).$$

Since $\alpha \in \left(\frac{z-1}{w_i}, \frac{z}{w_i}\right]$, we have $\alpha - 1/w_i \le (z-1)/w_i$. Recall that t_2 is the minimum such that $s_j(t_2) = \alpha - 1/w_i$. Since $s_j(t^*) \ge (z-1)/w_i \ge \alpha - 1/w_i$ and $s_j(t)$ is non-decreasing, we have $t_2 \le t^*$. Since $t_1 \in (t^*, t^*+1]$, we have $t_2 \le t^* < t_1$. In other words, in the second phase of Algorithm 1, the event that "agent i includes item e_z into X_i" happens strictly earlier[2] than the event that "agent j includes item e'_x into X_j". Since agent i picks item e_z when e'_x is still unallocated, we have $c_i(e_z) \le c_i(e'_x)$, which implies $\rho(\alpha) = c_i(e_z) \le c_i(e'_x) = \rho'(\alpha - 1/w_i)$ and completes the proof. □

Given Lemma 3.4 and 3.5, we have

$$\begin{aligned} \frac{c_i(X_i \setminus \{e_1\})}{w_i} &= \int_{\frac{1}{w_i}}^{\frac{k}{w_i}} \rho(\alpha) d\alpha \le \int_{\frac{1}{w_i}}^{\frac{k}{w_i}} \rho'\left(\alpha - \frac{1}{w_i}\right) d\alpha \\ &= \int_0^{\frac{k-1}{w_i}} \rho'(\alpha) d\alpha \le \int_0^{\frac{k'}{w_j}} \rho'(\alpha) d\alpha = \frac{c_i(X_j)}{w_j}, \end{aligned}$$

where the first inequality follows from Lemma 3.5 and the second inequality follows from Lemma 3.4. Hence agent i is WEF1 towards agent j. Since agents i and j are chosen arbitrarily, the allocation is WEF1. It is straightforward that Algorithm 1 runs in $O(mn)$ time, which finishes the proof. □

Remark: allocation of goods. Note that our continuous perspective can also be applied to analyze the picking sequences for the case of goods [26,20,27,49]. We take the weighted picking sequence protocol [20] as an example and provide in Appendix A.1 an alternative proof using the continuous perspective to show that the allocation is WEF1.

3.2. Analysis of general picking sequences

The RWPS algorithm we introduced falls into the class of picking sequence algorithms.

- In the first phase, the algorithm decides a sequence of agents, which depends on the weights of agents, but is independent of the cost functions.
- In the second phase, the algorithm lets agents take turns to pick their favorite unallocated item, following the picking sequence decided in the first phase.

[1] Recall that when $\alpha = z/w_i$, there can be multiple values of t for which $s_i(t) = \alpha$.

[2] Recall that agents pick items following the reversed sequence, and at most one agent is consuming item, at any point in time.

In the RWPS algorithm, a forward sequence $\sigma = (\sigma(1), \dots, \sigma(m))$ is chosen in the first phase, while the reversed sequence $(\sigma(m), \dots, \sigma(1))$ is used as the picking sequence. We refer to this class of algorithms[3] as the reversed picking sequence algorithm, and refer to $\sigma = (\sigma(1), \dots, \sigma(m))$ as the forward sequence, $(\sigma(m), \dots, \sigma(1))$ as the reversed sequence (which is the actual picking sequence). Recall that the size

$$s_i(t) = \frac{|\{t' : \sigma(t') = i, t' \le t\}|}{w_i} \tag{1}$$

of agent i measures the weighted number of appearances of agent i in the sequence up to time t, for all $t \in \{0, 1, \dots, m\}$. We have shown that if for all $t \in \{1, 2, \dots, m\}$,

$$\sigma(t) = \arg\min_{i \in N}\{s_i(t-1)\},$$

then the resulting allocation is WEF1. In the following, we consider other picking sequences, and establish the conditions under which the resulting allocation satisfies other fairness requirements. Note that we can also use the continuous perspective we have introduced, e.g., agents continuously consume items following the reverse sequence, to analyze the algorithms.

We consider the fairness notions of *weak weighted envy-freeness up to one item* (WWEF1) and *weighted envy-freeness up to one transfer* (WEF1T) that are first proposed for the allocation of goods, by Chakraborty et al. [20] and Aziz et al. [26], respectively. In the following, we extend these notions to the allocation of chores.

Definition 3.6 *(WWEF1).* An allocation $\mathbf{X}$ is *weakly weighted envy-free up to one item* (WWEF1) if for any agents $i, j \in N$, there exists an item $e \in X_i$ such that

$$\frac{c_i(X_i - e)}{w_i} \le \frac{c_i(X_j)}{w_j} \quad \text{or} \quad \frac{c_i(X_i)}{w_i} \le \frac{c_i(X_j + e)}{w_j}.$$

Definition 3.7 *(WEF1T).* An allocation $\mathbf{X}$ is *weighted envy-free up to one transfer* (WEF1T) if for any agents $i, j \in N$, there exists an item $e \in X_i$ such that

$$\frac{c_i(X_i - e)}{w_i} \le \frac{c_i(X_j + e)}{w_j}.$$

The above relaxations of weighted envy-freeness, together with WEF1, can be unified under the fairness notion of WEF(x, y), which is first proposed for the allocation of goods by Chakraborty et al. [38]. In the following, we extend it to the allocation of chores.

Definition 3.8 *(WEF(x, y)).* For any $x, y \in [0, 1]$, an allocation $\mathbf{X}$ is *weighted envy-free up to* (x, y) (WEF(x, y)), if for any agents $i, j \in N$, there exists an item $e \in X_i$ such that

$$\frac{c_i(X_i) - x \cdot c_i(e)}{w_i} \le \frac{c_i(X_j) + y \cdot c_i(e)}{w_j}$$

By definition, WEF1 is equivalent to WEF$(1, 0)$; WEF1T is equivalent to WEF$(1, 1)$; an allocation is WWEF1 if and only if for any agents $i, j, \in N$, agent i is WEF$(1, 0)$ or WEF$(0, 1)$ towards agent j. In the following, we establish the conditions under which the allocation returned by the picking sequence algorithm guarantees WEF(x, y).

Theorem 3.9. *A reversed picking sequence algorithm computes WEF(x, y) allocations if and only if for any $t \in \{1, 2, \dots, m\}$ and any agents $i, j \in N$, we have*

$$s_i(t) - \frac{x}{w_i} \le s_j(t) + \frac{y}{w_j},$$

where $s_i(t)$ is defined as in Equation (1).

Since the proof of Theorem 3.9 follows the same framework as that of Theorem 3.2, we defer the proof to Appendix A.2. From Theorem 3.9, we have the following corollaries.

Corollary 3.9.1. *A reversed picking sequence algorithm computes WEF1 allocations if and only if for any $t \in \{1, 2, \dots, m\}$ and any pair of agents $i, j \in N$, we have $s_i(t) - \frac{1}{w_i} \le s_j(t)$.*

[3] In fact, this class is equivalent to the general picking sequence algorithm since we can simply define the reversed one as the sequence. However, for convenience and consistency of our analysis, we let σ be the sequence decided in the first phase.

Consequently, to ensure WEF1, RWPS is basically the only picking sequence algorithm one can use. Any other picking sequence algorithm that ensures WEF1 can be regarded as RWPS with a different tie-breaking rule.

Corollary 3.9.2. *A reversed picking sequence algorithm computes WEF1T (WEF$(1,1)$) allocations if and only if for any $t \in \{1,2,\dots,m\}$ and any pair of agents $i,j \in N$, we have $s_i(t) - \frac{1}{w_i} \le s_j(t) + \frac{1}{w_j}$.*

Corollary 3.9.3. *A reversed picking sequence algorithm computes WWEF1 allocations if and only if for any $t \in \{1,2,\dots,m\}$ and any pair of agents $i,j \in N$, we have $s_i(t) \le s_j(t) + \frac{1}{\min\{w_i,w_j\}}$.*

Furthermore, we show that we can compute WEF(x,y) allocations for any $x+y \ge 1$, by providing an algorithm similar to Algorithm 1 (we defer the algorithm and analysis to Appendix A.2).

4. Best-of-both-worlds guarantee for weighted envy-freeness

In this section, we consider randomized algorithms with both ex-ante and ex-post fairness guarantees. We focus on algorithms that produce random allocations that are WEF in expectation and satisfy some relaxation of WEF for every realization. This kind of algorithm is often referred to as a lottery over a collection of fair allocations, and has best-of-both-worlds fairness guarantee. For the allocation of goods, Aziz et al. [26] and Hoefer et al. [27] provide lotteries that guarantee ex-ante WEF and ex-post WEF$(1,1)$. They further show that achieving ex-ante WEF and ex-post WEF(x,y) is impossible for all $x+y<2$. In this section, we show analogous results for chores. To begin with, we borrow an example from [27] to show that ex-ante WEF and ex-post WEF(x,y) are incompatible for the allocation of chores.

Lemma 4.1. *For all $x,y \ge 0$, $x+y<2$, ex-ante WEF and ex-post WEF(x,y) are incompatible.*

Proof. Consider an instance with two agents and two items with $c_i(e)=1$ for all $i \in N, e \in M$. Let $w_1 \in (\frac{y}{2+y-x}, \frac{1}{2})$ and $w_2 = 1-w_1$. To ensure ex-ante WEF, agent 1 receives less than one item in expectation. Therefore, the allocation $\mathbf{Y} = (\emptyset, M)$ must have a non-zero probability in any ex-ante WEF lottery. However, the allocation is not WEF(x,y) to agent 2 since for both $e \in Y_2$, we have

$$\frac{c_2(Y_2) - x \cdot c_2(e)}{w_2} = \frac{2-x}{1-\frac{y}{2+y-x}} = 2+y-x = \frac{y}{\frac{y}{2+y-x}} > \frac{c_2(Y_1) + y \cdot c_2(e)}{w_1},$$

where the last inequality holds because $c_2(Y_1)=0$ and $w_1 > \frac{y}{2+y-x}$. □

Next, we present a lottery that guarantees ex-ante WEF and ex-post WEF$(1,1)$. As in existing works [22,23,26,27], our lottery is based on the decomposition framework of Budish et al. [50] on a WEF allocation for divisible items. To distinguish between allocations for indivisible items and those for divisible items, we refer to the latter as fractional allocations.

First, let's consider algorithms for computing fractional WEF allocations. Recall that we have designed the RWPS algorithm for computing WEF1 allocations for indivisible chores and introduced a continuous perspective to look at the allocation process. An interesting observation is that if we replace each indivisible item by $1/\epsilon$ copies, each representing an ϵ fraction of the original item, then running the RWPS algorithm gives an allocation that is almost WEF (up to ϵ fraction of an item). Note that when $\epsilon \to 0$, we transform the indivisible items into divisible ones, and in the RWPS algorithm all agents $i \in N$ grow their size s_i at the same speed, i.e., we have $s_i(t) = t$ for all $i \in N$ and $t \in [0,m]$. In fact, such an algorithm (when $\epsilon \to 0$) becomes the Eating algorithm [22,27] in which all agents continuously eat their most preferred[4] unallocated items until all items are fully consumed. In the weighted setting, each agent i eats at a speed of w_i, and the algorithm is named DifferentSpeedsEating (DSE) by Hoefer et al. [27]. They show that DSE computes WEF allocations for the case of goods. Since DSE is equivalent to RWPS for divisible items, the above intuition implies that DSE also computes WEF allocation for the case of chores. For completeness, we give a formal presentation of the algorithm and a short proof here.

Notation. In a fractional allocation, we use $x_{ie} \in [0,1]$ to denote the fraction of item $e \in M$ allocated to agent $i \in N$. We use $\mathbf{x}_i = \{x_{ie}\}_{e\in M}$ to denote the fractional bundle allocated to agent i, and $\mathbf{X} = \{x_{ie}\}_{i\in N, e\in M}$ to denote the fractional allocation. The cost of agent i under the fractional allocation is given by $c_i(\mathbf{x}_i) = \sum_{e\in M}(x_{ie} \cdot c_i(e))$. We assume that each agent has a strict preference order for the items, and we say that item e is *better* than e' for agent i, if i prefers e to e'.

Lemma 4.2. *The fractional allocation $\mathbf{X}$ returned by Algorithm 2 is WEF.*

[4] Throughout this section, we break ties consistently by item id. In other words, if $c_i(e) = c_i(e')$ for two items e and e', then we assume that agent i prefers the item with a smaller id. Therefore, while there may be ties in item costs, every agent has a strict preference order for the items.

Proof. Fixing any two agents $i, j \in N$, we show that agent i is WEF towards j. Since (over all agents) one unit of items is consumed in one unit of time, DSE runs in m units of time. Let $e(i,t)$ be the item agent i is consuming at time t and let $\rho(t) = c_i(e(i,t))$. Similarly, let $e(j,t)$ be the item agent j is consuming at time t and $\rho'(t) = c_i(e(j,t))$. By definition,

$$\frac{c_i(\mathbf{x}_i)}{w_i} = \int_0^m \rho(t)dt, \quad \text{and} \quad \frac{c_i(\mathbf{x}_j)}{w_j} = \int_0^m \rho'(t)dt.$$

Since agent i is always consuming her favorite available item at time $t \in [0,m]$, we have $\rho(t) \le \rho'(t)$, which implies $\frac{c_i(\mathbf{x}_i)}{w_i} \le \frac{c_i(\mathbf{x}_j)}{w_j}$. □

Algorithm 2: DifferentSpeedsEating.

Input: An instance $\langle M, N, \mathbf{w}, \mathbf{c}\rangle$ with additive cost valuations.
initialize $x_{ie} \leftarrow 0$ for all agent $i \in N$ and item $e \in M$, the remaining fraction $z(e) \leftarrow 1$ for all item $e \in M$, and the set of available items $P \leftarrow M$;
while $P \neq \emptyset$ **do**
  initialize the consumption speed $\mathsf{speed}(e) \leftarrow 0$ for all $e \in P$;
  for $i \in N$ **do**
    let $e^i \leftarrow \operatorname{argmin}_{e \in P}\{c_i(e)\}$ be the most preferred available item of agent i ;
    increase the consumption speed of item e^i: $\mathsf{speed}(e^i) \leftarrow \mathsf{speed}(e^i) + w_i$;
  let $\Delta t = \min_{e \in P}\{ \frac{z(e)}{\mathsf{speed}(e)} \}$;
  for $i \in N$ **do**
    update the fraction of item e^i allocated to agent i: $x_{ie^i} \leftarrow x_{ie^i} + \Delta t \cdot w_i$;
    update the remaining fraction of item e^i: $z(e^i) \leftarrow z(e^i) - \Delta t \cdot w_i$;
  update the set of available items: $P \leftarrow \{e \in P : z(e) > 0\}$;
Output: $\mathbf{X} = \{x_{ie}\}_{i \in N, e \in M}$.

In the following, we show that the allocation $\mathbf{X}$ returned by DSE can be decomposed into a linear combination of (integral) WEF$(1,1)$ allocations. We show that each integral allocation can be regarded as the output of a reversed picking sequence algorithm that satisfies the condition in Corollary 3.9.2, and thus is WEF$(1,1)$. By interpreting the coefficient of each allocation as the probability of realizing the allocation, we guarantee ex-ante WEF and ex-post WEF$(1,1)$.

High-level ideas. Before we proceed, we would like to make a comparison of the problem for chores and that for goods. Following the analysis of Hoefer et al. [27], we can show that each allocation $\mathbf{Y}$ in the decomposition of $\mathbf{X}$ can be regarded as the output of some picking sequence π, in which the following property is maintained: For any prefix of the sequence, suppose agent i appears t_i times and agent j appears t_j times, then we have $\frac{t_i+1}{w_i} > \frac{t_j-1}{w_j}$, which implies that the allocation is WEF$(1,1)$. Intuitively speaking, the property ensures that, for any two agents, at any point in time, the weighted numbers of times they have been asked to pick items are close. This is crucial for the allocation of goods because items are chosen from the most valuable to the least valuable. Unfortunately, such property is not enough to guarantee WEF$(1,1)$ for the allocation of chores because what really matters in the fair allocation is the number of "large items" each agent receives, which is allocated at the end of the picking sequence. Therefore in contrast to the analysis for goods, we need to show that for any two agents, at any point in time, the weighted numbers of times they have to pick items in the *remaining sequence* are close.

Recall that DSE runs in m units of time. At any time $t \in [0,m]$, agent i has eaten $w_i \cdot t$ units of items and has $w_i \cdot (m-t)$ units of items to eat.

Definition 4.3 *(Large Items at Time t).* Let $e(i,t)$ be the item agent i is consuming at time t. If agent i has just finished consuming an item e at time t and starts to consume another item, we define $e(i,t) = e$. We further define $e(i,0)$ to be an imaginary dummy item that is most preferred by agent i. For all $t \in [0,m]$, let $L(i,t)$ be the items agent i less prefers over item $e(i,t)$, e.g., the "large items" agent i has yet to eat at time t.

