Improved Approximation of Weighted MMS Fairness for Indivisible Chores

Fangxiao Wang[1] , **Bo Li**[1] , **Pinyan Lu**[2,3]
[1]Department of Computing, The Hong Kong Polytechnic University
[2] Institute for Theoretical Computer Science, Shanghai University of Finance and Economics
[3] Key Laboratory of Interdisciplinary Research of Computation and Economics (Shanghai University of Finance and Economics), Ministry of Education
fangxiao.wang@connect.polyu.hk, comp-bo.li@polyu.edu.hk, lu.pinyan@mail.shufe.edu.cn

Abstract

We study how to fairly allocate a set of indivisible chores among n agents who may have different weights corresponding to their involvement in completing these chores. We found that some of the existing fairness notions may place agents with lower weights at a disadvantage, which motivates us to explore weighted maximin share fairness (WMMS). While it is known that a WMMS allocation may not exist, no non-trivial approximation has been discovered thus far. In this paper, we first design a simple sequential picking algorithm that solely relies on the agents' ordinal rankings of the items, which achieves an approximation ratio of $O(\log n)$. Then, for the case involving two agents, we improve the approximation ratio to $\frac{\sqrt{3}+1}{2} \approx 1.366$, and prove that it is optimal. We also consider an online setting when the items arrive one after another and design an $O(\sqrt{n})$-competitive online algorithm given the valuations are normalized.

1 Introduction

The task in fair job scheduling is to allocate a set of jobs $\mathcal{M} = \{o_1, \ldots, o_m\}$ (called chores or items throughout this paper) to a set of agents $\mathcal{N} = \{a_1, \ldots, a_n\}$ in a fair manner, where every job has to be entirely allocated to exactly one agent. Each agent a_i has a valuation function $v_i : 2^{\mathcal{M}} \to \mathbb{R}_{\geq 0}$ to evaluate the cost of completing the jobs allocated to her. In this paper, we focus on additive valuations. In the general situation, and most likely what happens in reality, the agents have possibly different obligations or responsibilities in completing these jobs. For example, a person in a leadership position is naturally expected to undertake higher responsibility than the others. To model this asymmetry of the agents, each agent a_i is assumed to hold a share of $0 < w_i < 1$ over all jobs, where $\sum_{i=1}^{n} w_i = 1$, and w_i's are called the agents' weights in the system. Among the various fairness notions, which will be briefly reviewed in Section 1.2, proportional fairness is a remarkable one, which requires the allocation to respects the agents' shares. Formally, an allocation $(A_1, \ldots, A_n)$ is *proportional* (PROP), if $v_i(A_i) \leq w_i \cdot v_i(\mathcal{M})$ for all agents a_i [Steinhaus, 1948; Robertson and Webb, 1998].

PROP is ideal, but it is very hard to satisfy. For example, if $\mathcal{M}$ contains a single job and all agents have non-zero cost on it, no matter which agent receives it, the allocation is not fair to her. Accordingly, several ways of relaxing the requirements of PROP are proposed in the literature. For example, proportionality up to any item (PROPX) is studied in [Li *et al.*, 2022], where an allocation is PROPX if for all agents a_i, $v_i(A_i \setminus \{e\}) \leq w_i \cdot v_i(\mathcal{M})$ holds for any $e \in A_i$. The good news is that a PROPX allocation is always guaranteed to exist and can be found easily if the valuations are additive. Another popular relaxation of PROP is the maximin share (MMS) fairness, which was first proposed for allocating goods (where agents prefer to get more items) and symmetric agents, i.e., when $w_1 = \cdots = w_n = \frac{1}{n}$, by [Budish, 2010]. The intuition of MMS fairness is to relax the weight $\frac{1}{n}$ to a weaker share that is easy to satisfy. Let $\mathcal{A}$ be the set of all allocations, then the MMS of agent a_i is

$$\mathsf{MMS}_i = \min_{(A_1,\ldots,A_n)\in\mathcal{A}} \max_{j=1,\ldots,n} v_i(A_j),$$

which is the minimum of the maximum cost for a_i in any n-partition of the chores. Clearly, $\mathsf{MMS}_i \geq \frac{1}{n}$ and thus agent a_i is satisfied if her cost is no greater than MMS_i. Although there are still, but rare, instances for which the MMS value cannot be guaranteed for every agent [Aziz *et al.*, 2017; Feige *et al.*, 2021], there are constant approximations [Huang and Lu, 2021; Huang and Segal-Halevi, 2023]. To generalize this share-based notion to asymmetric agents (when the agents have non-identical weights), AnyPrice share (APS) fairness was proposed in [Babaioff *et al.*, 2021]. The definition of APS is slightly more complicated; informally, APS_i defines a share for every agent a_i which is the maximum effort she needs to pay when her loan to the system equals her weight w_i and she completes a least painful set of items to repay her loan when the jobs are adversarially priced with a total price of 1. Since APS is not easy to understand, a simple notion, chore share (CS), is introduced in [Huang and Segal-Halevi, 2023], which provides a convenient replacement and a lower bound on the APS. For agent a_i with weight w_i,

$$\mathsf{CS}_i = \max\{w_i \cdot v_i(\mathcal{M}), v_i(\{o_1\}), v_i(\{o_k, o_{k+1}\})\},$$

where o_1, o_k, o_{k+1} are the items with the 1-st, k-th and $(k+1)$-th highest values and $k = \lfloor \frac{1}{w_i} \rfloor$. If m is not sufficiently large, $v_i(\{o_k, o_{k+1}\})$ can be dropped from the $\max$ operator.

agent	weight	item o_1	item o_2
a_1	$w_1 = 1 - \epsilon$	0.5	0.5
a_2	$w_2 = \epsilon$	0.5	0.5

Table 1: An instance with two agents and two items.