By definition we have $L(i,0) = M$ and $L(i,m) = \emptyset$ for all $i \in N$.

Definition 4.4 *(Eating Time).* For each item e and agent i such that $x_{ie} > 0$, we define

$$\mathsf{begin}(i,e) = \inf\{t \in [0,m] : e(i,t) = e\}, \quad \text{and}$$
$$\mathsf{end}(i,e) = \sup\{t \in [0,m] : e(i,t) = e\}$$

as the first and last time when item e is being consumed by agent i, respectively.

By definition, we have the following important observations:

Table 2
Agents' Cost Functions.

	e_1	e_2	e_3
agent 1	2	3	5
agent 2	4	1	5
agent 3	3	2	5

- If agent i starts eating item e, then she will keep eating the same item until it is fully consumed. Therefore item e is being eaten by agent i during the whole period $(\mathsf{begin}(i,e), \mathsf{end}(i,e)]$ (possibly with different consumption speeds at different times).
- For different agents, the times that they start to eat item e might be different. However, they will stop eating e at the same time (when e is fully consumed). We use $\mathsf{end}(e)$ to denote this common end time.
- At time $t = \mathsf{begin}(i,e)$, agent i has just finished eating an item and starts to eat e. Therefore we have $e(i, \mathsf{begin}(i,e)) \neq e$ and $L(i, \mathsf{begin}(i,e))$ is the set of items not better than e (including e).

Here we provide an instance I^* to illustrate the execution of DSE, the eating times, and the decomposition of $\mathbf{X}$.

Example 4.5. Consider the instance $I^* = \langle N, M, \mathbf{w}, \mathbf{c}\rangle$ where $N = \{1,2,3\}$, $M = \{e_1, e_2, e_3\}$ and $w_1 = \frac{1}{2}, w_2 = \frac{1}{3}, w_3 = \frac{1}{6}$. All cost functions are additive and the costs of items for each agent are presented in Table 2. Throughout this section, whenever we use I^*, we mean the instance we just described.

We first consider the execution of DSE on I^*.

During the running of DSE, each agent $i \in N$ consumes her most preferred available item with the speed of w_i. At time $t = 0$, agent 1 starts to consume e_1 while agents 2 and 3 start to consume e_2. At time $t = 2$, both items e_1, e_2 are fully consumed and $x_{1e_1} = 1, x_{2e_2} = \frac{2}{3}, x_{3e_3} = \frac{1}{3}$. Now item e_3 is preferred by all agents among the remaining items. At time $t = 3$, all items are fully consumed and $x_{1e_3} = \frac{1}{2}, x_{2e_3} = \frac{1}{3}, x_{3e_3} = \frac{1}{6}$. The fractional allocation $\mathbf{X}$ returned by DSE is presented by the following matrix:

$$\mathbf{X} = \begin{pmatrix} 1 & 0 & 1/2 \\ 0 & 2/3 & 1/3 \\ 0 & 1/3 & 1/6 \end{pmatrix}.$$

The eating times of items are as follows:

$$\mathsf{end}(e_1) = \mathsf{end}(e_2) = 2, \quad \mathsf{end}(e_3) = 3.$$

Next, we present the properties for decomposing $\mathbf{X}$.

Lemma 4.6. *Let $\mathbf{X}$ be the fractional allocation returned by DSE. We can decompose*

$$\mathbf{X} = \lambda_1 \mathbf{Y}^1 + \cdots + \lambda_k \mathbf{Y}^k,$$

where $\lambda_t > 0$ for all $t \le k$ and $\sum_{t=1}^{k} \lambda_t = 1$, such that every $\mathbf{Y} = (Y_1, Y_2, \ldots, Y_n)$ in the decomposition is a complete allocation and we have

- *$e \in Y_i$ only if $x_{ie} > 0$;*
- *for all $t \in [0, m]$, we have*

$$\left\lfloor \sum_{e \in L(i,t)} x_{ie} \right\rfloor \le |Y_i \cap L(i,t)| \le \left\lceil \sum_{e \in L(i,t)} x_{ie} \right\rceil.$$

Proof. We prove the lemma using the decomposition framework of Budish et al. [50]. Before we present the decomposition, we first introduce some notations.

- **Constraint Structure and Quotas.** A *constraint structure* $\mathcal{H}$ consists of a collection of subsets $S \subseteq N \times M$ where each S comes with a lower and upper integer quota denoted by $\underline{q}_S$ and $\bar{q}_S$, respectively. We denote by $\mathbf{q}$ the vector that contains all the quotas. An allocation $\mathbf{X}$ is feasible under $(\mathcal{H}, \mathbf{q})$ if for each $S \in \mathcal{H}$,

$$\underline{q}_S \le \sum_{(i,e) \in S} x_{ie} \le \bar{q}_S.$$

- **Hierarchy and Bi-hierarchy.** A constraint structure $\mathcal{H}$ is a *hierarchy* if, for every $S, S' \in \mathcal{H}$, either $S \cap S' = \emptyset$ or one is contained in the other. $\mathcal{H}$ is a *bi-hierarchy* if it can be partitioned into $\mathcal{H} = \mathcal{H}_1 \cup \mathcal{H}_2$, such that $\mathcal{H}_1 \cap \mathcal{H}_2 = \emptyset$ and both $\mathcal{H}_1$ and $\mathcal{H}_2$ are hierarchies.

Budish et al. [50] showed that given a fractional allocation $\mathbf{X}$, a bi-hierarchy $\mathcal{H}$ with quotas $\mathbf{q}$, if $\mathbf{X}$ is feasible under $(\mathcal{H},\mathbf{q})$, then we can decompose $\mathbf{X}$ into a linear combination of integral allocations such that each allocation is also feasible under $(\mathcal{H},\mathbf{q})$.

Theorem 4.7 *(Budish et al. [50]). Given any fractional allocation $\mathbf{X}$, a bi-hierarchy $\mathcal{H}$ and corresponding quotas $\mathbf{q}$, if $\mathbf{X}$ is feasible under $(\mathcal{H},\mathbf{q})$, then there exists a decomposition of $\mathbf{X}$ into integral allocations, where each allocation in the decomposition is feasible under $(\mathcal{H},\mathbf{q})$. Further, the decomposition can be obtained in strongly polynomial time.*

In the following, we construct the bi-hierarchy and the corresponding quotas with which we apply the above theorem. We set $\mathcal{H}_1 = \{C_e : e \in M\}$ where $C_e = \{(i,e) : i \in N\}$. Intuitively, the hierarchy $\mathcal{H}_1$ guarantees that any decomposed allocation is a complete integral allocation. Fix any agent $i \in N$. We assume w.l.o.g. that $(e_1, e_2, \dots, e_m)$ is the reversed preference order of agent i, e.g., item e_m is the most preferred item and e_1 is the least preferred. We set

$$\mathcal{S}_i = \{\{(i,e_1)\}, \{(i,e_1),(i,e_2)\}, \dots, \{(i,e_1),\dots,(i,e_m)\}\},$$

and $\mathcal{H}_2 = (\cup_{i\in N}\mathcal{S}_i) \cup \{\{(i,e)\} : i \in N, e \in M\}$. Let the bi-hierarchy $\mathcal{H} = \mathcal{H}_1 \cup \mathcal{H}_2$. We define the corresponding $\mathbf{q}$: for each $S \in \mathcal{H}$, $\underline{q}_S = \lfloor x_S \rfloor$ and $\bar{q}_S = \lceil x_S \rceil$, where $x_S = \sum_{(i,e)\in S} x_{ie}$. Note that $\mathbf{X}$ is obviously feasible under $(\mathcal{H},\mathbf{q})$. Applying Theorem 4.7, we can decompose $\mathbf{X} = \lambda_1 \mathbf{Y}^1 + \cdots + \lambda_k \mathbf{Y}^k$ such that every $\mathbf{Y}$ in the decomposition is also feasible under $(\mathcal{H},\mathbf{q})$. See Example 4.8 for an illustration of the decomposition.

- $\mathbf{Y}$ is a complete allocation since for any $C_e \in \mathcal{H}_1$, we have (y_{ie} is the indicator of $e \in Y_i$)

$$1 = \left\lfloor \sum_{(i,e)\in C_e} x_{ie} \right\rfloor \le \sum_{i\in N} y_{ie} \le \left\lceil \sum_{(i,e)\in C_e} x_{ie} \right\rceil = 1.$$

- For any $i \in N, e \in M$, since $\{(i,e)\} \in \mathcal{H}_2$, we have $y_{ie} \le \lceil x_{ie} \rceil = 0$ if $x_{ie} = 0$.
- For any $i \in N$, $t \in [0,m]$, suppose that $e(i,t) = e_z$, then we have $L(i,t) = \{e_1, e_2, \dots, e_{z-1}\}$. Since $\{(i,e_1),(i,e_2),\dots,(i,e_{z-1})\} \in \mathcal{S}_i$, we have (note that $|Y_i \cap L(i,t)| = \sum_{e\in L(i,t)} y_{ie}$)

$$\left\lfloor \sum_{e\in L(i,t)} x_{ie} \right\rfloor \le |Y_i \cap L(i,t)| \le \left\lceil \sum_{e\in L(i,t)} x_{ie} \right\rceil.$$

Hence we have proved the stated properties and the lemma follows. $\square$

Example 4.8. The quotas $\mathbf{q}$ related to $\mathcal{H}_1$ is quite simple: for each item $e \in M$, we have

$$1 = \sum_{i\in N} y_{ie} = 1.$$

In other words, each decomposed allocation $\mathbf{Y}$ is a complete allocation.

In the following, we show the corresponding $\mathbf{q}$ related to the hierarchy $\mathcal{H}_2$. We first consider the quotas related to agent 1. Note that

$$\mathcal{S}_1 = \{\{(1,e_1)\}, \{(1,e_1),(1,e_2)\}, \{(1,e_1),(1,e_2),(1,e_3)\}\}$$

and we only have to consider S such that $S \in \mathcal{S}_1 \cup \{\{1,e_2\},\{1,e_3\}\}$. The corresponding $\mathbf{q}$ can be represented by the following constraints:

$$\begin{aligned} y_{1e_1} &= 1, \\ y_{1e_1} + y_{1e_2} &= 1, \\ 1 \le y_{1e_1} + y_{1e_2} + y_{1e_3} &\le 2, \\ \text{and} \quad y_{1e_2} = 0, \quad 0 \le y_{1e_3} &\le 1. \end{aligned}$$

In other words, in any decomposed allocation $\mathbf{Y}$, agent 1 receives 1 or 2 items, and one must be e_1. Agent 1 will never receive item e_2.

Similarly, the quotas for agent 2 can be represented by constraints:

$$\begin{aligned} 0 \le y_{2e_2} &\le 1, \\ y_{2e_2} + y_{2e_3} &= 1, \\ y_{2e_2} + y_{2e_3} + y_{2e_1} &= 1, \\ \text{and} \quad y_{2e_1} = 0, \quad 0 \le y_{2e_3} &\le 1. \end{aligned}$$

In other words, agent 2 receives 1 item which is either e_2 or e_3.

The quotas for agent 3 can be represented by constraints:

$$\begin{aligned} 0 \le y_{3e_2} &\le 1, \\ 0 \le y_{3e_2} + y_{2e_3} &\le 1, \\ 0 \le y_{3e_2} + y_{3e_3} + y_{3e_1} &\le 1, \\ \text{and} \quad y_{3e_1} = 0, \quad 0 \le y_{3e_3} &\le 1. \end{aligned}$$

In other words, agent 3 receives at most 1 item, which cannot be e_1.

Finally, we provide a decomposition of $\mathbf{X}$. We use three different integral allocations to present the fractional allocation $\mathbf{X}$.

$$\mathbf{X} = \frac{1}{2} \cdot \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} + \frac{1}{3} \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} + \frac{1}{6} \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

In the following, we fix any deterministic allocation $\mathbf{Y}$ in the decomposition of $\mathbf{X}$ stated in Lemma 4.6. We show that $\mathbf{Y}$ can be considered as the output of a reversed picking sequence algorithm and it satisfies WEF$(1,1)$.

Definition 4.9 *(Realized Time).* Consider any item $e \in M$ and suppose that $e \in Y_i$. Let $k = |Y_i \cap L(i, \mathsf{begin}(i,e))|$. That is, there are k items in Y_i that are not better than e (including e itself). We define the realized time $t(e)$ of item e as:

$$t(e) = \max \left\{ \mathsf{begin}(i,e), m - \frac{k}{w_i} \right\}.$$

Note that realized time is well-defined since $e \in Y_i$ only if $x_{i,e} > 0$, following Lemma 4.6. While the eating times of item e depend only on $\mathbf{X}$, the realized time of e depends also on which agent receives item e in the realized allocation $\mathbf{Y}$ (and hence the name). In the following, we prove some useful properties regarding the realized time.

Lemma 4.10. *For all $i \in N$ and $e \in Y_i$, we have $t(e) \in [m - \frac{k}{w_i}, m - \frac{k-1}{w_i})$.*

Proof. By definition, $t(e) = \max\{\mathsf{begin}(i,e), m - \frac{k}{w_i}\} \ge m - \frac{k}{w_i}$. For contradiction, suppose $\mathsf{begin}(i,e) \ge m - (k-1)/w_i$. Recall that at time $\mathsf{begin}(i,e)$, agent i has $w_i \cdot (m - \mathsf{begin}(i,e))$ units of items to eat in DSE, and all these items are in $L(i, \mathsf{begin}(i,e))$. From Lemma 4.6 we have

$$|Y_i \cap L(i, \mathsf{begin}(i,e))| \le \left\lceil \sum_{e \in L(i, \mathsf{begin}(i,e))} x_{ie} \right\rceil = \lceil w_i \cdot (m - \mathsf{begin}(i,e)) \rceil \le k - 1,$$

which is a contradiction. Hence $\mathsf{begin}(i,e) < m - (k-1)/w_i$ and the lemma follows. $\square$

Lemma 4.11. *For all $i \in N$ and $e \in Y_i$, if $\mathsf{begin}(i,e) < m - \frac{k}{w_i}$, then we have $\mathsf{end}(e) > m - \frac{k}{w_i}$.*

Proof. Suppose otherwise, i.e., $\mathsf{end}(e) \le m - \frac{k}{w_i}$. Fix any $t \in [\mathsf{end}(e), m - \frac{k}{w_i}]$. Then we have $e \notin L(i,t)$ and $|Y_i \cap L(i,t)| \le k - 1$. However, since at time t, agent i has $w_i \cdot (m - t) \ge k$ units of items to eat in DSE and all these items are in $L(i,t)$, applying Lemma 4.6, we have

$$|Y_i \cap L(i,t)| \ge \left\lfloor \sum_{e \in L(i,t)} x_{ie} \right\rfloor \ge k,$$

which is a contradiction. $\square$

By the above lemma and that $\mathsf{end}(e) > \mathsf{begin}(i,e)$, we have the following immediately.

Corollary 4.11.1. *For all $e \in M$, we have $t(e) < \mathsf{end}(e)$.*

Using the above property, we prove some monotonicity properties in the realized times.

Lemma 4.12. *For all $i \in N$, $e \in Y_i$ and e' better than e (for agent i), we have $t(e') < t(e)$.*

Proof. Since item e' is better than e for agent i, and agent i will start eating e only when e' has been fully consumed. Hence we have $\mathsf{begin}(i,e) \ge \mathsf{end}(e')$. By Corollary 4.11.1, we have $t(e') < \mathsf{end}(e') \le \mathsf{begin}(i,e) \le t(e)$. □

Theorem 4.13. *We can compute in polynomial time a lottery that guarantees ex-ante WEF and ex-post WEF*$(1,1)$.

Proof. We first run Algorithm 2 to obtain the fractional allocation $\mathbf{X}$, which by Lemma 4.2 is WEF. Thus we have ex-ante WEF. Then we use the decomposition stated in Lemma 4.6 to obtain a collection of integral allocations. It remains to show that every integral allocation $\mathbf{Y}$ in the decomposition is WEF$(1,1)$. We prove it by constructing a forward sequence σ such that the output of the reversed picking sequence algorithm with σ is exactly $\mathbf{Y}$, and showing that σ satisfies the condition stated in Corollary 3.9.2.

First, we define the sequence σ. We sort the items by their realized times, with ties broken arbitrarily. By re-indexing the items, we assume that

$$t(e_1) \ge t(e_2) \ge \cdots \ge t(e_m). \tag{2}$$

For each $z = 1,\dots,m$, let $\sigma(z) = i$, where i is the agent who receives item e_z in $\mathbf{Y}$, i.e., $e_z \in Y_i$. In other words, for each agent i, the positions of items in Y_i in the ordering in Equation (2) will decide when agent i gets to pick items in the algorithm.

Claim 4.1. *The output of the reversed picking sequence algorithm with* σ *is exactly* $\mathbf{Y}$.

Proof. Recall that in the reversed picking sequence algorithm, the agents take turns to pick their most preferred unallocated item in the order of $(\sigma(m), \sigma(m-1), \dots, \sigma(1))$. We prove by induction on $z = m, m-1, \dots, 1$ that when agent $\sigma(z)$ gets to pick an item, e_z is her most preferred unallocated item (and thus will be chosen).

For $z = m$, when agent $i = \sigma(m)$ gets to pick items, all items are available, and item e_m is her most preferred item because otherwise there exists e that is better than e_m for agent i. Since $e_m \in Y_i$, we can use Lemma 4.12 to show that $t(e) < t(e_m)$, which is a contradiction. Assuming the statement is true for $z = m, m-1, \dots, l+1$, where $l \ge 1$, we show that it is also true for $z = l$. First of all, item e_l must be available when agent $i = \sigma(l)$ gets to pick items by the inductive hypothesis: the set of items allocated is $\{e_m, e_{m-1}, \dots, e_{l+1}\}$. Assume otherwise that there exists item e that is better than e_l for agent i. By Lemma 4.12 we have $t(e) < t(e_l)$, which implies that e should have been allocated already, and is a contradiction. □

Claim 4.2. *For all* $t \in \{1,2,\dots,m\}$, *we have* $s_i(t) - \frac{1}{w_i} \le s_j(t) + \frac{1}{w_j}$, *where*

$$s_i(t) = \frac{|\{t' < t : \sigma(t') = i\}|}{w_i} \qquad \text{and} \qquad s_j(t) = \frac{|\{t' < t : \sigma(t') = j\}|}{w_j}$$

are the weighted numbers of appearances of agents i *and* j *in the prefix* $\{\sigma(1),\dots,\sigma(t)\}$.

Proof. Fix any t. If $s_j(t) = |Y_j|/w_j$, then by Lemma 4.6 we have $w_j \cdot s_j(t) = |Y_j| = |Y_j \cap L(i,0)| \ge \lfloor w_j \cdot m \rfloor$, which implies

$$s_j(t) + \frac{1}{w_j} \ge \frac{\lfloor w_j \cdot m \rfloor + 1}{w_j} > m.$$

Similarly we use Lemma 4.6 to prove that $s_i(t) - \frac{1}{w_i} \le s_i(m) - \frac{1}{w_i} < m$ and the claim follows.

Otherwise j appears at least $(w_j \cdot s_j(t) + 1)$ times in σ. Let e^i and e^j be the corresponding items selected by i at her $(w_i \cdot s_i(t))$-th occurrence in σ and by j at her $(w_j \cdot s_j(t)+1)$-th occurrence in σ, respectively. By the definition of σ, we have $t(e^i) \ge t(e^j)$. Note that $|Y_i \cap L(i, \mathsf{begin}(i,e^i))| = w_i \cdot s_i(t)$ and $|Y_j \cap L(j, \mathsf{begin}(j,e^j))| = w_i \cdot s_j(t) + 1$. Following Lemma 4.10, we have

$$m - \frac{w_j \cdot s_j(t) + 1}{w_j} \le t(e^j) \le t(e^i) < m - \frac{w_i \cdot s_i(t) - 1}{w_i},$$

which implies $s_i(t) - \frac{1}{w_i} < s_j(t) + \frac{1}{w_j}$, as we claimed. □

The above two claims and Corollary 3.9.2 imply that each realized allocation $\mathbf{Y}$ is WEF$(1,1)$. Therefore ex-post WEF$(1,1)$ is guaranteed.

Finally, we remark that the lottery runs in polynomial time since DSE can be implemented in polynomial time and the decomposition can be obtained in strongly polynomial time (see Theorem 4.7). □

5. WEF1 and PO allocations for bi-valued instances

In this section, we focus on the computation of allocations that are fair and efficient, and explore the existence of WEF1 and PO allocations. Garg et al. [29] and Ebadian et al. [30] show that EF1 and PO allocations exist for bi-valued instances (see below for the definition) when agents have equal weights. In this section, we prove a more general result that WEF1 and PO allocations always exist and can be computed efficiently for bi-valued instances, using a similar proof framework.

We first give the definition of bi-valued instances.

Definition 5.1 *(Bi-valued Instances).* An instance is called *bi-valued* if there exist constants $a, b \geq 0$ such that for any agent $i \in N$ and item $e \in M$ we have $c_i(e) \in \{a, b\}$.

For non-zero a, b with $a \neq b$,[5] we can scale the cost functions so that $c_i(e) = \{1, k\}$ for some $k > 1$. If $c_i(e) = k$ we call item e *large* to agent i; otherwise we call it *small* to i. Moreover, if there exists an agent $i \in N$ such that $c_i(e) = k$ for all $e \in M$, we can rescale the costs so that $c_i(e) = 1$ for all $e \in M$. Hence we can assume w.l.o.g. that for all $i \in N$, there exists at least one item $e \in M$ such that $c_i(e) = 1$.

In this section, we prove the following main result.

Theorem 5.2. *There exists an algorithm that computes a WEF1 and PO allocation for any given bi-valued instance in polynomial time.*

We classify the items into two groups depending on their costs as follows.

Definition 5.3 *(Item Groups).* We call item $e \in M$ a *consistently large* item if for all $i \in N, c_i(e) = k$. Let M^+ include all consistently large items, and M^- contain the other items:

$$M^+ = \{e \in M : \forall i \in N, c_i(e) = k\}, \quad M^- = \{e \in M : \exists i \in N, c_i(e) = 1\}.$$

Fisher market. In the Fisher market, there is a *price* vector $\mathbf{p}$ that assigns each chore $e \in M$ a price $p(e) > 0$. For any subset $X_i \subseteq M$, let $p(X_i) = \sum_{e \in X_i} p(e)$. Given the price vector $\mathbf{p}$, we define the *pain-per-buck* ratio $\alpha_{i,e}$ of agent i for chore e to be $\alpha_{i,e} = c_i(e)/p(e)$, and the *minimum pain-per-buck* (MPB) ratio α_i of agent i to be $\alpha_i = \min_{e \in M}\{\alpha_{i,e}\}$. For each agent i, we define $\mathsf{MPB}_i = \{e \in M : \alpha_{i,e} = \alpha_i\}$, and we call each item $e \in \mathsf{MPB}_i$ an MPB item of agent i. An allocation $\mathbf{X}$ with price $\mathbf{p}$ forms a (Fisher market) equilibrium $(\mathbf{X}, \mathbf{p})$ if each agent only receives her MPB chores, i.e. $X_i \subseteq \mathsf{MPB}_i$ for any $i \in N$.

Definition 5.4 *(pWEF1).* An equilibrium $(\mathbf{X}, \mathbf{p})$ is called *price weighted envy-free up to one item* (pWEF1) if for any $i, j \in N$, there exists an item $e \in X_i$ such that

$$\frac{p(X_i - e)}{w_i} \leq \frac{p(X_j)}{w_j}.$$

Throughout this section, we call $p(X_i)/w_i$ the (weighted) spending of agent i. For convenience of notation, given an equilibrium $(\mathbf{X}, \mathbf{p})$, we use $\hat{p}_i$ to denote the (weighted) spending of agent i after removing the item with maximum price, i.e.,

$$\hat{p}_i = \min_{e \in X_i} \left\{ \frac{p(X_i - e)}{w_i} \right\}.$$

In the following, we say that agent i *strongly envies* agent j if $\hat{p}_i > \frac{p(X_j)}{w_j}$. Note that the equilibrium is pWEF1 if and only if no agent strongly envies another agent.

Lemma 5.5. *If an equilibrium* $(\mathbf{X}, \mathbf{p})$ *is pWEF1, the allocation* $\mathbf{X}$ *is WEF1 and PO.*

Proof. We first show that the allocation $\mathbf{X}$ is Pareto optimal. If the allocation $\mathbf{X}$ with price $\mathbf{p}$ is an equilibrium, then for any agent $i \in N$, any item $e \in X_i$ and any $j \neq i$ we have

$$\frac{c_i(e)}{\alpha_i} = \frac{c_i(e)}{\alpha_{i,e}} = p(e) = \frac{c_j(e)}{\alpha_{j,e}} \leq \frac{c_j(e)}{\alpha_j}.$$

Thus the allocation minimizes the objective $\sum_{i \in N} \frac{c_i(X_i)}{\alpha_i}$. Any Pareto improvement would strictly decrease this objective, which leads to a contradiction. So the allocation $\mathbf{X}$ is PO.

Next, we show that the allocation $\mathbf{X}$ is WEF1. Since the equilibrium $(\mathbf{X}, \mathbf{p})$ is pWEF1, for any agents $i, j \in N$, there exists an item $e \in X_i$ such that $\frac{p(X_i - e)}{w_i} \leq \frac{p(X_j)}{w_j}$. Note that both agents i and j only receive items holding the MPB ratio since $(\mathbf{X}, \mathbf{p})$ is an equilibrium. Then we have

[5] When one of a, b is 0, the instances reduce to the *binary instances*, in which case WEF1 and PO allocations can be trivially computed by first assigning chores to agents having 0 cost on the items, and then allocating the remaining chores (which cost 1 to all agents) following the reversed weighted picking sequence.

$$\frac{c_i(X_i - e)}{w_i} = \alpha_i \cdot \frac{p(X_i - e)}{w_i} \le \alpha_i \cdot \frac{p(X_j)}{w_j} \le \frac{c_i(X_j)}{w_j},$$

where last inequality holds since the pain-per-buck ratio of agent i on any item is at least α_i. $\square$

Definition 5.6 *(Big and Least Spenders).* Given an equilibrium $(\mathbf{X}, \mathbf{p})$, an agent $b \in N$ is called a big spender if $b = \arg\max_{i \in N} \{\hat{p}_i\}$; an agent l is called a least spender if $l = \arg\min_{i \in N} \left\{ \frac{p(X_i)}{w_i} \right\}$.[6]

We show that if the big spender does not strongly envy the least spender, then the equilibrium is pWEF1.

Lemma 5.7. *If an equilibrium $(\mathbf{X}, \mathbf{p})$ holds that the big spender b does not strongly envy the least spender l, then the equilibrium is pWEF1.*

Proof. For any agent $i, j \in N$, we show that agent i does not strongly envy agent j. Note that b is the big spender and l is the least spender. From the definitions we have

$$\min_{e \in X_i} \left\{ \frac{p(X_i - e)}{w_i} \right\} = \hat{p}_i \le \hat{p}_b = \min_{e \in X_b} \left\{ \frac{p(X_b - e)}{w_b} \right\} \quad \text{and} \quad \frac{p(X_l)}{w_l} \le \frac{p(X_j)}{w_j}.$$

Recall that the big spender b does not strongly envy the least spender l. In other words, we have

$$\min_{e \in X_b} \left\{ \frac{p(X_b - e)}{w_b} \right\} \le \frac{p(X_l)}{w_l}.$$

Hence we have $\min_{e \in X_i} \left\{ \frac{p(X_i - e)}{w_i} \right\} \le \frac{p(X_j)}{w_j}$, and thus $(\mathbf{X}, \mathbf{p})$ is pWEF1. $\square$

Given the above lemma, to see if $(\mathbf{X}, \mathbf{p})$ is pWEF1, it suffices to consider the envy from the big spender b to the least spender l. If b strongly envies l, then we try to reallocate some items from X_b to X_l, and possibly update the price of some items, while ensuring that the resulting allocation and price form a new equilibrium. We show that such reallocations are always possible, and by polynomially many reallocations, we can eliminate the envy from the big spender to the least spender. To begin with, we first compute an initial equilibrium $(\mathbf{X}^0, \mathbf{p}^0)$, based on which we partition the agents into groups.

5.1. Initial equilibrium and agent groups

In the following, we give an algorithm (Algorithm 3) that computes an initial equilibrium $(\mathbf{X}, \mathbf{p})$ and agent groups $\{N_r\}_{r \in [R]}$ with some useful properties (see Lemma 5.9). Our algorithm follows a framework that is similar to that of Zhou and Wu [13], and thus is slightly different compared with that of Garg et al. [29].

Algorithm 3: Compute Initial Price, Allocation and Agent Groups.

Input: A bi-valued instance $\langle M, N, \mathbf{w}, \mathbf{c} \rangle$
initialize $X_i \leftarrow \varnothing$ for all $i \in N$, $P \leftarrow M$;
set $p(e) = 1$ for all $e \in M^-$ and $p(e) = k$ for all $e \in M^+$;
`// Phase 1: Computation of the Initial Allocation`
compute a social cost minimizing allocation $\mathbf{X}$;
construct a direct graph $G_X = (N, E)$: add an MPB edge from j to i if $X_i \cap \mathsf{MPB}_j \ne \varnothing$;
while *there exists a path* $i_\tau \to \cdots \to i_0$ *such that* $\hat{p}_{i_0} > \frac{p(X_{i_\tau})}{w_{i_\tau}}$ **do**
    for $l = 1, 2, \cdots, \tau$ **do**
        pick an item $e \in X_{i_{l-1}} \cap \mathsf{MPB}_{i_l}$, breaking ties by picking the item with the maximum price ;
        update $X_{i_{l-1}} \leftarrow X_{i_{l-1}} - e$ and $X_{i_l} \leftarrow X_{i_l} + e$;
`// Phase 2: Computation of Agent Groups`
initialize $R \leftarrow 0, N' \leftarrow N$;
while $N' \ne \varnothing$ **do**
    let $b \leftarrow \mathrm{argmax}_{i \in N'} \{\hat{p}_i\}$, breaking ties by picking the agent with smallest index ;
    update $R \leftarrow R + 1$ and let $N_R \leftarrow \{b\} \cup \{i \in N' :$ exists an MPB path from i to b in $G_X\}$;
    $N' \leftarrow N' \setminus N_R$;
Output: $\mathbf{X} = \{X_1, \cdots, X_n\}$, $\mathbf{p}$, $\{N_r\}_{r \in [R]} = \{N_1, \cdots, N_R\}$.

[6] Throughout the whole paper, when selecting a big spender we break ties by picking the agent with the smallest index. The same principle applies to the selection of least spender. Therefore, in the rest of the paper, we call b (resp. l) "the" big (resp. least) spender when this tie-breaking rule is applied.

Table 3
Agents' Cost Functions.

	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	e_9
agent 1	1	1	1	1	1	1	5	5	5
agent 2	5	5	5	5	5	1	1	1	5
agent 3	5	5	5	5	5	5	5	1	1

The initial price. We set the price vector $\mathbf{p}$ as $p(e) = \min_{i\in N}\{c_i(e)\}$. Therefore, we have $p(e) = 1$ for all $e \in M^-$ and $p(e) = k$ for all $e \in M^+$.

The allocation. We first compute an allocation $\mathbf{X}$ that minimizes the social cost: for each $e \in M^-$, we allocate it to an arbitrary agent i with $c_i(e) = 1$; items in M^+ are allocated arbitrarily. Then we construct a directed graph G_X based on $\mathbf{X}$ as follows. For each pair of $i, j \in N$, if there exists an item $e \in X_i$ such that $e \in \mathsf{MPB}_j$, then we create an MPB edge from j to i. While there exists a path from an agent j to another agent i such that $\hat{p}_i > \frac{p(X_j)}{w_j}$, we implement a sequence of item transfers backward along the path. When there is no path of such type, we finish the computation of the initial allocation $\mathbf{X}$.

The agent groups. We select the big spender b_1 and obtain the agents that can reach b_1 via MPB paths. We add these agents to the first agent group N_1, together with agent b_1. After identifying group N_1, we repeat the above procedure by picking the big spender b_2 among the remaining agents $N \setminus N_1$ and letting N_2 contain b_2 and the agents that can reach b_2 via MPB paths. Recursively, we partition agents into groups $(N_1, N_2, \cdots, N_R)$. We call N_1 the *highest* group and N_R the *lowest*. By construction, each group N_r has a *representative* agent b_r, to which every agent in $N_r \setminus \{b_r\}$ has an MPB path.

Here we present an instance I^*, which we use to illustrate the execution of our algorithms throughout the whole section.

Example 5.8. Consider running Algorithm 3 on instance $I^* = \langle N, M, \mathbf{w}, \mathbf{c}\rangle$ where $N = \{1,2,3\}$, $M = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9\}$ and $w_1 = \frac{1}{2}$, $w_2 = \frac{1}{3}$, $w_3 = \frac{1}{6}$. All the cost functions are additive and presented in Table 3.

Initially, we have the price vector $\mathbf{p}$ being $p(e) = 1$ for all $e \in M$. We assume that the initial allocation is $X_1 = \{e_1, e_2, e_3, e_4, e_5, e_6\}$, $X_2 = \{e_7, e_8\}$, $X_3 = \{e_9\}$. We observe that there exists a path $2 \to 1$ and we can transfer item e_6 to agent 2. The allocation returned by Algorithm 3 is $X_1 = \{e_1, e_2, e_3, e_4, e_5\}$, $X_2 = \{e_6, e_7, e_8\}$, $X_3 = \{e_9\}$ and each agent belongs to an unique agent group: $N_1 = \{1\}$, $N_2 = \{2\}$, $N_3 = \{3\}$.