Consider a simple instance of allocating two identical items to two agents with $w_1 = 1 - \epsilon$, $w_2 = \epsilon$, where $\epsilon > 0$ is a sufficiently small number. The values are shown in Table 1. It can be verified that, $\mathsf{APS}_2 = \mathsf{CS}_2 = 0.5$, which means if we only allocate one item to agent a_2 (and the other item to a_1), a_2 would accept this allocation since the item's value is also 0.5, i.e., the allocation is APS and CS fair. The allocation is also PROPX, since by removing the item from a_2's allocation, her bundle is empty. However, when ϵ approaches 0, we may not regard this allocation as a fair one because agent a_2 bears little responsibility. This drawback is commonly observed in all the aforementioned notions: if an agent receives a single item, they would consider the allocation fair, regardless of the item's value.

In fact, in the above instance, there is an allocation that is fairer when ϵ is small – allocating both items to agent a_1. Agent a_1 understands that she is expected to take $1 - \epsilon$ fraction of all the items, and allocating both items to her is not too far away from a PROP one; however, if one of the items is given to a_2, agent a_2's value is $0.5 \gg \epsilon$ (even so in a_1's perspective), which is far from a fair allocation. This intuition has been formalized as *weighted maximin share fairness* (WMMS) defined in [Aziz *et al.*, 2019]. Given an allocation $(A_1, \ldots, A_n)$, we define the unfairness ratio of agent a_i as

$$\max_{j=1,\ldots,n} \frac{v_i(A_j)}{w_j},$$

and then the "fairest" allocation is to minimize the unfairness ratio. Thus, the weighted MMS of a_i is her weight times the smallest unfairness ratio, i.e.,

$$\mathsf{WMMS}_i = w_i \cdot \min_{\mathbf{A} \in \mathcal{A}} \max_{A_j \in \mathbf{A}} \frac{v_i(A_j)}{w_j}.$$

It is easy to see that when all agents have the same weight, WMMS_i is exactly MMS_i. An allocation is WMMS fair if all agents' values are no greater than their WMMS.

Recall the example in Table 1,

$$\mathsf{WMMS}_1 = (1 - \epsilon) \cdot \min\{\frac{v_1(\{o_1, o_2\})}{1 - \epsilon}, \frac{v_1(o_1)}{\epsilon}\} = 1,$$

and

$$\mathsf{WMMS}_2 = \epsilon \cdot \min\{\frac{v_2(\{o_1, o_2\})}{1 - \epsilon}, \frac{v_2(o_1)}{\epsilon}\} = \frac{\epsilon}{1 - \epsilon}.$$

By the definition of WMMS_i, we need to enumerate all possible allocations to find the smallest unfairness ratio. But in the example, it is clear that this ratio appears between allocating both items to agent a_1 and allocating one of the items, say o_1, to agent a_2. When $\epsilon \to 0$, we can see that WMMS_i is closer to the proportionality w_i, and the only WMMS fair allocation is to allocated both items to agent a_1. Moreover, in this case, any finite approximation of WMMS would allocate both items to agent a_1.

Then our question is if we can guarantee (approximate) WMMS fairness for all instances. A trivial solution is to allocate all items to the agent with the largest weight, which achieves n-approximate WMMS, but beyond that nothing is known. In this paper, we show how to improve the approximation ratio via simple sequential algorithms, and also investigate the inherit difficulties of approximating WMMS.

1.1 Our Contribution

Our main results can be summarized as follows.

It is proved in [Aziz *et al.*, 2019] that allocating all items to the agent with the highest weight achieves n-approximation of WMMS, but no non-trivial approximation ratio beyond n has been proved thus far. Our first contribution is a simple sequential picking algorithm that improves the approximation ratio to $O(\log n)$, where the algorithm only accesses each agent's ordinal ranking of the items. Informally, the algorithm uses the weights of the agents to divide the run of the algorithm in rounds. The agents with higher weights will join the algorithm in earlier rounds and thus receive more items. Within each round, we fractionally allocate the items to the agents proportional to their weights, and show that such a fractional allocation can be converted to an integral one without loss of much approximation ratio, using a rounding technique in [Feige and Huang, 2023]. By carefully choosing all parameters, we show that no agent obtains cost higher than $O(\log n)$ times of her WMMS. Although the analysis of our algorithm is not trivial, given these parameters, the implementation of the algorithm is straightforward.

Main Result 1. There exists an $O(\log n)$-approximate WMMS allocation for all instances with additive valuations.