We say that a group N_r is pWEF1 if all agents in N_r are pWEF1 towards each other.

Lemma 5.9. *Algorithm 3 returns an allocation $\mathbf{X}$ with price $\mathbf{p}$ and agent groups $\{N_r\}_{r\in[R]}$ with the following properties*

1. *$(\mathbf{X}, \mathbf{p})$ is an equilibrium, and $\alpha_i = 1$ for all $i \in N$.*
2. *For all $i, j \in N$, if j is from a group lower than i (i.e., there exist $r < r'$ such that $i \in N_r, j \in N_{r'}$) then for all $e \in X_i$ we have $c_j(e) = k$.*
3. *All consistently large items are allocated to the lowest group: $M^+ \subseteq \bigcup_{i\in N_R} X_i$.*
4. *For all $r \in [R]$, the agent group N_r is pWEF1.*

Proof. For property 1: by the way we set the initial price, we have $\alpha_i = \min_{e\in M}\left\{\frac{c_i(e)}{p(e)}\right\} = 1$ for all $i \in N$. Since we start from the social cost minimizing allocation (in which $X_i \subseteq \mathsf{MPB}_i$ for all $i \in N$) and reallocate an item e to agent i only if $e \in \mathsf{MPB}_i$, we can ensure that $X_i \subseteq \mathsf{MPB}_i$ for all $i \in N$ in the final allocation. Hence $(\mathbf{X}, \mathbf{p})$ is an equilibrium.

For property 2: suppose that there exists $e \in X_i$ such that $c_j(e) = 1$. Then we have $e \in \mathsf{MPB}_j$ and there is an MPB edge from j to i. Let b_r be the representative agent of group N_r. By the construction of group N_r, i can reach b_r via an MPB path. Hence j can also reach b_r via an MPB path, which is a contradiction because it implies that j should be included into N_r.

For property 3: following a similar argument, suppose that there exists an item $e \in M^+$ that is allocated to an agent $i \in N_r$ with $r < R$. Then every agent $j \in N_R$ has an MPB edge to i since $e \in \mathsf{MPB}_j$. Thus j can reach the representative agent b_r of N_r, which leads to a contradiction since $j \notin N_r$.

For property 4: recall that every agent $i \in N_r$ can reach the representative b_r via an MPB path, and $\hat{p}_{b_r} \geq \hat{p}_i$. Moreover, we have $\hat{p}_{b_r} \leq \frac{p(X_i)}{w_i}$ because otherwise the path from i to b_r should have been resolved in the first phase of Algorithm 3 when $\mathbf{X}$ is computed.

Hence for all $i, j \in N_r$, we have $\hat{p}_i \leq \hat{p}_{b_r} \leq \frac{p(X_j)}{w_j}$. Thus i is pWEF1 toward j. $\square$

Lemma 5.10. *Algorithm 3 runs in polynomial time.*

Proof. Observe that (1) the computation of the initial price $\mathbf{p}$ takes $O(m)$ time; (2) computing the social cost minimizing allocation takes $O(nm)$ time; (3) in Phase 2, given the initial allocation, computing the agent groups takes $O(nm)$ time because each group can be computed by identifying a connected component of G_X. Therefore, to prove that the algorithm runs in polynomial time, it suffices to argue that the while loops in lines 5 - 8 of Algorithm 3 finish in polynomial time. Observe that computing G_X takes $O(nm)$ time and resolving an MPB path takes $O(m)$ time.

In the following, we show that by carefully choosing the MPB paths to resolve, the while loops break after $O(knm)$ rounds. Recall that in each while loop, we identify a path $i_\tau \to \cdots \to i_0$ with $\hat{p}_{i_0} > p(X_{i_\tau})/w_{i_\tau}$, and resolve the path by transferring an MPB item from agent i_{l-1} to i_l, for all $l = 1, 2, \ldots, \tau$. We call i_τ the start-agent of the path and i_0 the end-agent. When there exist multiple such paths, we choose the one that maximizes $\hat{p}_{i_0}$. Observe that such a path can be identified in $O(nm)$ time. We refer to a while loop as a round, and index the rounds by $t = 1, 2, \ldots$. We use X_i^t to denote the bundle of agent i at the beginning of round t. Likewise, we define $\hat{p}_i^t$ and the path $i_k^t \to \cdots \to i_0^t$ at the beginning of round t.

Claim 5.1. *We have $\hat{p}^0_{i_0^0} \ge \hat{p}^1_{i_0^1} \ge \cdots \ge \hat{p}^t_{i_0^t}$.*

Proof. Consider a round t in which path $i_\tau^t \to \cdots \to i_0^t$ is chosen. We show that for all agents i_l^t, where $l \neq \tau$, her spending would not increase after the item transfers in this round, i.e., $p(X^t_{i_l^t}) \ge p(X^{t+1}_{i_l^t})$. Assume otherwise, then it must be that i_l^t receives some item e with $p(e) = k$ from i_{l-1}^t and i_l^t transfers an item e' with $p(e') = 1$ to agent i_{l+1}^t. However, this is impossible because the consistently large item e is an MPB item to all agents, which should be transferred to agent i_{l+1}^t as we break ties by choosing the item with maximum price. Hence, in each round, only the spending of the start-agent would increase. This implies $\hat{p}^{t+1}_{i_0^{t+1}} \le \hat{p}^t_{i_0^t}$ because at the beginning of round $t+1$, compared to round t, the only agent whose spending is increased is i_τ^t, and its current spending is $\hat{p}^{t+1}_{i_\tau^t} \le p(X^t_{i_\tau^t})/w_{i_\tau^t} < \hat{p}^t_{i_0^t}$. ☐

We further show that if an agent is identified as an end agent twice, then her spending (up to the removal of one item) is strictly smaller at its second appearance.

Claim 5.2. *If agent i is the end-agent in rounds t_1 and t_2, where $t_1 < t_2$, then we have $\hat{p}_i^{t_1} > \hat{p}_i^{t_2}$.*

Proof. Assume otherwise, i.e., $\hat{p}_i^{t_1} = \hat{p}_i^{t_2}$, then i must receive some item as a start-agent in some rounds between t_1 and t_2. Let $t < t_2$ be the last round in which i is a start-agent. Let j be the corresponding end-agent. We have $\hat{p}_j^t > \frac{p(X_i^t)}{w_i}$. Since t is the last round before t_2 in which i's spending increases, we have $\hat{p}_i^{t_2} \le \frac{p(X_i^t)}{w_i} < \hat{p}_j^t$, which is a contradiction because from Claim 5.1, we have $\hat{p}_i^{t_2} = \hat{p}_i^{t_1} \ge \hat{p}_j^t$. ☐

Note that for each agent i, the value of $\hat{p}_i$ is at most $\frac{km-1}{w_i}$. As argued above, each agent can be identified as an end-agent at most km times because each of its appearances decreases $\hat{p}_i$ by at least $\frac{1}{w_i}$. Hence the total number of rounds is at most knm. Hence Algorithm 3 finishes in $O(kn^2m^2)$ time. ☐

5.2. The allocation algorithm and the invariants

In this section, we present an algorithm that starts from the initial equilibrium (denoted by $(\mathbf{X}^0, \mathbf{p}^0)$) and constructs a pWEF1 equilibrium $(\mathbf{X}, \mathbf{p})$ by a sequence of item reallocations and price raises. The algorithm proceeds in rounds. In each round, we identify the big spender b and least spender l.

- If b is pWEF1 towards l then the algorithm terminates and outputs the allocation. By Lemma 5.7, we can guarantee that the output equilibrium is pWEF1.
- Otherwise we reallocate an item from b to l following an MPB path, during which we may raise the price of all items owned by agents from the group containing b by a factor of k.

Before we present the full details of the algorithm, we remark that throughout the whole process, the following invariants are always maintained. We would also like to remark that this is where our analysis deviates from that of Garg et al. [29]: their analysis depends on the relationship between bundle price and bundle size, which is no longer helpful when agents have arbitrary weights. In the weighted case, we need a more direct characterization of the bundle prices. This leads us to adopt a more general perspective on the properties maintained by the algorithm and summarize them into several intuitive and useful invariants that are maintained throughout the whole algorithm. As a consequence, our analysis does not need to break the algorithm into two different phases. We also believe that these invariants can help to provide a better understanding of the algorithm and make the proofs more well-structured and easier to follow.

For convenience, we use $N_{\le i}$ to denote $\bigcup_{j \le i} N_j$. Likewise, we define $N_{<i}, N_{\ge i}$ and $N_{>i}$.

Invariant 5.11 *(Equilibrium Invariant).* At any point in time, $(\mathbf{X}, \mathbf{p})$ is an equilibrium.

Invariant 5.12 *(pWEF1 Invariant)*. For all $r \in [R]$, group N_r is pWEF1 in equilibrium $(\mathbf{X}, \mathbf{p})$.

Invariant 5.13 *(Raised Group Invariant)*. There exists $r^* \in [R]$ such that all groups $N_1, \dots, N_{r^*-1}$ are raised exactly once; all groups $N_{r^*}, \dots, N_R$ are not raised. Moreover

- for all $i \in N_{<r^*}$ we have $\alpha_i = 1/k$ and $X_i \subseteq X_i^0$, i.e., agent i did not receive any new item;
- for all $i \in N_{\geq r^*}$ we have $\alpha_i = 1$ and $X_i^0 \subseteq X_i$, i.e., agent i did not lose any item.

Note that both invariants hold true at the beginning of the algorithm when $(\mathbf{X}, \mathbf{p}) = (\mathbf{X}^0, \mathbf{p}^0)$ and all groups are not raised (e.g., $r^* = 1$), by Lemma 5.9. Moreover, Invariant 5.13 implies that the last group N_R is never raised because $R \geq r^*$ for all $r^* \in [R]$, which further implies that $M^+ \subseteq \bigcup_{i \in N_R} X_i$ because the unraised agents did not lose any item.

Corollary 5.13.1. *Group N_R is not raised, and $M^+ \subseteq \bigcup_{i \in N_R} X_i$.*

Next, we present the full details of the algorithm (see Algorithm 4 for the pseudo-code).

The full algorithm. We first call Algorithm 3, which returns the initial equilibrium $(\mathbf{X}, \mathbf{p})$, and modify the equilibrium until it becomes pWEF1. For analysis purposes, we use $(\mathbf{X}^0, \mathbf{p}^0)$ to denote this initial equilibrium. In each round (of the while loop), we identify the big spender b and the least spender l. If b is pWEF1 toward l then we output $(\mathbf{X}, \mathbf{p})$ and terminate (Lemma 5.7 ensures that the equilibrium is pWEF1). Otherwise, we check whether l is raised. If l is not raised, we raise the group containing b if this group has not been raised, and allocate an item from b to l (we show that all items in X_b are MPB items to l). If l is raised, we show (see Lemma 5.20) that b must be raised already, and there exists an unraised agent i that received item from l in some past round. Then we return the item we have reallocated from agent l to agent i back to X_l, and reallocate an item from b to l.

Algorithm 4: Find a WEF1 and PO allocation.

Input: A bi-valued instance $\langle M, N, \mathbf{w}, \mathbf{c} \rangle$

let $(\mathbf{X}, \mathbf{p}, \{N_r\}_{r \in [R]})$ be returned by Algorithm 3 ; `// Denote the equilibrium by` $(\mathbf{X}^0, \mathbf{p}^0)$
initialize the set of unraised agents $U \leftarrow N$;
while *True* **do**
  let $b \leftarrow \arg\max_{i \in N} \{\hat{p}_i\}$ be the big spender ;
  let $l \leftarrow \arg\min_{i \in N} \left\{ \frac{p(X_i)}{w_i} \right\}$ be the least spender ;
  if $\hat{p}_b \leq \frac{p(X_l)}{w_l}$ **then**
    output $(\mathbf{X}, \mathbf{p})$ and **terminate**.
  suppose that $b \in N_r$ and $l \in N_{r'}$; `// we have` $r \neq r'$ `due to Invariant 5.12`
  if $l \in U$ **then**
    if $b \in U$ **then**
      raise the price of all chores in $\bigcup_{i \in N_r} X_i$ by a factor of k ;
      $U \leftarrow U \setminus N_r$;
    pick an arbitrary $e \in X_b$; `// we have` $X_b \subseteq \mathsf{MPB}_l$, `see Lemma 5.19`
    update $X_b \leftarrow X_b - e$, $X_l \leftarrow X_l + e$;
  else
    we have $b \in N \setminus U$ and $\exists i \in U$ with $X_i \cap X_l^0 \neq \emptyset$; `// see Lemma 5.20`
    pick any $e_1 \in X_b \cap \mathsf{MPB}_i$ and $e_2 \in X_i \cap X_l^0$;
    update $X_b \leftarrow X_b - e_1$, $X_l \leftarrow X_l + e_2$ and $X_i \leftarrow X_i + e_1 - e_2$;

Example 5.14. Consider running Algorithm 4 on instance I^*. We remark that Algorithm 4 starts by calling Algorithm 3 and $\mathbf{X}^0 = (X_1^0, X_2^0, X_3^0)$ where $X_1^0 = \{e_1, e_2, e_3, e_4, e_5\}$, $X_2^0 = \{e_6, e_7, e_8\}$, $X_3^0 = \{e_9\}$ is returned. We see that the allocation $\mathbf{X}^0$ is not pWEF1 since agent 1 is not pWEF1 towards agent 3. The Algorithm 4 raises the prices of the chores in X_1^0 by a factor of 5 and transfers a chore to agents 2, 3, i.e.,

$$X_1^1 = \{e_1, e_2, e_3\}, X_2^1 = \{e_5, e_6, e_7, e_8\}, X_3^1 = \{e_4, e_9\}$$

It can be verified that the allocation is pWEF1 (and thus WEF1).

5.3. Properties of the algorithm

Notations. In the following, we refer to a while loop as a round, and index the rounds by $t = 0, 1, 2, \ldots$. We use $\mathbf{X}^t$ and $\mathbf{p}^t$ to denote the allocation and the price at the beginning of round t.[7] We use b^t and l^t to denote the big spender and the least spender, respectively, we identify at the beginning of round t. Note that $\mathbf{X}^{t+1}$ and $\mathbf{p}^{t+1}$ are the allocation and the price at the end of round t.

In addition to the invariants we have introduced, we introduce a few more invariants that involve multiple rounds. It is trivial to check that these invariants hold true at the beginning of the first round (when $t = 0$).

Invariant 5.15 *(Least Spending Invariant).* The spending of the least spender across different rounds is non-decreasing: $\frac{p^0(X^0_{l^0})}{w_{l^0}} \le \frac{p^1(X^1_{l^1})}{w_{l^1}} \le \cdots \le \frac{p^t(X^t_{l^t})}{w_{l^t}}$.

Invariant 5.16 *(Big Spender Invariant).* The big spender b^t identified at the beginning of round t has never been identified as a least spender in rounds $1, 2, \ldots, t$. Moreover, if b has not been raised, i.e., $b \in N_r$ for some $r \ge r^*$, then this property holds true for all $i \in N_r$.

Invariant 5.17 *(Price Invariant).* In each round, the spending of agent i does not change, unless i is raised during this round or i is the big or least spender in this round. Moreover, for all $e \in M$, $p(e) \in \{1, k\}$.

In the following, we fix some round t, and assume that at the beginning of round t, the Equilibrium Invariant (Invariant 5.11), the pWEF1 Invariant (Invariant 5.12), the Raised Group Invariant (Invariant 5.13), the Least Spending Invariant (Invariant 5.15), the Big Spender Invariant (Invariant 5.16) and the Price Invariant (Invariant 5.17) are maintained. Recall that all invariants hold true at the beginning of round 0. Moreover, we assume the algorithm is well-defined thus far. We assume that b^t strongly envies l^t (otherwise the algorithm terminates). We show several properties of the algorithm, which will be used to show that the invariants are maintained at the end of round t. When the context is clear, we drop the superscript t for ease of notation. We first show a few properties regarding the big spender b and least spender l in round t, which show that the algorithm is well defined. Let r^* be defined as in Invariant 5.13, i.e., all agents in $N_{<r^*}$ are raised exactly once and all agents in $N_{\ge r^*}$ are not raised. Using Invariant 5.16, we prove that if the big spender b has not been raised, then b must be in the unraised group with smallest index. This would be helpful in showing that Invariant 5.13 is maintained at the end of round t: when we raise the group containing b, we need to ensure that the raised group is N_{r^*}.

Lemma 5.18. *If $b \in U$, then we have $b \in N_{r^*}$.*

Proof. By Invariant 5.13, for all unraised agent $i \in U$, we have $X_i^0 \subseteq X_i$. That is, agent i did not lose any item. Moreover, by the design of the algorithm, X_i^0 is a proper subset of X_i if and only if i was identified as a least spender in some round before t.

Suppose that $b \in N_r$ for some $r \ge r^* + 1$. By Invariant 5.16, agent b has never been identified as a least spender, which (together with $b \in U$) implies that $X_b = X_b^0$ and $\hat{p}_b = \hat{p}_b^0$. Let $b' = \arg\max_{i \in N_{r^*}} \{\hat{p}_i^0\}$. Note that $\hat{p}_{b'} \ge \hat{p}_{b'}^0$ because b' did not lose any item. Since b is the big spender, we have

$$\hat{p}_b^0 = \hat{p}_b \ge \hat{p}_{b'} \ge \hat{p}_{b'}^0 \ge \hat{p}_b^0,$$

where the last inequality holds due to the principle for computing the initial agent groups. Therefore, all inequalities should hold with equality, which leads to a contradiction because b' should be the big spender in round t (recall that we use the same rule for breaking ties, i.e., by selecting the agent with minimum index, every time we select a big spender). □

The above lemma implies the following almost straightforwardly.

Lemma 5.19. *If $l \in U$, then we have $X_b \subseteq \mathsf{MPB}_l$ in line 13.*

Proof. Suppose that $l \in N_{r'}$ and $b \in N_r$. Since b strongly envies l, by Invariant 5.12, we know that b and l are from different groups. Additionally, by Lemma 5.18, we have $r \le r^* < r'$. By Invariant 5.16, we have $X_b \subseteq X_b^0$. Then by the property of agent groups (see Lemma 5.9), we have $c_l(e) = k$ for all $e \in X_b$. Since in line 13, agent b has already been raised, for all $e \in X_b$ we have $p(e) = k$, which implies that $e \in \mathsf{MPB}_l$ because $\alpha_l = 1$ (by Invariant 5.13). □

The next lemma shows that line 16 of Algorithm 4 is well defined.

[7] Note that the notations are consistent with the notation for the initial equilibrium $(\mathbf{X}^0, \mathbf{p}^0)$ returned by Algorithm 3.

Lemma 5.20. *If l has been raised, then b has also been raised, and there exists an unraised agent $i \in U$ that received some item from l in some round before t.*

Proof. We first show that X_l is a proper subset of X_l^0. By Invariant 5.13, since l is raised, we have $X_l \subseteq X_l^0$. Suppose $X_l = X_l^0$, then either (1) l has never been identified as a big spender (and thus did not lose any item); or (2) l was identified as a big spender and lost some item, but was identified as a (raised) least spender later and retrieved the lost item.

For case (1), let $t' < t$ be the round in which l is raised. By assumption l is in the same group as $b^{t'}$ and thus $b^{t'}$ is pWEF1 towards l in $(\mathbf{X}^{t'}, \mathbf{p}^{t'})$, by Invariant 5.12. Thus we have

$$\frac{p(X_l)}{w_l} \geq \frac{k \cdot p^{t'}(X_l^{t'})}{w_l} \geq k \cdot \hat{p}^{t'}_{b^{t'}} \geq k \cdot \hat{p}^{t'}_b \geq \hat{p}_b,$$

where the first inequality holds because l did not lose any item, and l is raised during round t'; the second inequality holds because $b^{t'}$ is pWEF1 toward l in $(\mathbf{X}^{t'}, \mathbf{p}^{t'})$; the third inequality holds because $b^{t'}$ is the big spender at the beginning of round t'; the last inequality holds because b did not receive any item before round t, by Invariant 5.16. Then we have a contradiction because b strongly envies l (in $(\mathbf{X}, \mathbf{p})$).

For case (2), let t_1 be the round during which l is raised and $t_2 \geq t_1$ be the first round in which l is identified as a big spender. By definition of t_2 and Invariant 5.16, l did not lose or receive any item before round t_2. Therefore we have $X_l^{t_1} = X_l^0 = X_l$. Since l is raised during round t_1, following the same analysis as in case (1) we have

$$\frac{p(X_l)}{w_l} = \frac{k \cdot p^{t_1}(X_l^{t_1})}{w_l} \geq k \cdot \hat{p}^{t_1}_{b^{t_1}} \geq k \cdot \hat{p}^{t_1}_b \geq \hat{p}_b,$$

which also contradicts with the fact that b strongly envies l (in $(\mathbf{X}, \mathbf{p})$).

Therefore, l must have lost some item (as a big spender) before round t and the item has not been retrieved. Let $e \in X_l^0 \setminus X_l$ be any such item and suppose $e \in X_i$. Then by Invariant 5.13, i has not been raised when round t begins, because $X_i \not\subseteq X_i^0$. It remains to show that b is raised (when round t begins).

Assume that b has not been raised when round t begins. Let $t' < t$ be the last round during which agent l loses some item as a big spender. Then we have

$$\hat{p}^{t'}_l \leq \frac{p(X)}{w_l} < \hat{p}_b \leq \hat{p}^{t'}_b,$$

where the first inequality holds because t' is the last round during which l loses some item, the second inequality holds because b strongly envies l in $(\mathbf{X}, \mathbf{p})$, the third inequality holds since we assume that b has never been raised, and did not receive any item (by Invariant 5.16). Therefore we have a contradiction because b should be the big spender in round t'. Hence b must have been raised when round t begins, and the proof is complete. □

5.4. Maintenance of invariants

We show in this section that assuming all the stated invariants hold true when round t begins (after identifying the big and least spenders), all of them will continue to hold at the end of round t (assuming that the algorithm does not terminate during round t).

Lemma 5.21. *The Equilibrium Invariant (Invariant 5.11) is maintained when round t ends.*

Proof. To prove that $(\mathbf{X}^{t+1}, \mathbf{p}^{t+1})$ is an equilibrium, we need to argue that

(1) raising the price of a group does not cause $X_i \not\subseteq \mathsf{MPB}_i$ for any $i \in N$.
(2) all reallocations happened during round t follow MPB edges.

For the first property, note that by Invariant 5.16 and 5.13, if agent $i \in N_{r^*}$ is raised during this round, then we have $X_i = X_i^0$ and $p(e) = 1$ for all $e \in X_i$.

- For all $j \in N_{<r^*}$, since $\alpha_j = 1/k$, raising group N_{r^*} does not affect $X_j \subseteq \mathsf{MPB}_j$.
- For all $j \in N_{r^*}$, α_j decreases from 1 to $1/k$ but for all $e \in X_j$ we have $\alpha_{j,e} = 1/k$, which implies $X_j \subseteq \mathsf{MPB}_j$.
- For all $j \in N_{>r^*}$, by Lemma 5.9, we have $c_j(e) = k$ for all item e whose price is raised (from 1 to k). Hence α_j remains 1 and we still have $X_j \subseteq \mathsf{MPB}_j$.

For the second property, we first consider the case when l is not raised. By Lemma 5.19, the item allocated to l is in MPB_l. Now suppose that l is raised. Then by Lemma 5.20, b is also raised, and the algorithm reallocates an item in $e_2 \in X_i \cap X_l^0$ to l and an item e_1 from b to i. By Invariant 5.13, we have $\alpha_b = \alpha_l = 1/k$ and $\alpha_i = 1$. When e_2 was allocated from l to i, $p(e_2)$ has already been raised (from 1 to k). Thus $\alpha_{l,e_2} = \frac{c_l(e_2)}{p(e_2)} = 1/k$, which implies that $e_2 \in \mathsf{MPB}_l$. Since b is in a raised group and i is not raised, we have $c_i(e_1) = k$. Furthermore, since $p(e_1) = k$, we have $\alpha_{i,e_1} = 1$, which implies that $e_1 \in \mathsf{MPB}_i$. □

Lemma 5.22. *The pWEF1 Invariant (Invariant 5.12) is maintained at the end of round t.*

Proof. By Invariant 5.17, during round t, only the price of the big spender b, the least spender l and agents whose price is raised during this round would change. Moreover, raising the price of a whole group does not affect the pWEF1 property within this group. Hence it suffices to consider the case when no agent is raised during this round and argue that the group N_r containing b and the group $N_{r'}$ containing l remain pWEF1 when round t ends. By Invariant 5.12, both groups N_r and N_r' are pWEF1 when round t begins.

- For N_r, since only the spending of b decreases, it suffices to argue that no agent $i \in N_r \setminus \{b\}$ envies b when round t ends. Note that by Invariant 5.17, the spending of i does not change. Since the spending of b at the end of this round is at least $\hat{p}_b$ (since b loses one item in this round), and $\hat{p}_i \le \hat{p}_b$ (since b is the big spender), we conclude that i is pWEF1 towards b when round t ends.
- For $N_{r'}$, since only the spending of l increases, it suffices to argue that agent l does not envy any other agent $i \in N_{r'}$ when round t ends, which trivially holds because agent l has the minimum spending when round t begins, and l receives only one item during this round.

Hence the equilibrium $(\mathbf{X}^{t+1}, \mathbf{p}^{t+1})$ is also pWEF1. □

Lemma 5.23. *The Raised Group Invariant (Invariant 5.13) is maintained at the end of round t.*

Proof. It suffices to consider the case when the group containing b is raised during this round. Recall that by Lemma 5.18, we have $b \in N_{r^*}$. Hence raising N_{r^*} causes r^* to be increased by one, at the end of round t. Note that we must have $r^* \le R-1$ because otherwise ($b \in N_R$) the least spender $l \notin N_R$ is raised, which implies that b is already raised (by Lemma 5.20), and is a contradiction. Moreover, by Invariant 5.16, every agent $i \in N_{r^*}$ did not receive any item, which implies $X_i^{t+1} \subseteq X_i^0$. Since for all $i \in N_{r^*}$ we have $\alpha_i = 1$ when round t begins, we have $\alpha_i = 1/k$ when round t ends. Therefore Invariant 5.13 is maintained. □

Lemma 5.24. *The Least Spending Invariant (Invariant 5.15) is maintained at the end of round t.*

Proof. To show that $\frac{p^{t+1}(X_{l^{t+1}}^{t+1})}{w_{l^{t+1}}} \ge \frac{p(X_l)}{w_l}$, it suffices to argue that the spending of every agent $i \in N$ is at least $\frac{p(X_l)}{w_l}$. By Invariant 5.17, only the spending of b decreases. Since b loses only one item, her spending at the end of round t is at least $\hat{p}_b$. Since b strongly envies l, we have $\hat{p}_b \ge \frac{p(X_l)}{w_l}$, which implies that the spending of b when round t ends is at least $\frac{p(X_l)}{w_l}$. □

Lemma 5.25. *The Big Spender Invariant (Invariant 5.16) is maintained when round t ends.*

Proof. Recall that the big and least spenders in round $t+1$ are decided by the equilibrium we compute at the end of round t. To prove the lemma, we need to show that

(1) at the beginning of round $t+1$, the big spender b^{t+1} has never been identified as a least spender in rounds $1, 2, \dots, t+1$;
(2) if $b^{t+1} \in N_r$ has not been raised (when round $t+1$ begins) then all agent $i \in N_r$ has never been identified as a least spender in rounds $1, 2, \dots, t+1$.

We first consider the case when b^{t+1} has not been raised when round $t+1$ begins. Assume the contrary of statement, and let $t' \le t$ be the last round in which some agent $i \in N_r$ was identified as the least spender, i.e., $i = l^{t'}$. Note that i can be b^{t+1}. By definition, in rounds $t'+1, t'+2, \dots, t$, agents in N_r did not receive any item; in round t' only agent $i \in N_r$ receives one item.

- If $i \neq b^{t+1}$, we have

$$\hat{p}_{b^{t+1}}^{t+1} \le \hat{p}_{b^{t+1}}^{t'} \le \frac{p^{t'}(X_i^{t'})}{w_i} \le \frac{p^{t+1}(X_{l^{t+1}}^{t+1})}{w_{l^{t+1}}},$$

where the first inequality holds because b^{t+1} did not receive any item and is not raised in rounds $t', t'+1, \dots, t$; the second inequality holds because agents i and b^{t+1} are in the same group; and the last inequality follows from Invariant 5.15.
- Similarly, if $i = b^{t+1}$, we have

$$\hat{p}_{b^{t+1}}^{t+1} \le \hat{p}_{b^{t+1}}^{t'+1} \le \frac{p^{t'}(X_{b^{t+1}}^{t'})}{w_{b^{t+1}}} \le \frac{p^{t+1}(X_{l^{t+1}}^{t+1})}{w_{l^{t+1}}}, \tag{3}$$

where the first inequality holds because b^{t+1} did not receive any item and is not raised in rounds $t'+1, t'+2, \dots, t$; the second inequality holds because agent b^{t+1} loses only one item during round t'; and the last inequality follows from Invariant 5.15.

Therefore we have a contradiction since b^{t+1} strongly envies l^{t+1}.

It remains to consider the case that b^{t+1} has been raised when round $t+1$ begins, and prove statement (1). Assume otherwise and let $t' \leq t$ be the last round in which b^{t+1} is identified as the least spender. If b^{t+1} has been raised when round t' begins, then Inequality (3) remains valid, and we have a contradiction. If b^{t+1} is not raised when round t' begins, then b^{t+1} can not be the big spender or raised in rounds $t'+1, t'+2, \dots, t$ because $X^{t'+1}_{b^{t+1}} \not\subseteq X^0_{b^{t+1}}$, by Invariant 5.16 and 5.13. Therefore we also have a contradiction as we assumed that b^{t+1} has been raised when round $t+1$ begins. $\square$

Lemma 5.26. *The Price Invariant (Invariant 5.17) is maintained at the end of round t.*

Proof. We first show that $p(e) \in \{1, k\}$ for all $e \in M$ when round t ends. Suppose N_r is raised during round t. By Invariant 5.16, for all $i \in N_r$, agent i did not receive any item before round t, i.e. $X_i \subseteq X_i^0$. By Invariant 5.13, for all $i \in N_r$ we have $\alpha_i = 1$, which implies that for all $e \in X_i \subseteq \mathsf{MPB}_i$, $p(e) = \frac{c_i(e)}{\alpha_i} = 1$. Hence raising the price of e does not violate the invariant.

Next, we show that for all agent i that is not raised, nor the big or least spender, the spending of i does not change during round t. If l is not raised when round t begins, then the algorithm reallocates an item from b to l, and the statement trivially holds. If l is already raised when round t begins, then it suffices to consider that case when the reallocate happens following path $b \to i \to l$. By Lemma 5.20, i (which is not raised) receives an item $e_1 \in X_b \cap \mathsf{MPB}_i$ and loses an item $e_2 \in X_i \cap X_l^0$. Since every item our algorithm reallocates is raised, we have $p(e_1) = p(e_2) = k$, which implies that the spending of i remains unchanged after updating $X_i \leftarrow X_i + e_1 - e_2$. $\square$

Given that all invariants are maintained, to prove Theorem 5.2, it suffices to argue that the algorithm terminates in a polynomial number of rounds.

Lemma 5.27. *Algorithm 4 terminates in polynomial time.*

Proof. It can be verified that each round of Algorithm 4 finishes in $O(m+n)$ time. Therefore it suffices to argue that the algorithm terminates after a polynomial number of rounds. We consider the least spenders across different rounds and we focus on the appearances of an arbitrary agent i. By the design of our algorithm, in every round in which i is the least spender, the size of X_i increases by 1. Moreover, by Invariant 5.16, agent i would not be a big spender after the first time i is identified as a least spender. Hence i can be identified as a least spender at most m times because each of its appearance (as a least spender) increases its bundle size by one. Therefore, the total number of rounds is at most nm and the algorithm terminates in $O((m+n)nm)$ time. $\square$

6. Weighted adjust winner and efficient allocations

6.1. WEF1 and PO allocations for two agents

When there are only two agents, the weighted adjusted winner algorithm [20] computes a WEF1 and PO allocation for goods in polynomial time. In this section, we show that there exists an algorithm that computes a WEF1 and PO for instances of chores with two agents in polynomial time.

Theorem 6.1. *For the weighted allocation of chores with two agents, there exists an algorithm that can always compute a WEF1 and PO allocation for chores in polynomial time.*

Proof. Given any instance of chores $I = \langle N, M, \mathbf{w}, \mathbf{c}\rangle$, we construct a corresponding instance of goods $I' = \langle M, N, \mathbf{w}', \mathbf{v}\rangle$ while the valuation functions hold $v_i(e) = c_i(e)$ for any $i \in N, e \in M$, and the weights hold $w'_1 = w_2, w'_2 = w_1$. Note that for the allocation of goods, WEF1 and PO allocations have been proven to exist for two agents [20]. Let $\mathbf{X}' = \{X'_1, X'_2\}$ be a WEF1 and PO allocation for I'. Then we compute an allocation $\mathbf{X}$ for chores by $X_1 = X'_2, X_2 = X'_1$. We argue the allocation $\mathbf{X}$ is WEF1 and PO.