We next focus on a typical case of two agents. Intuitively, if the two agents possess similar weights, the divide-and-choose algorithm should yield favorable results. Conversely, if one agent has a significantly larger weight than the other, allocating all items to that agent should also produce a satisfactory allocation. However, the challenge lies in the situations that fall between these two extremes. Through a more intricate analysis and by carefully distinguishing these three cases, we show that we can achieve $\frac{\sqrt{3}+1}{2} \approx 1.366$ approximation. Further, we prove that this is the optimal approximation ratio an algorithm can reach, surpassing the previously proven inapproximability of $\frac{4}{3} \approx 1.33$ in [Aziz *et al.*, 2019].

Main Result 2. For the case of two agents, the optimal approximation ratio of WMMS fairness is $\frac{\sqrt{3}+1}{2}$.

This result also shows the distinction between weighted and unweighted cases, where 1.182-MMS allocation exists [Huang and Segal-Halevi, 2023] when the agents have identical weights for arbitrary number of agents and an MMS allocation exists when there are two agents.

Finally, we consider the online setting when the items arrive one after another and the algorithm does not know the information on the future events. The problem has been studied in [Zhou *et al.*, 2023] when the agents are symmetric. It is shown that when the valuations are normalized, constant competitive ratios can be achieved. It is also known that if the valuations are not normalized, no online algorithm can be better than $\Omega(n)$-competitive. We follow this trend and consider the general weighted case, and prove that we can achieve a competitive ratio of $O(\sqrt{n})$ via a simple greedy algorithm. Although we do not view this as a main result of the paper, it may show some insights in further improving the approximation ration in the offline setting. As mentioned in Main Result 1, the algorithm there only relies on the agents' weights and their ordinal valuations. We suspect that using the cardinal values (more information than an ordering) can give a better approximation ratio. The algorithm here shows one possible way to utilize the values.

1.2 Related Works

Fair division is a multidisciplinary field that intersects computer science, economics, and mathematics. It remains a subject of ongoing debate regarding its fundamental concepts. Among the various fairness notions, two extensively studied and widely accepted ones are proportionality (PROP) [Steinhaus, 1948] and envy-freeness (EF) [Foley, 1967; Varian, 1973]. However, when it is not possible to divide the items, satisfying PROP and EF is not always feasible. Consequently, relaxations of these notions have been proposed, including MMS [Budish, 2010], PROP1 [Conitzer *et al.*, 2017], PROPX [Aziz *et al.*, 2020], EF1 [Budish, 2010; Lipton *et al.*, 2004], and EFX [Gourvès *et al.*, 2014; Caragiannis *et al.*, 2019]. MMS fairness was first introduced in [Budish, 2010] for goods and later [Kurokawa *et al.*, 2018] proved that MMS may not exist in some instances but a $2/3$-approximation always exists. The approximation ratio is later improved or extended to more general valuations in a series of works, e.g., [Ghodsi *et al.*, 2018; Barman and Krishnamurthy, 2020; Garg and Taki, 2020], and the best-known approximation is $(3/4 + 3/3836)$ so far, given in [Akrami and Garg, 2024].

[Aziz *et al.*, 2017] extended the MMS fairness to chores, and proved that it may not be satisfiable for all additive valuations but a 2-approximate MMS allocation can be easily found. Later the approximation ratio is improved in [Barman and Krishnamurthy, 2020; Huang and Lu, 2021; Huang and Segal-Halevi, 2023] and generalized to more general valuations in [Li *et al.*, 2023]. The best-known approximation for additive valuations is $13/11$ [Huang and Segal-Halevi, 2023]. For a more comprehensive introduction to these concepts, interested readers can refer to a recent survey [Amanatidis *et al.*, 2023].

In recent years, most of the concepts mentioned earlier have been extended to the *weighted* versions that accommodate asymmetric agents. For goods, while a weighted PROP1 (PROP after removing some item) allocation exists but a weighted PROPX (PROP after removing any item) allocation may not exist [Aziz *et al.*, 2020]. In contrast, for chores, a weighted PROPX (implying weighted PROP1) allocation exists and can be found easily [Li *et al.*, 2022]. For both goods and chores, a weighted EF1 allocation can be computed in polynomial time [Wu *et al.*, 2023], but the existence of weighted EFX allocations is not guaranteed [Hajiaghayi *et al.*, 2023]. Regarding weighted MMS, Farhadi et al. [Farhadi *et al.*, 2019] proved that the best possible approximation ratio is $\frac{1}{n}$ for goods, and Aziz [Aziz *et al.*, 2019] investigated the problem for chores but the best possible approximation ratio is still unknown. Apart from the aforementioned fairness criteria, there are additional notions, such as AnyPrice share (APS) [Babaioff *et al.*, 2021; Feige and Huang, 2023] and maximin aware (MMA) up to one or any [Wei *et al.*, 2023] for which constant approximations are allowed for chores.