Note that for the cases of two agents, we have $M = X_1 \cup X_2$, leading to $X_1 = M \setminus X'_1, X_2 = M \setminus X'_2$. We first show that the allocation $\mathbf{X}$ is PO. Assume otherwise and there exists another allocation $\mathbf{X}^* = \{X^*_1, X^*_2\}$ that dominates $\mathbf{X}$. We assume w.l.o.g. that $c_i(X^*_1) < c_1(X_1), c_2(X^*_2) \leq c_2(X_2)$. Then for the instance I', we have $v_1(M \setminus X^*_1) > v_1(M \setminus X_1) = v_1(X'_1)$ and $v_2(M \setminus X^*_2) \geq v_2(M \setminus X_2) = v_2(X'_2)$, which contradict the fact that $\mathbf{X}'$ is PO. Next, we show that the allocation $\mathbf{X}$ is WEF1. Note that in the allocation $\mathbf{X}'$, for any $i \neq j$ there exists an item $e \in X_j$ such that

$$\frac{v_i(X'_i)}{w'_i} \geq \frac{v_i(X'_j - e)}{w'_j} \Rightarrow \frac{c_i(X_j)}{w_j} \geq \frac{c_i(X_i - e)}{w_i}.$$

The deduction holds since $X_i = X'_j$ and $w'_i = w_j$. Hence the allocation $\mathbf{X}$ is WEF1 and PO for the instance of chores I. $\square$

6.2. Price of fairness

In this section, we focus on the price of fairness, which is introduced by Bertsimas et al. [33], Caragiannis et al. [34] to capture the efficiency loss due to the fairness constraints. As in previous works [37,35,12], we assume that the cost functions are normalized,

Table 4
Hard instance for lower bounding the price of WEF1.

	e_1	e_2	e_3
agent 1	0	1/2	1/2
agent 2	$\frac{\alpha}{\alpha+2} - 2\epsilon$	$\frac{1}{\alpha+2} + \epsilon$	$\frac{1}{\alpha+2} + \epsilon$

i.e. $c_i(M) = 1$ for all $i \in N$. We use $\mathrm{sc}(\mathbf{X}) = \sum_{i\in N} c_i(X_i)$ to denote the social cost of allocation $\mathbf{X}$. In this section we analyze the price of fairness (PoF) for WEF1. Given an instance I, the PoF with respect to WEF1 is the ratio between the minimum social cost of WEF1 allocations and the (unconstrained) optimal social cost.

Definition 6.2 *(PoF for WEF1).* The price of fairness for WEF1 on a given instance I is defined as

$$\mathsf{PoF}(I) = \min_{\text{WEF1 allocation } \mathbf{X}} \left\{ \frac{\mathrm{sc}(\mathbf{X})}{\mathrm{opt}(I)} \right\},$$

where $\mathrm{opt}(I)$ denotes the unconstrained optimal social cost. The price of fairness for WEF1 is defined as the maximum price of fairness over all instances, e.g., $\mathsf{PoF} = \sup_I \{\mathsf{PoF}(I)\}$.

For the unweighted setting, Sun et al. [37] have shown that the price of EF1 is unbounded when $n \geq 3$. For $n = 2$, the price of EF1 is $\frac{5}{4}$ and there exists an algorithm that computes an EF1 allocation $\mathbf{X}$ achieving a social cost $\mathrm{sc}(\mathbf{X}) \leq \frac{5}{4} \cdot \mathrm{opt}(I)$ for any given instance I on two agents. In this section, we consider the price of fairness for WEF1 (price of WEF1). The result of Sun et al. [37] implies that the price of WEF1 is unbounded for $n \geq 3$. Thus we consider the case when $n = 2$ and show that the price of WEF1 is $(4+\alpha)/4$, where $\alpha = \frac{\max\{w_1,w_2\}}{\min\{w_1,w_2\}} \geq 1$ is the weight ratio between the two agents. Note that the ratio coincides with $\frac{5}{4}$ in the unweighted setting when $\alpha = 1$.

Theorem 6.3. *The price of WEF1 is $(4+\alpha)/4$ for instances with two agents, where $\alpha = \frac{\max\{w_1,w_2\}}{\min\{w_1,w_2\}}$.*

To prove the theorem, we first provide an instance I for which any WEF1 allocation has social cost at least $\frac{4+\alpha}{4} \cdot \mathrm{opt}(I)$; then we present an algorithm that computes a WEF1 allocation with social cost at most $\frac{4+\alpha}{4} \cdot \mathrm{opt}(I)$, for any given instance I.

Consider the following instance I with $w_1 = \frac{\alpha}{1+\alpha}$ and $w_2 = \frac{1}{1+\alpha}$, for some $\alpha \geq 1$. The cost functions of the two agents are shown in Table 4, where $\epsilon > 0$ is arbitrarily small.

For the given instance the optimal social cost is $\mathrm{opt}(I) = \frac{2}{\alpha+2} + 2\epsilon$, achieved by the allocation $X_1 = \{e_1\}$, $X_2 = \{e_2, e_3\}$. However, the allocation is not WEF1 since for all $e \in X_2$, we have

$$\frac{c_2(X_2 - e)}{w_2} = \frac{\alpha+1}{\alpha+2} + \epsilon \cdot (\alpha+1) > \frac{\alpha+1}{\alpha+2} - 2\epsilon \cdot (1 + \frac{1}{\alpha}) = \frac{c_2(X_1)}{w_1}.$$

Hence, any WEF1 allocation $\mathbf{X}$ allocates at most one item in $\{e_2, e_3\}$ to agent 2, leading to

$$sc(\mathbf{X}) \geq \frac{1}{2} + \frac{1}{\alpha+2} + \epsilon.$$

Therefore the price of WEF1 for two agents with $\frac{\max\{w_1,w_2\}}{\min\{w_1,w_2\}} = \alpha$ is

$$\mathsf{PoF} \geq \frac{\frac{1}{2} + \frac{1}{\alpha+2} + \epsilon}{\frac{2}{\alpha+2} + 2\epsilon} = \frac{\alpha + 4 + 2\epsilon \cdot (\alpha+2)}{4 + 4\epsilon \cdot (\alpha+2)},$$

which tends to $\frac{4+\alpha}{4}$ when $\epsilon \to 0$.

Next, we present an algorithm that computes WEF1 allocations achieving the price of WEF1.

Lemma 6.4. *There exists an algorithm that computes a WEF1 allocation with social cost at most $\frac{4+\alpha}{4} \cdot \mathrm{opt}(I)$, for any given instance I with two agents having weight ratio $\frac{\max\{w_1,w_2\}}{\min\{w_1,w_2\}} = \alpha$.*

Proof. We consider the weighted adjusted winner algorithm (see Algorithm 5) for two agents. We index the items in non-decreasing order of their cost-ratios $\frac{c_1(e)}{c_2(e)}$, i.e.,

$$\frac{c_1(e_1)}{c_2(e_1)} \leq \frac{c_1(e_2)}{c_2(e_2)} \leq \cdots \leq \frac{c_1(e_m)}{c_2(e_m)}.$$

Let $O_1 = \{e \in M : c_1(e) < c_2(e)\}$ and $O_2 = \{e \in M : c_1(e) \ge c_2(e)\}$. Note that the allocation $\mathbf{O} = (O_1, O_2)$ minimizes the social cost, i.e., $\mathrm{sc}(\mathbf{O}) = \mathrm{opt}(I)$. If $\mathbf{O}$ is WEF1 then our algorithm terminates and outputs $\mathbf{O}$. Otherwise we compute a WEF1 allocation as follows.

For all integer t, let $L(t) = \{e_1, e_2, \dots, e_t\}$ and $R(t) = \{e_t, e_{t+1}, \dots, e_m\}$. Note that $L(t)$ and $R(t)$ can be empty, e.g., when $t < 1$ or $t > m$. Let f be the maximum index satisfying

$$\frac{c_2(R(f+1))}{w_2} > \frac{c_2(L(f-1))}{w_1}.$$

We return the allocation $\mathbf{X}$ with $X_1 = L(f)$ and $X_2 = R(f+1)$.

Algorithm 5: Weighted Adjusted Winner Algorithm.

Input: Instance with two agents satisfying $\frac{c_1(e_1)}{c_2(e_1)} \le \frac{c_1(e_2)}{c_2(e_2)} \le \cdots \le \frac{c_1(e_m)}{c_2(e_m)}$

let $O_1 = \{e \in M : c_1(e) < c_2(e)\}$ and $O_2 = \{e \in M : c_1(e) \ge c_2(e)\}$;
if *allocation* (O_1, O_2) *is WEF1* **then**
    Output: $\mathbf{O} = \{O_1, O_2\}$.
else
    assume w.l.o.g. that $\frac{c_1(O_1)}{w_1} \le \frac{c_2(O_2)}{w_2}$ (otherwise we reverse the index of agents and items) ;
    find the maximum index f such that $\frac{c_2(R(f+1))}{w_2} > \frac{c_2(L(f-1))}{w_1}$;
    $X_1 \leftarrow L(f)$, $X_2 \leftarrow R(f+1)$;
    Output: $\mathbf{X} = \{X_1, X_2\}$.

In the following we show that the allocation $\mathbf{X}$ is WEF1 and $\mathrm{sc}(\mathbf{X}) \le \frac{4+\alpha}{4} \cdot \mathrm{sc}(\mathbf{O})$. By definition of f we know that agent 2 is WEF1 towards agent 1 because

$$\frac{c_2(X_2 - e_{f+1})}{w_2} = \frac{c_2(R(f+2))}{w_2} \le \frac{c_2(L(f))}{w_1} = \frac{c_2(X_1)}{w_1}.$$

We also have that agent 1 is WEF1 towards agent 2 because

$$\begin{aligned}\frac{c_1(X_1 - e_f)}{w_1} &= \frac{c_1(L(f-1))}{w_1} = \frac{c_1(L(f-1))}{c_2(L(f-1))} \cdot \frac{c_2(L(f-1))}{w_1}\\ &< \frac{c_1(L(f-1))}{c_2(L(f-1))} \cdot \frac{c_2(R(f+1))}{w_2}\\ &\le \frac{c_1(X_2)}{c_2(X_2)} \cdot \frac{c_2(R(f+1))}{w_2} = \frac{c_1(X_2)}{w_2},\end{aligned}$$

where the second inequality follows from (recall that the items are sorted in non-decreasing order of $\frac{c_1(e)}{c_2(e)}$)

$$\frac{c_1(L(f-1))}{c_2(L(f-1))} = \frac{\sum_{t=1}^{f-1} c_1(e_t)}{\sum_{t=1}^{f-1} c_2(e_t)} \le \frac{c_1(e_f)}{c_2(e_f)} \le \frac{\sum_{t=f+1}^{m} c_1(e_t)}{\sum_{t=f+1}^{m} c_2(e_t)} = \frac{c_1(X_2)}{c_2(X_2)}.$$

Next, we prove that $\mathrm{sc}(\mathbf{X}) = c_1(X_1) + c_2(X_2) \le \frac{4+\alpha}{4} \cdot \mathrm{sc}(\mathbf{O})$.

Recall that $O_1 = \{e \in M : c_1(e) < c_2(e)\}$ and $O_2 = \{e \in M : c_1(e) \ge c_2(e)\}$, which implies $c_1(O_1) \le c_2(O_1)$ and $c_2(O_2) \le c_1(O_2)$. Moreover, we assumed w.l.o.g. that $\frac{c_1(O_1)}{w_1} \le \frac{c_2(O_2)}{w_2}$. For ease of notation in the following we use A and B to denote $c_1(O_1)$ and $c_2(O_2)$, respectively. Then we have $\frac{A}{w_1} \le \frac{B}{w_2}$ and $\mathrm{sc}(\mathbf{O}) = A + B$. Since the cost functions are normalized, we have

$$c_1(O_2) = 1 - A \quad \text{and} \quad c_2(O_1) = 1 - B.$$

In the following, we upper bound $\mathrm{sc}(\mathbf{X})$ by a function of A and B.

Claim 6.1. *We have $A \le w_1$, $B \ge w_2$ and $O_1 \subsetneq X_1$.*

Proof. By definition and assumption, we have

$$\frac{A}{w_1} \le \frac{B}{w_2} \le \frac{c_1(O_2)}{w_2} = \frac{1-A}{w_2},$$

which implies $A \le w_1$. The above inequality also implies that agent 1 does not envy agent 2 in allocation $\mathbf{O}$. Since $\mathbf{O}$ is not WEF1, agent 2 must envy agent 1, i.e.,

$$\frac{B}{w_2} > \frac{c_2(O_1)}{w_1} = \frac{1-B}{w_2},$$

which implies $B \geq w_2$. For the last property, if O_1 is not a proper subset of X_1, then we have $X_1 \subseteq O_1$ and $O_2 \subseteq X_2$. Since agent 2 is WEF1 towards agent 1 in allocation $\mathbf{X}$, agent 2 should also be WEF1 towards agent 1 in $\mathbf{O}$, which implies that $\mathbf{O}$ is WEF1 and is a contradiction. $\square$

Claim 6.2. *We have* $\mathrm{sc}(\mathbf{X}) \leq 1 - \frac{1-A-B}{B} \cdot \frac{w_2}{w_1} \cdot (1-B)$

Proof. Given Claim 6.1, we have $O_1 \subsetneq X_1$ and $X_2 \subsetneq O_2$. Therefore we have

$$c_1(X_1) = 1 - c_1(X_2) \leq 1 - \frac{c_1(O_2)}{c_2(O_2)} \cdot c_2(X_2),$$

where the inequality holds because X_2 is a proper subset of O_2 and it contains the items in O_2 with maximum cost-ratios. Furthermore, by definition of f we have

$$c_2(X_2) = c_2(R(f+1)) > \frac{w_2}{w_1} \cdot c_2(L(f-1)) \geq \frac{w_2}{w_1} \cdot c_2(O_1).$$

Putting the bounds together, we get

$$\begin{aligned}
c_1(X_1) + c_2(X_2) &\leq 1 - \frac{c_1(O_2)}{c_2(O_2)} \cdot c_2(X_2) + c_2(X_2) \\
&= 1 - \left(\frac{c_1(O_2)}{c_2(O_2)} - 1\right) \cdot c_2(X_2) \\
&\leq 1 - \left(\frac{c_1(O_2)}{c_2(O_2)} - 1\right) \cdot \frac{w_2}{w_1} \cdot c_2(O_1) \\
&= 1 - \frac{1-A-B}{B} \cdot \frac{w_2}{w_1} \cdot (1-B),
\end{aligned}$$

where the second inequality holds because $c_1(O_2) \geq c_2(O_2)$ and $c_2(X_2) > \frac{w_2}{w_1} \cdot c_2(O_1)$. $\square$

Given the above claim, letting $C = \frac{w_2}{w_1}$, we have

$$\frac{\mathrm{sc}(\mathbf{X})}{\mathrm{sc}(\mathbf{O})} \leq \frac{1 - \frac{1-A-B}{B} \cdot C \cdot (1-B)}{A+B} = \frac{C}{A+B} \cdot \left(\frac{1}{C} - \frac{1}{B} + 1\right) + \frac{C(1-B)}{B}.$$

By Claim 6.1 we have $B \geq w_2$, which implies

$$\frac{1}{C} - \frac{1}{B} + 1 = \frac{1}{w_2} - \frac{1}{B} \geq 0.$$

Hence the upper bound on $\frac{\mathrm{sc}(\mathbf{X})}{\mathrm{sc}(\mathbf{O})}$ is maximized when $A = 0$, which gives

$$\begin{aligned}
\frac{\mathrm{sc}(\mathbf{X})}{\mathrm{sc}(\mathbf{O})} &\leq \frac{C}{B}\left(\frac{1}{C} - \frac{1}{B} + 1\right) + \frac{C(1-B)}{B} = \frac{2C+1}{B} - \frac{C}{B^2} - C \\
&= \left(\frac{2C+1}{2\sqrt{C}}\right)^2 - \left(\frac{2C+1}{2\sqrt{C}} - \frac{\sqrt{C}}{B}\right)^2 - C \\
&\leq \left(\frac{2C+1}{2\sqrt{C}}\right)^2 - C = 1 + \frac{1}{4C} = 1 + \frac{w_2}{4 \cdot w_1} \leq \frac{4+\alpha}{4},
\end{aligned}$$

where the last inequality follows from the definition $\alpha = \frac{\max\{w_1,w_2\}}{\min\{w_1,w_2\}}$. Hence we have $\mathrm{sc}(\mathbf{X}) \leq \frac{4+\alpha}{4} \cdot \mathrm{sc}(\mathbf{O})$, and the proof is complete. $\square$

7. WEF1 and other fairness notions

In this section, we discuss the relation between the fairness notion of WEF1 and other notions including *weighted proportional up to one item* (WPROP1) and *AnyPrice Share* (APS). We show that WEF1 implies WPROP1 and $(2 - \min_{i \in N} w_i)$ approximation of APS.