2 Preliminaries

2.1 Model

We first formally introduce our problem. For any integer $k \geq 1$, let $[k] = \{1, \ldots, k\}$. For any set S and $e \in S$, let $S_{-e} = S \setminus \{e\}$. In a fair allocation instance, there are n agents and m indivisible chores (called items for convenience), denoted by $\mathcal{N} = \{a_1, \ldots, a_n\}$ and $\mathcal{M} = \{o_1, \ldots, o_m\}$, respectively. The agents have asymmetric shares for completing the items, and let $w_i > 0$ represent agent a_i's share (also called weight). Without loss of generality, the weights are normalized, i.e., $\sum_{a_i \in \mathcal{N}} w_i = 1$, and assume $w_1 \geq \cdots \geq w_n$. Denote $\mathbf{w} = (w_1, \ldots, w_n)$. Each agent a_i has a valuation function $v_i : 2^{\mathcal{M}} \to \mathbb{R}_{\geq 0}$, where $v_i(S)$ represents her cost on completing the items in S. For simplicity, when a bundle contains a single item o, denote $v_i(o) = v_i(\{o\})$. In this paper, we only consider additive valuation functions, i.e., $v_i(S) = \sum_{o_j \in S} v_i(o_j)$ for every $S \subseteq \mathcal{M}$. Let $\mathbf{v} = (v_1, \ldots, v_n)$. We sometimes assume, without loss of generality, that the valuations are normalized, i.e., $v_i(\mathcal{M}) = 1$ for all agents a_i. In summary, a chore allocation instance is represented by $\mathcal{I} = (\mathcal{N}, \mathcal{M}, \mathbf{w}, \mathbf{v})$.

An *allocation*, denoted by $\mathbf{A} = (A_1, \ldots, A_n)$, is an ordered n-partition of $\mathcal{M}$ where A_i is the set of items allocated to agent a_i such that $\bigcup_{a_i \in \mathcal{N}} A_i = \mathcal{M}$ and $A_i \cap A_j = \emptyset$ for all $i \neq j$. Denote by $\mathcal{A}$ the set of all allocations. An allocation is called *partial* if $\bigcup_{a_i \in \mathcal{N}} A_i \subsetneq \mathcal{M}$. Next, we introduce the solution concepts.

Definition 1 (WMMS). *Given any chore allocation instance* $\mathcal{I} = (\mathcal{N}, \mathcal{M}, \mathbf{w}, \mathbf{v})$*, for agent* $a_i \in \mathcal{N}$*, its* weighted maximin share *(WMMS), denoted by* $\mathsf{WMMS}_i(\mathcal{I})$*, is defined as*

$$\mathsf{WMMS}_i(\mathcal{I}) = w_i \cdot \min_{\mathbf{A} \in \mathcal{A}} \max_{A_j \in \mathbf{A}} \frac{v_i(A_j)}{w_j}.$$

When the instance is clear from the context, $\mathsf{WMMS}_i(\mathcal{I})$ *is also written as* WMMS_i *for simplicity.*

An partition $A_1, \ldots, A_n$ of $\mathcal{M}$ is called a WMMS-defining partition for agent a_i, if

$$\frac{v_i(A_j)}{w_j} \cdot w_i \leq \mathsf{WMMS}_i, \text{ for all } j = 1, \ldots, n.$$

A simple observation from the definition of WMMS is that

$$\mathsf{WMMS}_i \geq w_i \cdot v_i(\mathcal{M}), \text{ for all } j = 1, \ldots, n. \quad (1)$$

This is because that for any partition $B_1, \ldots, B_n$ of $\mathcal{M}$, there must be some j such that $v_i(B_j) \geq w_j \cdot v_i(\mathcal{M})$ since the weights of the agents are normalized.

Definition 2 (α-WMMS). *Given* $\alpha \geq 1$*, an allocation* $\mathbf{A} = (A_1, \ldots, A_n)$ *is called* α*-approximate WMMS fair (*α*-WMMS), if* $v_i(A_i) \leq \alpha \cdot \mathsf{WMMS}_i$ *for all agents* $a_i \in \mathcal{N}$*.*

2.2 Proportional and Ordered Instances

We next introduce some properties of the fair allocation instances, which will be used to simplify our analysis.

We start with additional definitions. A fair allocation instance $\mathcal{I} = (\mathcal{N}, \mathcal{M}, \mathbf{w}, \mathbf{v})$ is called *ordered* if all agents have the same ranking over the items, i.e., $v_i(o_1) \geq v_i(o_2) \geq \cdots \geq v_i(o_m)$ for all $a_i \in \mathcal{N}$. It is widely known that the hardest case to approximate WMMS fairness is the ordered instances. The following lemma has been proved, for example, in [Barman and Krishnamurthy, 2020; Huang and Lu, 2021; Li *et al.*, 2022].

Lemma 1. *For any $\alpha \geq 1$, if there is an algorithm Ψ_1 that returns an α-WMMS allocation for all ordered instances, then there is another algorithm Ψ_2 that ensures an α-WMMS allocation for all instances.*

It deserves to note that in Lemma 1, if Ψ_1 is a polynomial-time algorithm, then Ψ_2 also runs in polynomial time.

Next, given any normalized instance, a valuation function v_i is called *proportional* if $\mathsf{WMMS}_i = w_i$, i.e., $\mathcal{M}$ can be partitioned into n bundles $A_1, \ldots, A_n$ such that $v_i(A_j) = w_j$ for $j = 1, \ldots, n$. The instance is called *proportional* if $\mathsf{WMMS}_i = w_i$ for all $a_i \in \mathcal{N}$. We prove that, similar as Lemma 1, to approximate WMMS fairness, it suffices to focus on the proportional instances.

Lemma 2. *For any $\alpha \geq 1$, if there is an algorithm Ψ_1 that returns an α-WMMS allocation for all proportional instances, then there is another algorithm Ψ_2 that ensures an α-WMMS allocation for all instances.*

Proof. Given the algorithm Ψ_1, we design the algorithm Ψ_2 as follows. For any normalized instance $\mathcal{I} = (\mathcal{N}, \mathcal{M}, \mathbf{w}, \mathbf{v})$, Ψ_2 first constructs a proportional instance $\mathcal{I}' = (\mathcal{N}, \mathcal{M}', \mathbf{w}, \mathbf{v}')$. For each agent a_i, let $A_1^i, \ldots, A_n^i$ be an arbitrary WMMS-defining partition of her and construct n extra items, denoted by $\mathcal{M}_i = \{o_1^i, \ldots, o_n^i\}$, such that for $j = 1, \ldots, n$,

$$v_i'(o_j^i) = \frac{\mathsf{WMMS}_i(\mathcal{I})}{w_i} \cdot w_j - v_i(A_j^i).$$

Since $A_1^i, \ldots, A_n^i$ is a WMMS-defining partition, we have $\frac{v_i(A_j^i)}{w_j} \leq \frac{\mathsf{WMMS}_i(\mathcal{I})}{w_i}$ and thus $v_i'(o_j^i) \geq 0$. Let $v_l'(o_j^i) = 0$ for all $l \neq i$. Let $\mathcal{M}' = \mathcal{M} \cup \mathcal{M}_1 \cup \cdots \cup \mathcal{M}_n$ and $v_i'(o) = v_i(o)$ for all $o \in \mathcal{M}$ and $a_i \in \mathcal{N}$. Accordingly, we have

$$v_i'(A_j \cup \{o_j^i\}) = \frac{\mathsf{WMMS}_i(\mathcal{I})}{w_i} \cdot w_j \text{ for all } j = 1, \ldots, n$$

and thus

$$\frac{v_i'(A_1^i \cup \{o_1^i\})}{w_1} = \cdots = \frac{v_i'(A_n^i \cup \{o_n^i\})}{w_n} = \frac{\mathsf{WMMS}_i(\mathcal{I})}{w_i}.$$

That is $A_1^i \cup \{o_1^i\}, \ldots, A_n^i \cup \{o_n^i\}$, plus $\mathcal{M}_{-i}$ being arbitrarily distributed among these n bundles, is a WMMS-defining partition of v_i' with items $\mathcal{M}'$ and $\mathsf{WMMS}_i(\mathcal{I}') = \mathsf{WMMS}_i(\mathcal{I})$. We remark that we do not normalize the valuations of $\mathcal{I}'$ to make the connection between $\mathcal{I}$ and $\mathcal{I}'$ evident.

Suppose $\mathbf{A} = (A_1, \ldots, A_n)$ is an α-WMMS allocation of instance $\mathcal{I}'$ returned by algorithm Ψ_1. For $i = 1, \ldots, n$, let $B_i = A_i \cap \mathcal{M}$ and we have

$$v_i(B_i) \leq v_i'(A_i) \leq \alpha \cdot \mathsf{WMMS}_i(\mathcal{I}') = \alpha \cdot \mathsf{WMMS}_i(\mathcal{I}).$$

Thus $\mathbf{B} = (B_1, \ldots, B_n)$ is an α-WMMS allocation of $\mathcal{I}$, which completes the construction of Ψ_2. □

Remark The construction of instance $\mathcal{I}'$ in the proof of Lemma 2 does not run in polynomial time, which means Ψ_2 may not be a polynomial-time algorithm even if Ψ_1 is. By incorporating the rounding method used in [Aziz *et al.*, 2019], our algorithm can be transformed to run in polynomial time, albeit with a constant factor loss in the approximation ratio.

For any instance, we first reduce it to a proportional instance, then reduce it to an ordered one, which is still proportional. Due to Lemmas 1 and 2, in the remaining of this paper, without loss of generality, it suffices to restrict our attention on the ordered and proportional instances.

Finally, we present the following technical lemma, which will be used to design our algorithms. It is easy to see that since agent a_1 has the largest weight and the instance is proportional and ordered, we must have $v_1(o_1) \leq w_1$; otherwise, $\mathsf{WMMS}_1 > w_1$ no matter which bundle contains o_1 in a_1's WMMS-defining partition. In general, we have the following property for all agents a_i and $i \geq 2$.

Lemma 3. *For any proportional and ordered instance, given any agent a_i with $i \geq 2$, if $\sum_{l \in [i-1]} \lfloor \frac{w_l}{w_i} \rfloor < m$,*

$$v_i(o_j) \leq w_i \text{ for all } j > \sum_{l \in [i-1]} \lfloor \frac{w_l}{w_i} \rfloor.$$

Proof. Given any WMMS-defining partition $B_1, \ldots, B_n$ for agent a_i, B_l cannot contain any item whose value is greater than w_i for any $l \geq i$ since the instance is proportional. Moreover, as $v_i(B_l) = w_l$ for all $l < i$, B_l contains at most $\lfloor \frac{w_l}{w_i} \rfloor$ items whose value is greater than w_i. Thus in total the number of such large items is at most $\sum_{l \in [i-1]} \lfloor \frac{w_l}{w_i} \rfloor$. Noting that the instance is ordered, we have the lemma. □

3 The Main Algorithm

In this section, we prove our main result, where an $O(\log n)$-WMMS allocation can be found in polynomial time.