Definition 7.1 *(WPROP1).* An allocation is called weighted proportional up to one item (WPROP1) if for any $i \in N$, there exists an item $e \in X_i$ such that

$$c_i(X_i - e) \leq w_i \cdot c_i(M).$$

Table 5
Example showing that WPROP1 does not imply WEF1, where $\epsilon > 0$ is arbitrarily small.

	e_1	e_2	e_3
Agent 1	ϵ	$\boxed{1}$	$\boxed{1}$
Agent 2	$\boxed{\epsilon}$	1	1

It is well known that EF1 implies PROP1 for both allocations of goods and chores, in the unweighted setting. However, when agents have general weights, Chakraborty et al. [20] show that WEF1 allocations are not necessarily WPROP1 for the allocation of goods. In contrast, we show in the following that any WEF1 allocation is WPROP1 for the allocation of chores.

Lemma 7.2. *Any WEF1 allocation is WPROP1, but not vice versa.*

Proof. Let $\mathbf{X}$ be any WEF1 allocation. Fix any agent i and let $e^* = \arg\max_{e\in X_i}\{c_i(e)\}$ be the item with maximum cost in X_i. By the definition of WEF1, for all $j \in N$ we have

$$\frac{c_i(X_i - e^*)}{w_i} \le \frac{c_i(X_j)}{w_j} \quad \Rightarrow \quad \frac{w_j}{w_i} \cdot c_i(X_i - e^*) \le c_i(X_j).$$

Therefore we have

$$c_i(X_i - e^*) = w_i \cdot \frac{\sum_{j\in N} w_j}{w_i} \cdot c_i(X_i - e^*) \le w_i \cdot \sum_{j\in N} c_i(X_j) = w_i \cdot c_i(M),$$

which implies that the allocation is WPROP1.

Finally, via the following simple example, we show that WPROP1 can not guarantee (any bounded approximation of) WEF1, even for two symmetric agents with identical cost functions (see Table 5).

It can be easily observed that the allocation highlighted by the boxed items is WPROP1,[8] but is not WEF1 because $c_1(X_1 - e) \ge 1/\epsilon \cdot c_1(X_2)$, for all $e \in X_1$. □

Next we study the fairness notion of AnyPrice Share (APS) fair. The notion is introduced by Babaioff et al. [42] for the allocation of goods, and is then adapted to the case of chores in [12,45].

Fix any agent $i \in M$. We call $r = (r_1, r_2, \cdots, r_m)$ a *reward vector* if $r_e \ge 0$ for all $e \in M$ and $r(M) = \sum_{e\in M} r_e = c_i(M)$. Let R be the set of all reward vectors. The AnyPrice Share APS_i of agent i is defined as the minimum value such that no matter how the reward vector is set, agent i can always find a subset of chores with total reward at least w_i and total cost at most APS_i.

Definition 7.3 *(APS).* The AnyPrice Share (APS) of agent i with weight w_i is defined as

$$\mathsf{APS}_i = \max_{r\in R} \min_{S\subseteq M} \left\{ c_i(S) : \sum_{e\in S} r_e \ge w_i \cdot c_i(M) \right\}.$$

An allocation is called α-APS allocation if for all $i \in N$, it holds that $c_i(X_i) \le \alpha \cdot \mathsf{APS}_i$.

Lemma 7.4. *If an allocation $\mathbf{X}$ is WEF1, then it is $(2 - \min_{i\in N} w_i)$-APS. However, APS allocations cannot guarantee a constant approximation of WEF1.*

Proof. Fix any agent i. We show that $\mathbf{X}$ is $(2 - w_i)$-APS to i. As it is commonly observed, e.g., see [12], we have the following property regarding APS_i:

$$\mathsf{APS}_i \ge \max\left\{ w_i \cdot c_i(M),\ \max_{e\in M}\{c_i(e)\} \right\}.$$

Let $e^* = \arg\max_{e\in X_i}\{c_i(e)\}$ be the item with maximum cost in X_i. Since $\mathbf{X}$ is WEF1, we have

$$c_i(X_i - e^*) = w_i \cdot \frac{\sum_{j\in N} w_j}{w_i} \cdot c_i(X_i - e^*)$$

[8] In fact the allocation is WPROPX, a fairness notion that requires $c_i(X_i - e) \le w_i \cdot c_i(M)$ for all $i \in N$ and $e \in X_i$.

Table 6
Instance showing that the approximation ratio $2 - \min_{i\in N} w_i = 2 - \frac{1}{n}$ is tight.

Agents	e_1	e_2	e_3	$\cdots$	e_n	e_{n+1}	$\cdots$	e_{2n-1}
1	$\boxed{\frac{1}{n}}$	$\boxed{\left(1-\frac{1}{n}\right)\cdot\frac{1}{n}}$	$\left(1-\frac{1}{n}\right)\cdot\frac{1}{n}$	$\cdots$	$\left(1-\frac{1}{n}\right)\cdot\frac{1}{n}$	$\frac{1}{n^2}$	$\cdots$	$\frac{1}{n^2}$
2	$\frac{1}{n}$	$\left(1-\frac{1}{n}\right)\cdot\frac{1}{n}$	$\boxed{\left(1-\frac{1}{n}\right)\cdot\frac{1}{n}}$	$\cdots$	$\left(1-\frac{1}{n}\right)\cdot\frac{1}{n}$	$\frac{1}{n^2}$	$\cdots$	$\frac{1}{n^2}$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
$n-1$	$\frac{1}{n}$	$\left(1-\frac{1}{n}\right)\cdot\frac{1}{n}$	$\left(1-\frac{1}{n}\right)\cdot\frac{1}{n}$	$\cdots$	$\boxed{\left(1-\frac{1}{n}\right)\cdot\frac{1}{n}}$	$\frac{1}{n^2}$	$\cdots$	$\frac{1}{n^2}$
n	$\frac{1}{n}$	$\left(1-\frac{1}{n}\right)\cdot\frac{1}{n}$	$\left(1-\frac{1}{n}\right)\cdot\frac{1}{n}$	$\cdots$	$\left(1-\frac{1}{n}\right)\cdot\frac{1}{n}$	$\boxed{\frac{1}{n^2}}$	$\cdots$	$\boxed{\frac{1}{n^2}}$

$$\leq w_i \cdot \sum_{j\in N\setminus\{i\}} c_i(X_j) + w_i \cdot c_i(X_i - e^*) = w_i \cdot c_i(M) - w_i \cdot c_i(e^*).$$

Hence we have

$$\begin{aligned} c_i(X_i) &\leq w_i \cdot c_i(M) + (1 - w_i)\cdot c_i(e^*) \\ &\leq \mathsf{APS}_i + (1-w_i)\cdot \mathsf{APS}_i = (2 - w_i)\cdot \mathsf{APS}_i, \end{aligned}$$

where the second inequality follows because $\mathsf{APS}_i \geq w_i \cdot c_i(M)$ and $\mathsf{APS}_i \geq c_i(e^*)$.

Next, we give an example instance showing that the approximation ratio $(2 - \min_{i\in N} w_i)$ is tight. Consider the following instance with n symmetric agents with identical cost functions.

As shown in Table 6, there are $2n-1$ items to be allocated to n symmetric agents, for which we have $\mathsf{APS}_i = \mathsf{APS} \geq 1/n$ for all $i \in N$. In fact, we have $\mathsf{APS} = 1/n$ because given any reward vector, one of the bundles $\{e_1\}, \{e_2, e_{n+1}\}, \{e_3, e_{n+2}\}, \dots, \{e_n, e_{2n-1}\}$ must have reward at least $1/n$, and can be chosen to ensure a cost of $1/n$ when defining APS. Consider the allocation $\mathbf{X}$ indicated by the boxed items. Since all agents other than 1 have cost $\left(1 - \frac{1}{n}\right)\cdot\frac{1}{n}$ and agent 1 has cost $\left(1-\frac{1}{n}\right)\cdot\frac{1}{n}$ after removing e_1, the allocation is EF1. However, the allocation is not better than $\left(2-\frac{1}{n}\right)$-APS because $c_1(X_1) = \left(2 - \frac{1}{n}\right)\cdot \mathsf{APS}$.

Finally, we provide a simple example showing that APS allocations do not guarantee any constant approximation of WEF1, even for the unweighted setting. Consider an instance with four items $\{e_1, e_2, e_3, e_4\}$ and three identical agents with cost function $c = (1, 1/2, 1/2, \epsilon)$, where $\epsilon > 0$ is arbitrarily small. By allocating e_1 to agent 1, $\{e_2, e_3\}$ to agent 2 and e_4 to agent 3, we have an APS allocation $\mathbf{X}$ since $\mathsf{APS} \geq c(e_1) = 1$ for all $i \in N$. However, this allocation is far from being EF1 since $c(X_2 - e) = 1/2 \geq \omega(1)\cdot\epsilon = \omega(1)\cdot c(X_3)$ for all $e \in X_2$. □

8. Conclusion

In this paper, we consider the fairness notion of weighted EF1 and try to paint a complete picture of WEF1 for the allocation of indivisible chores. We show that WEF1 allocations always exist for chores and propose a polynomial time algorithm to compute one, based on the weighted picking sequence algorithm. We further consider the picking sequences that satisfy other fairness notions, e.g. WEF(x, y) with $x + y \geq 1$. For the results of best-of-both-worlds, we propose a lottery that provides tight guarantees of ex-ante WEF and ex-post WEF(1,1). We also consider allocations that are fair and efficient, showing that WEF1 and PO allocations exist for bi-valued instances and for two agents. Finally, we consider the price of fairness regarding WEF1 and provide a tight characterization of the price of WEF1 for two agents.

Our work leaves many interesting problems open. For example, whether WEF1 allocations exist in the mixed manner (mixture of goods and chores) remains unknown. It is also unknown whether (weighted) EF1 and PO allocations exist for the allocation of chores, when agents have general additive cost functions. It would also be an interesting direction to explore the existence of WEF1 allocations when agents have cost functions beyond additive.

CRediT authorship contribution statement

Xiaowei Wu: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Project administration, Methodology, Formal analysis. **Cong Zhang:** Writing – original draft, Methodology, Formal analysis. **Shengwei Zhou:** Writing – review & editing, Writing – original draft, Visualization, Validation, Project administration, Methodology, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Analysis of other picking sequence algorithms

A.1. Allocation of goods

In this section, we apply our continuous perspective to the allocation of goods. We provide an alternative and arguably simpler proof of Theorem 3.3 in [20]. In the allocation of goods, each agent has an additive valuation function $v_i : 2^M \to \mathbb{R}^+ \cup \{0\}$, and agents want to maximize the utility of their bundles.

Definition A.1 *(WEF1 for Goods)*. An allocation is *weighted envy-free up to one good* (WEF1) if for any $i, j \in N$, there exists an item $e \in X_j$ such that

$$\frac{v_i(X_i)}{w_i} \geq \frac{v_i(X_j - e)}{w_j}.$$

Lemma A.2. *The weighted picking sequence protocol [20] computes WEF1 allocations for the allocation of goods in polynomial time.*

Proof. Fix any two agents $i, j \in N$, we show that agent i is WEF1 towards j. Let $X_i = \{e_1, \cdots, e_k\}$ and $X_j = \{e'_1, \cdots, e'_{k'}\}$ be the bundles agent i and j receive in the final allocation, respectively. Similar to our previous analysis, we assume the items are ordered in increasing order of the index of rounds in which they are chosen. In the following we show that $\frac{v_i(X_i)}{w_i} \geq \frac{v_i(X_j - e'_1)}{w_j}$.

We define a continuous non-increasing function $\rho : (0, k/w_i] \to \mathbb{R}^+$ such that $\rho(\alpha) = v_i(e_z)$, for $\alpha \in \left(\frac{z-1}{w_i}, \frac{z}{w_i}\right]$, where $z \in \{1, 2, \ldots, k\}$. Similarly, we define $\rho' : (0, k'/w_j] \to \mathbb{R}^+$ be a continuous function: $\rho'(\alpha) = v_i(e'_z)$, for $\alpha \in \left(\frac{z-1}{w_j}, \frac{z}{w_j}\right]$, where $z \in \{1, 2, \ldots, k'\}$. By definition of ρ and ρ', we have

$$\frac{c_i(X_i)}{w_i} = \int_0^{\frac{k}{w_i}} \rho(\alpha) d\alpha, \quad \text{and} \quad \frac{c_i(X_j - e'_1)}{w_j} = \int_{\frac{1}{w_j}}^{\frac{k'}{w_j}} \rho'(\alpha) d\alpha.$$

Next we establish two technical claims to show that $\frac{v_i(X_i)}{w_i} \geq \frac{v_i(X_j - e'_1)}{w_j}$.

Claim A.1. *We have $k/w_i \geq (k' - 1)/w_j$.*

Proof. Consider the moment in time t' when $s_j(t)$ starts to grow from $(k'-1)/w_j$, i.e. $t' = \max\{t : s_j(t) = (k'-1)/w_j\}$. At time t', since $s_j(t')$ is chosen to grow, we must have that $s_j(t') \leq s_i(t')$. Therefore we have $k/w_i \geq s_i(t') \geq s_j(t') = (k'-1)/w_j$, as claimed. □

Claim A.2. *For all $\alpha \in \left(\frac{1}{w_j}, \frac{k'}{w_j}\right]$, we have $\rho\left(\alpha - \frac{1}{w_j}\right) \geq \rho'(\alpha)$.*

Proof. Fix any α and suppose that $\rho'(\alpha) = v_i(e'_z)$, i.e., $\alpha \in \left(\frac{z-1}{w_j}, \frac{z}{w_j}\right]$. Let t_2 be the minimum such that $s_j(t_2) = \alpha$. Let t^* be the maximum integer that is smaller than t_2. By definition, from time t^* to t^*+1, $s_j(t)$ grows from $(z-1)/w_j$ to z/w_j and $t_2 \in (t^*, t^*+1]$. Let t_1 be the minimum such that $s_i(t_1) = \alpha - 1/w_j$. By definition we have $\rho(\alpha - 1/w_j) = v_i(e_x)$, where e_x is the item agent i is consuming at time t_1. Since at time t^*, $s_j(t)$ is chosen to grow, we have $s_j(t^*) = (z-1)/w_j \leq s_i(t^*)$. Since $\alpha \in \left(\frac{z-1}{w_j}, \frac{z}{w_j}\right]$, we have $\alpha - 1/w_j \leq (z-1)/w_j$. Recall that t_1 is the minimum such that $s_i(t_1) = \alpha - 1/w_j$. Since $s_i(t^*) \geq (z-1)/w_j \geq \alpha - 1/w_j$ and $s_i(t)$ is non-decreasing, we have $t_1 \leq t^*$. Since $t_2 \in (t^*, t^*+1]$, we have $t_1 \leq t^* < t_2$. In other words, the event that "agent i picks item e_x" happens strictly earlier than the event "agent j picks item e'_z". Since agent i picks item e_x when e'_z is still available, we have $v_i(e_x) \geq v_i(e'_z)$, which implies $\rho(\alpha - 1/w_j) = v_i(e_x) \geq v_i(e'_z) = \rho'(\alpha)$ and concludes the proof. □

Combining Claim Appendix A.1 and Appendix A.2, we have

$$\frac{c_i(X_i)}{w_i} = \int_0^{\frac{k}{w_i}} \rho(\alpha)d\alpha \geq \int_0^{\frac{k'-1}{w_j}} \rho(\alpha)d\alpha \geq \int_0^{\frac{k'-1}{w_j}} \rho'\left(\alpha + \frac{1}{w_j}\right)d\alpha$$
$$= \int_{\frac{1}{w_j}}^{\frac{k'}{w_j}} \rho'(\alpha)d\alpha = \frac{c_i(X_j - e'_1)}{w_j},$$

where the first inequality follows from Claim Appendix A.1 and the second inequality follows from Claim Appendix A.2. Hence agent i is WEF1 towards agent j, which finishes the proof. □

A.2. Computation of WEF(x, y) allocations for chores

Proof of Theorem 3.9. We first show that the property is sufficient for ensuring WEF(x, y). Fix any agent $i, j \in N$ and let X_i and X_j be the bundles that agent i and j receive in the final allocation, respectively. Similar to Theorem 3.2, let $X_i = \{e_1, e_2, \dots, e_k\}$ and $X_j = \{e'_1, e'_2, \dots, e'_k\}$, where the items are ordered in increasing order of the index of rounds in which they are chosen. Therefore, we have $c_i(e_1) \geq c_i(e_2) \geq \dots \geq c_i(e_k)$. In the following, we show that

$$\frac{c_i(X_i) - x \cdot c_i(e_1)}{w_i} \leq \frac{c_i(X_j) + y \cdot c_i(e_1)}{w_j}.$$

Let $\rho : (0, k/w_i] \to \mathbb{R}^+$ be a continuous function such that $\rho(\alpha)$ represents the cost of the item agent i is consuming, when $s_i(t)$ reaches α. In particular, we have

$$\rho(\alpha) = c_i(e_z), \quad \text{for } \alpha \in \left(\frac{z-1}{w_i}, \frac{z}{w_i}\right], \text{ where } z \in \{1, 2, \dots, k\}.$$

Similarly, we define $\rho' : (0, k'/w_j] \to \mathbb{R}^+$ be a continuous function such that $\rho'(\alpha)$ represents the cost of the item agent j is consuming, under the cost function of agent i, when $s_j(t)$ reaches α. Hence we have

$$\rho'(\alpha) = c_i(e'_z), \quad \text{for } \alpha \in \left(\frac{z-1}{w_j}, \frac{z}{w_j}\right], \text{ where } z \in \{1, 2, \dots, k'\}.$$

By definition of ρ and ρ', we have

$$\frac{c_i(X_i) - x \cdot c_i(e_1)}{w_i} = \int_{\frac{x}{w_i}}^{\frac{k}{w_i}} \rho(\alpha)d\alpha, \quad \text{and}$$

$$\frac{c_i(X_j) + y \cdot c_i(e_1)}{w_j} = \int_0^{\frac{k'}{w_j}} \rho'(\alpha)d\alpha + \frac{y}{w_j} \cdot c_i(e_1).$$

Using the condition given in the theorem at $t = m$, we obtain the following immediately.