3.1 Fractional Allocations

Before designing our main algorithm, we first introduce a technical lemma which can further simplify the description of the algorithm. An allocation is fractional if some items are divided and allocated among agents. Formally, denote by $0 \leq x_{ij} \leq 1$ the fraction of item o_j that is allocated to agent a_i, and let $\mathbf{x}_i = (x_{i1}, \ldots, x_{im})$. Denote by $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ a fractional allocation where it is required that $\sum_{a_i \in \mathcal{N}} x_{ij} = 1$ for all $o_j \in \mathcal{M}$. The following lemma, proved in [Feige and Huang, 2023], shows that we can convert a fractional allocation to an integral one without sacrificing much on the approximation ratio of WMMS, if (1) the fractional allocation is a good approximation of WMMS and (2) the value

Algorithm 1 fractional allocation $\to$ integral allocation

Input: A fractional allocation $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_n)$ for ordered instance $\mathcal{I} = (\mathcal{M}, \mathcal{N}, \mathbf{w}, \mathbf{v})$.
Output: An integral allocation $\mathbf{A} = (A_1, \dots, A_n)$.

1: Initialize that $A_i = \emptyset$ for $a_i \in \mathcal{N}$.
2: **for** $j = 1$ to m **do**
3: Choose one agent a_i for whom $|A_i| < \sum_{k=1}^{j} x_{ik}$.
4: $A_i \leftarrow A_i \cup \{o_j\}$.
5: **end for**
6: **return** $\mathbf{A} = (A_1, \dots, A_n)$.

of the largest item each agent positively receives is not much larger than her WMMS. We include the proof of Lemma 4 for completeness.

Lemma 4. [Feige and Huang, 2023] *Given any ordered instance $\mathcal{I} = (\mathcal{M}, \mathcal{N}, \mathbf{w}, \mathbf{v})$ and a fractional allocation $\mathbf{x}$, where for every agent $a_i \in \mathcal{N}$, $v_i(\mathbf{x}_i) \le \alpha \cdot \mathsf{WMMS}_i$ and $v_i(o_j) \le \beta \cdot \mathsf{WMMS}_i$ for all $o_j \in \mathcal{M}$ such that $x_{ij} > 0$, Algorithm 1 returns an integral $(\alpha + \beta)$-WMMS allocation.*

Proof. First, observe that Algorithm 1 is well-defined, i.e., for each iteration $j = 1, \dots, m$, there exists at least one agent a_i such that $|A_i| < \sum_{k=1}^{j} x_{ij}$. This can be seen by induction. Suppose Algorithm 1 successfully reaches the j-th iteration, and thus exactly $j - 1$ items have been allocated. Let $(A_1^j, \dots, A_n^j)$ be the current partial allocation. Since $\mathbf{x}$ is a complete allocation, we have

$$\sum_{a_i \in \mathcal{N}} |A_i^j| = j - 1 < j = \sum_{i \in \mathcal{N}} \sum_{k=1}^{j} x_{ik},$$

and thus $|A_i| < \sum_{k=1}^{j} x_{ik}$ for at least one agent a_i.

Next we bound the value of $v_i(A_i)$ for an arbitrary agent a_i. If $x_{ij} = 0$ for all item $o_j \in \mathcal{M}$, then a_i cannot be selected in Step 3 for any iteration j. Thus $A_i = \emptyset$ and $v_i(A_j) = 0$, which means the lemma holds trivially. Otherwise, let f_i be the first item for which $x_{if_i} > 0$, then $v_i(o) \le v_i(o_{f_i})$ for any $o \in A_i$. Suppose $A_i = \{o_{j_1}, o_{j_2}, \dots, o_{j_p}\}$. Then $j_1 \ge f_i$ and $v_i(o_{j_1}) \le v_i(o_{f_i})$. Moreover, for any $l \ge 2$, o_{j_l} can be added to A_i only when $\sum_{k \le j_l} x_{ik} > l - 1$ and o_{j_l} has smallest value among all the items before o_{j_l}. Accordingly $v_i(A_i \setminus \{o_{j_1}\}) \le v_i(\mathbf{x}_i)$ and thus

$$\begin{aligned} v_i(A_i) = v_i(A_i \setminus \{o_{j_1}\}) + v_i(o_{j_1}) &\le v_i(\mathbf{x}_i) + v_i(o_{f_i}) \\ &\le (\alpha + \beta) \cdot \mathsf{WMMS}_i, \end{aligned}$$

which completes the proof of the lemma. □

3.2 The Algorithm

The goal of our algorithm is to find a fractional allocation $\mathbf{x}$, where for any agent $a_i \in \mathcal{N}$, $v_i(\mathbf{x}_i) \le \log n \cdot \mathsf{WMMS}_i$ and $v_i(o_j) \le \mathsf{WMMS}_i$ for all o_j such that $x_{ij} > 0$. By Lemma 4, $\mathbf{x}$ can be easily converted into an integral allocation $\mathbf{A}$ that is $(\log n + 1)$-WMMS. Recall that it suffices for us to focus on proportional and ordered instances. Given any such instance $\mathcal{I} = (\mathcal{N}, \mathcal{M}, \mathbf{w}, \mathbf{v})$ with $w_1 \ge \dots \ge w_n$, we categorize the items by $n + 1$ disjoint groups $\mathcal{G} = \{G_0, G_1, \dots, G_n\}$:

$$\begin{aligned} &G_0 = \emptyset; \\ &G_1 = \left\{ o_j \in \mathcal{M} \mid j \le \lfloor \frac{w_1}{w_2} \rfloor \right\}; \\ &G_i = \left\{ o_j \in \mathcal{M} \mid j \le \sum_{l \in [i]} \lfloor \frac{w_l}{w_{i+1}} \rfloor \right\} \setminus \bigcup_{l < i} G_l, \qquad (2) \\ &\quad \text{for all } 1 < i < n; \\ &G_n = \mathcal{M} \setminus \bigcup_{l < n} G_l. \end{aligned}$$

Note that G_l may be empty for $l \ge 1$ if there are not enough items in $\mathcal{M}$. Our algorithm ensures that the items in each G_i, $1 \le i \le n$, are only allocated to agents in $\{a_1, \dots, a_i\}$, and thus by Lemma 3, the value of each single item allocated to an agent is no greater than her weight. In the following, we show how to ensure everyone's value for her assigned items to be no greater than $\log n$ times her weight.

The algorithm is described in Algorithm 2. The key idea is to allocate the items in each group G_i to agents $\{a_1, \dots, a_i\}$ proportionally to their weights. Given any $\mathcal{N}, \mathcal{M}, \mathbf{w}$, the fractional allocation $(x_1, \dots, x_n)$ is fixed no matter what the valuation profile $\mathbf{v}$ is. To bound the approximation ratio of the algorithm, we next show that, agent a_1, who has the largest weight, is the worst-off one in the algorithm, and her worst-case valuation is when it aligns the weights of the agents, i.e., $v_{1j} = w_j$ for $1 \le j \le n$ and $v_{1j} = 0$ for $j > n$. Formally, let v_1' be the worst-case valuation, i.e., the proportional valuation v_i that maximizes the value of $\frac{v_1(\mathbf{x}_1)}{w_1}$.

Lemma 5. *Given any $\mathcal{N}, \mathcal{M}, \mathbf{w}$, for any agent $a_i \in \mathcal{N}$ and any proportional valuation v_i,*

$$\frac{v_i(\mathbf{x}_i)}{w_i} \le \frac{v_1'(\mathbf{x}_1)}{w_1} \le \frac{\sum_{j=1}^{n} x_{1j} \cdot w_j}{w_1}.$$

Proof. To prove the first inequality, it suffices to consider $i \ge 2$. Let $\mathbf{x}_1'$ be the projected allocation of $\mathbf{x}_i$ on groups $G_i, \dots, G_n$, i.e., $x_{1j}' = 0$ for all $o_j \in G_1 \cup \dots \cup G_{i-1}$ and $x_{1j}' = x_{1j}$ otherwise. Then

$$\frac{v_i(\mathbf{x}_1)}{w_1} \ge \frac{v_i(\mathbf{x}_1')}{w_1} = \frac{v_i(\mathbf{x}_i)}{w_i},$$

where the equality is because the items in $G_i, \dots, G_n$ are proportionally allocated among the agents according to their

Algorithm 2 The main algorithm

Input: A proportional and ordered instance $\mathcal{I} = (\mathcal{M}, \mathcal{N}, \mathbf{w}, \mathbf{v})$.
Output: A fractional allocation $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_n)$.

for $l = 1$ to n **do**
  for $i = 1$ to l **do**
    For each $o_j \in G_l$, set $x_{ij} = \frac{w_i}{\sum_{k \in [l]} w_k}$.
  end for
end for
return $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_n)$.

weights. Since v'_1 maximizes the value of $\frac{v_1(\mathbf{x}_1)}{w_1}$,

$$\frac{v_i(\mathbf{x}_i)}{w_i} \leq \frac{v_1(\mathbf{x}_1)}{w_1} \leq \frac{v'_1(\mathbf{x}_1)}{w_1}.$$