Claim A.3. *We have* $\frac{k}{w_i} - \frac{x}{w_i} - \frac{y}{w_j} \leq \frac{k'}{w_j}$.

Next, we establish a claim that is very similar to Lemma 3.5.

Claim A.4. *For all* $\alpha \in \left(\frac{x}{w_i} + \frac{y}{w_j}, \frac{k}{w_i}\right)$, *we have*

$$\rho(\alpha) \leq \rho'\left(\alpha - \frac{x}{w_i} - \frac{y}{w_j}\right).$$

Proof. Fix any α and suppose that $\rho(\alpha) = c_i(e_z)$, i.e., $\alpha \in \left(\frac{z-1}{w_i}, \frac{z}{w_i}\right]$. Let t_1 be the minimum such that $s_i(t_1) = \alpha$. Using the condition given in the theorem at $t = t_1$, we have

$$s_i(t_1) - \frac{x}{w_i} - \frac{y}{w_j} = \alpha - \frac{x}{w_i} - \frac{y}{w_j} \le s_j(t_1).$$

Let t_2 be the minimum such that $s_j(t_2) = \alpha - x/w_i - y/w_j$. By definition we have $\rho'(\alpha - x/w_i - y/w_j) = c_i(e'_p)$, where e'_p is the item agent j is consuming at time t_2. Since $s_j(t_2) = \alpha - \frac{x}{w_i} - \frac{y}{w_j} \le s_j(t_1)$ and $s_j(t)$ is non-decreasing, we have $t_2 \le t_1$. In other words, in the second phase of the algorithm, the event that "agent i includes item e_z into X_i" happens strictly earlier than the event that "agent j includes item e'_p into X_j". Since agent i picks item e_z when e'_p is still unallocated, we have $c_i(e_z) \le c_i(e'_p)$, which implies $\rho(\alpha) = c_i(e_z) \le c_i(e'_p) = \rho'(\alpha - x/w_i - y/w_j)$ and finishes the proof. □

Combining Claim Appendix A.3 and Appendix A.4, we have

$$\frac{c_i(X_i) - x \cdot c_i(e_1)}{w_i} = \int_{\frac{x}{w_i}}^{\frac{x}{w_i}+\frac{y}{w_j}} \rho(\alpha) d\alpha + \int_{\frac{x}{w_i}+\frac{y}{w_j}}^{\frac{k}{w_i}} \rho(\alpha) d\alpha$$

$$\le \frac{y}{w_j} \cdot c_i(e_1) + \int_{\frac{x}{w_i}+\frac{y}{w_j}}^{\frac{k}{w_i}} \rho'\left(\alpha - \frac{x}{w_i} - \frac{y}{x_j}\right) d\alpha$$

$$= \frac{y}{w_j} \cdot c_i(e_1) + \int_{0}^{\frac{k}{w_i}-\frac{x}{w_i}-\frac{y}{w_j}} \rho'(\alpha) d\alpha$$

$$\le \frac{y}{w_j} \cdot c_i(e_1) + \int_{0}^{\frac{k'}{w_j}} \rho'(\alpha) d\alpha = \frac{c_i(X_j) + y \cdot c_i(e_1)}{w_j}.$$

where the first and second inequalities hold due to Claim Appendix A.4 and Claim Appendix A.3, respectively.

Finally, we show that the condition is necessary for ensuring WEF(x, y). Assume otherwise, and let $t \in \{1, 2, \ldots, m\}$ be such that $s_i(t) - \frac{x}{w_i} > s_j(t) + \frac{y}{w_j}$ for some $i, j \in N$. Consider an instance with t items having cost 1 to all agents and $m - t$ items having cost 0 to all agents. Following the reversed picking sequence, we have

$$\frac{c_i(X_i) - x \cdot c_i(e)}{w_i} = s_i(t) - \frac{x}{w_i}, \quad \text{and} \quad \frac{c(X_j) + y \cdot c_i(e)}{w_j} = s_j(t) + \frac{y}{w_j},$$

which implies that agent i is not WEF(x, y) towards agent j, and is a contradiction. □

Next, we propose an algorithm that computes WEF(x, y) allocations for chores, for any $x + y \ge 1$, that is similar to Algorithm 1. We first construct a forward sequence and then let agents follow the reversed sequence to pick their favorite remaining items. At the beginning of the algorithm, we set $s_i = 0$ for all $i \in N$. And we decide the forward sequence $(\sigma(1), \cdots, \sigma(m))$ by the following steps: for each $l \in \{1, 2, \cdots, m\}$ we set $\sigma(l) \leftarrow i^*$ where i^* satisfies that $s_{i^*} + \frac{1-x}{w_i} \le s_j + \frac{y}{w_j}$ for any $j \ne i^*$, and then update $s_{i^*} \leftarrow s_{i^*} + \frac{1}{w_{i^*}}$. We argue (in the proof of Lemma A.3) that such an agent always exists when $x + y \ge 1$. Then by Theorem 3.9, we can show that the resulting allocation is WEF(x, y).

Algorithm 6: RWPS Algorithm for WEF(x, y).

Input: An instance $\langle M, N, \mathbf{w}, \mathbf{c} \rangle$ with additive cost functions.

```
Initialize X_i ← ∅ and s_i ← 0 for all i ∈ N, and P ← M;
for t ∈ {1, 2, ⋯, m} do
    let i* be any agent such that s_{i*} + (1−x)/w_{i*} ≤ s_j + y/w_j for any j ∈ N \ {i*};
    σ(t) ← i*, s_{i*} ← s_{i*} + 1/w_{i*};
for t ∈ {m, m−1, ⋯, 1} do
    i ← σ(t), e* ← argmin_{e∈P} {c_i(e)}, breaking ties arbitrarily;
    X_i ← X_i + e*, P ← P \ {e*};
```

Output: $\mathbf{X} = \{X_1, X_2, \cdots, X_n\}$.

Lemma A.3. *Algorithm 6 computes WEF(x, y) allocations for all $x, y \in [0, 1]$ with $x + y \ge 1$.*

Proof. We show that the agent i^* specified in line 3 of Algorithm 6 exists. In particular, let

$$i^* = \arg\min_{i\in N} \left\{ s_i + \frac{1-x}{w_i} \right\}.$$

Since $x + y \geq 1$, for all $j \in N$ we have

$$s_{i^*} + \frac{1-x}{w_{i^*}} \leq s_j + \frac{1-x}{w_j} \leq s_j + \frac{y}{w_j}.$$

Therefore the agent i^* specified in line 3 always exists and the algorithm is well-defined. Hence for any prefix of length t of the forward picking sequence, we have $s_i - \frac{x}{w_i} \leq s_j + \frac{y}{w_j}$ for any $i, j \in N$. Then by Theorem 3.9, Algorithm 6 computes WEF(x, y) allocations in polynomial time. □

We remark that $x + y \geq 1$ is necessary for ensuring that the above algorithm is well-defined. Suppose otherwise ($x + y < 1$), then the agent i^* does not exist when $t = 1$, when all agents have the same weight, because for all $i, j \in N$ we always have $\frac{1-x}{w_i} > \frac{y}{w_j}$.

Data availability

No data was used for the research described in the article.

References

[1] H. Steinhaus, The problem of fair division, Econometrica 16 (1948) 101–104.
[2] D. Foley, Resource allocation and the public sector, Yale Econ. Essays (1967) 45–98.
[3] N. Alon, Splitting necklaces, Adv. Math. 63 (1987) 247–253.
[4] F. Edward Su, Rental harmony: Sperner's lemma in fair division, Am. Math. Mon. 106 (1999) 930–942.
[5] R.J. Lipton, E. Markakis, E. Mossel, A. Saberi, On approximately fair allocations of indivisible goods, in: EC, ACM, 2004, pp. 125–131.
[6] E. Budish, The combinatorial assignment problem: approximate competitive equilibrium from equal incomes, J. Polit. Econ. 119 (2011) 1061–1103.
[7] U. Bhaskar, A.R. Sricharan, R. Vaish, On approximate envy-freeness for indivisible chores and mixed resources, in: APPROX-RANDOM, Volume 207 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021, pp. 1:1–1:23.
[8] H. Aziz, I. Caragiannis, A. Igarashi, T. Walsh, Fair allocation of indivisible goods and chores, Auton. Agents Multi-Agent Syst. 36 (2022) 3.
[9] I. Caragiannis, D. Kurokawa, H. Moulin, A.D. Procaccia, N. Shah, J. Wang, The unreasonable fairness of maximum Nash welfare, ACM Trans. Econ. Comput. 7 (2019) 12:1–12:32.
[10] B.R. Chaudhury, J. Garg, K. Mehlhorn, EFX exists for three agents, J. ACM 71 (2024) 4:1–4:27.
[11] G. Amanatidis, E. Markakis, A. Ntokos, Multiple birds with one stone: beating 1/2 for EFX and GMMS via envy cycle elimination, Theor. Comput. Sci. 841 (2020) 94–109.
[12] H. Aziz, B. Li, H. Moulin, X. Wu, X. Zhu, Almost proportional allocations of indivisible chores: computation, approximation and efficiency, Artif. Intell. 331 (2024) 104118.
[13] S. Zhou, X. Wu, Approximately EFX allocations for indivisible chores, Artif. Intell. 326 (2024) 104037.
[14] H. Akrami, N. Alon, B.R. Chaudhury, J. Garg, K. Mehlhorn, R. Mehta, EFX: a simpler approach and an (almost) optimal guarantee via rainbow cycle number, Oper. Res. 73 (2025) 738–751.
[15] B. Tao, X. Wu, Z. Yu, S. Zhou, On the existence of efx (and Pareto-optimal) allocations for binary chores, Theor. Comput. Sci. (2025) 115248.
[16] V. Conitzer, R. Freeman, N. Shah, Fair public decision making, in: EC, ACM, 2017, pp. 629–646.
[17] H. Aziz, H. Moulin, F. Sandomirskiy, A polynomial-time algorithm for computing a Pareto optimal and almost proportional allocation, Oper. Res. Lett. 48 (2020) 573–578.
[18] G. Amanatidis, H. Aziz, G. Birmpas, A. Filos-Ratsikas, B. Li, H. Moulin, A.A. Voudouris, X. Wu, Fair division of indivisible goods: recent progress and open questions, Artif. Intell. 322 (2023) 103965.
[19] H. Aziz, B. Li, H. Moulin, X. Wu, Algorithmic fair allocation of indivisible items: a survey and new questions, SIGecom Exch. 20 (2022) 24–40.
[20] M. Chakraborty, A. Igarashi, W. Suksompong, Y. Zick, Weighted envy-freeness in indivisible item allocation, ACM Trans. Econ. Comput. 9 (2021) 18:1–18:39.
[21] W. Suksompong, Weighted fair division of indivisible items: a review, Inf. Process. Lett. 187 (2025) 106519.
[22] H. Aziz, Simultaneously achieving ex-ante and ex-post fairness, in: WINE, in: Lecture Notes in Computer Science, vol. 12495, Springer, 2020, pp. 341–355.
[23] R. Freeman, N. Shah, R. Vaish, Best of both worlds: ex-ante and ex-post fairness in resource allocation, in: EC, ACM, 2020, pp. 21–22.
[24] M. Babaioff, T. Ezra, U. Feige, On best-of-both-worlds fair-share allocations, in: WINE, in: Lecture Notes in Computer Science, vol. 13778, Springer, 2022, pp. 237–255.
[25] H. Aziz, R. Freeman, N. Shah, R. Vaish, Best of both worlds: ex ante and ex post fairness in resource allocation, Oper. Res. 72 (2024) 1674–1688.
[26] H. Aziz, A. Ganguly, E. Micha, Best of both worlds fairness under entitlements, in: AAMAS, ACM, 2023, pp. 941–948.
[27] M. Hoefer, M. Schmalhofer, G. Varricchio, Best of both worlds: agents with entitlements, in: AAMAS, ACM, 2023, pp. 564–572.
[28] J. Garg, A. Murhekar, J. Qin, New algorithms for the fair and efficient allocation of indivisible chores, in: IJCAI, ijcai.org, 2023, pp. 2710–2718.
[29] J. Garg, A. Murhekar, J. Qin, Fair and efficient allocations of chores under bivalued preferences, in: AAAI, AAAI Press, 2022, pp. 5043–5050.
[30] S. Ebadian, D. Peters, N. Shah, How to fairly allocate easy and difficult chores, in: AAMAS, International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS), 2022, pp. 372–380.
[31] H. Aziz, J. Lindsay, A. Ritossa, M. Suzuki, Fair allocation of two types of chores, in: AAMAS, ACM, 2023, pp. 143–151.
[32] J. Garg, A. Murhekar, J. Qin, Weighted EF1 and PO allocations with few types of agents or chores, in: IJCAI, ijcai.org, 2024, pp. 2799–2806.
[33] D. Bertsimas, V.F. Farias, N. Trichakis, The price of fairness, Oper. Res. 59 (2011) 17–31.
[34] I. Caragiannis, C. Kaklamanis, P. Kanellopoulos, M. Kyropoulou, The efficiency of fair division, Theory Comput. Syst. 50 (2012) 589–610.
[35] X. Bei, X. Lu, P. Manurangsi, W. Suksompong, The price of fairness for indivisible goods, Theory Comput. Syst. 65 (2021) 1069–1093.
[36] S. Barman, U. Bhaskar, N. Shah, Optimal bounds on the price of fairness for indivisible goods, in: WINE, in: Lecture Notes in Computer Science, vol. 12495, Springer, 2020, pp. 356–369.
[37] A. Sun, B. Chen, X.V. Doan, Connections between fairness criteria and efficiency for allocating indivisible chores, in: AAMAS, ACM, 2021, pp. 1281–1289.
[38] M. Chakraborty, E. Segal-Halevi, W. Suksompong, Weighted fairness notions for indivisible items revisited, ACM Trans. Econ. Comput. 12 (2024) 9:1–9:45.

[39] J. Garg, A. Murhekar, Computing fair and efficient allocations with few utility values, Theor. Comput. Sci. 962 (2023) 113932.
[40] G. Amanatidis, G. Birmpas, A. Filos-Ratsikas, A. Hollender, A.A. Voudouris, Maximum Nash welfare and other stories about EFX, Theor. Comput. Sci. 863 (2021) 69–85.
[41] E. Budish, The combinatorial assignment problem: approximate competitive equilibrium from equal incomes, J. Polit. Econ. 119 (2011) 1061–1103.
[42] M. Babaioff, T. Ezra, U. Feige, Fair-share allocations for agents with arbitrary entitlements, in: EC, ACM, 2021, p. 127.
[43] A. Farhadi, M. Ghodsi, M.T. Hajiaghayi, S. Lahaie, D.M. Pennock, M. Seddighin, S. Seddighin, H. Yami, Fair allocation of indivisible goods to asymmetric agents, J. Artif. Intell. Res. 64 (2019) 1–20.
[44] H. Aziz, H. Chan, B. Li, Weighted maxmin fair share allocation of indivisible chores, in: IJCAI, ijcai.org, 2019, pp. 46–52.
[45] U. Feige, X. Huang, On picking sequences for chores, in: EC, ACM, 2023, pp. 626–655.
[46] S. Brânzei, F. Sandomirskiy, Algorithms for competitive division of chores, Math. Oper. Res. (2023).
[47] M. Springer, M. Hajiaghayi, H. Yami, Almost envy-free allocations of indivisible goods or chores with entitlements, in: AAAI, AAAI Press, 2024, pp. 9901–9908.
[48] H. Aziz, I. Caragiannis, A. Igarashi, T. Walsh, Fair allocation of indivisible goods and chores, in: IJCAI, ijcai.org, 2019, pp. 53–59.
[49] M. Chakraborty, U. Schmidt-Kraepelin, W. Suksompong, Picking sequences and monotonicity in weighted fair division, Artif. Intell. 301 (2021) 103578.
[50] E. Budish, Y.-K. Che, F. Kojima, P. Milgrom, Designing random allocation mechanisms: theory and applications, Am. Econ. Rev. 103 (2013) 585–623.