To prove the second inequality, it suffices to show $v'_1(\mathbf{x}_1) = \sum_{j=1}^{m} x_{ij} \cdot v'_1(o_j) \leq \sum_{j=1}^{n} x_{1j} \cdot w_j$. Recall that $v'_1(\cdot)$ is proportional, i.e., $\mathsf{WMMS}_1 = w_1$ or equivalently the items can be partitioned into n bundles whose values are exactly $w_1, \ldots, w_n$. Then we must have that for any $1 \leq j \leq n$,

$$\sum_{l=1}^{j} v'_1(o_l) \leq \sum_{l=1}^{j} w_l.$$

This is because, otherwise, to ensure proportionality, any WMMS-defining partition would assign at least one item in $\{o_1, \ldots, o_j\}$ (whose value is greater than w_j) to a bundle with value in $w_{j+1}, \ldots, w_n$, which also contradicts $v_1(\cdot)$ being proportional. Note that by the design of the algorithm, $x_{11} \geq x_{12} \geq \cdots \geq x_{1m}$, and thus we have

$$(x_{1j} - x_{1,j+1}) \cdot \sum_{l=1}^{j} v'_1(o_l) \leq (x_{1j} - x_{1,j+1}) \cdot \sum_{l=1}^{j} w_l, \quad (3)$$

for all $j < n$ and

$$x_{1j} \cdot \sum_{l=1}^{j} v'_1(o_l) \leq x_{1j} \cdot \sum_{l=1}^{j} w_l \quad (4)$$

for all $j \leq n$. Therefore, summing up Inequality 3 for $j = 1, \ldots, n-1$ and Inequality 4 for $j = n$, we have

$$\sum_{l=1}^{n} x_{1l} \cdot v'_1(o_l) \leq \sum_{l=1}^{n} x_{1l} \cdot w_l.$$

By the linearity and normalization of the valuation function, assigning positive value to items in $\{o_{n+1}, \ldots, o_m\}$ can only make the total cost to be smaller. Thus

$$\sum_{l=1}^{m} x_{1l} \cdot v'_1(o_l) \leq \sum_{l=1}^{n} x_{1l} \cdot w_l,$$

which completes the proof of the lemma. □

It remains to bound the value of $\frac{\sum_{j=1}^{n} x_{1j} \cdot w_j}{w_1}$, which gives the approximatio ratio.

Lemma 6. *For any instance $\mathcal{I} = (\mathcal{N}, \mathcal{M}, \mathbf{w}, \mathbf{v})$, we have*

$$\frac{\sum_{j=1}^{n} x_{1j} \cdot w_j}{w_1} = \sum_{j=1}^{n} \frac{x_{1j} \cdot w_j}{w_1} < \log n.$$

Proof. For each $j = 1, \ldots, n$, let G_{i_j} be the group that contains o_j; recall the definition of G_i in Equation 2. If $j \leq i_j$,

$$x_{1j} = \frac{w_1}{\sum_{k \in [i_j]} w_k} \leq \frac{w_1}{\sum_{k \in [j]} w_k} \leq \frac{w_1}{j \cdot w_j},$$

and thus,

$$\frac{x_{1j} \cdot w_j}{w_1} \leq \frac{1}{j}.$$

Actually, j cannot be strictly smaller than i_j by the definition of G_i's, but it does not affect the analysis.

If $j > i_j$, by the definition of G_{i_j}, we have

$$j \leq \sum_{k \in [i_j]} \lfloor \frac{w_k}{w_{i_j+1}} \rfloor \leq \sum_{k \in [i_j]} \lfloor \frac{w_k}{w_j} \rfloor,$$

and equivalently, $j \cdot w_j \leq \sum_{k \in [i_j]} w_k$. Therefore,

$$\frac{x_{1j} \cdot w_j}{w_1} = \frac{w_1}{\sum_{k \in [i_j]} w_k} \cdot \frac{w_j}{w_1} \leq \frac{w_1}{j \cdot w_j} \cdot \frac{w_j}{w_1} = \frac{1}{j}.$$

Combining both cases and summing up for all j,

$$\sum_{j=1}^{n} \frac{x_{1j} \cdot w_j}{w_1} \leq \sum_{j=1}^{n} \frac{1}{j} < \log n,$$

which completes the proof. □

Combining the above lemmas, we have the theorem, whose proof is omitted.

Theorem 1. *For any fair allocation instance, there is an $O(\log n)$-approximate WMMS allocation.*

Finally, we note that our analysis is tight even when $m = n$, all agents have the same weight, and all the items are identical to the agents. This is because the fractional allocation returned by Algorithm 2 allocates $\frac{1}{i}$ fraction of item o_i to agent a_1 for each $i = 1, \ldots, n$, and then Algorithm 1 realizes an integral allocation where a_1 gets $\log n$ complete items. However, any WMMS allocation allocates only one item to a_1.

4 Tight Bound for 2 Agents

In this section, we restrict our focus on the case of two agents and prove the following result.

Theorem 2. *When $n = 2$, the optimal approximation ratio of WMMS is $\frac{\sqrt{3}+1}{2} \approx 1.366$.*

We prove Theorem 2 through Lemmas 7 and 8 in the subsequent two subsections.

4.1 Lower Bound

Lemma 7. *When $n = 2$, no algorithm can be better than $\frac{\sqrt{3}+1}{2}$-WMMS.*

Proof. We construct the following instance with two agents a_1 and a_2 and three items o_1, o_2, o_3. For agent a_1, $w_1 = \sqrt{3}-1$,$v_{1,1} = \sqrt{3}-1$, $v_{1,2} = 2-\sqrt{3}$, $v_{1,3} = 0$. For agent a_2, $w_2 = 2-\sqrt{3}$, $v_{2,1} = \frac{\sqrt{3}-1}{2}$, $v_{2,2} = \frac{\sqrt{3}-1}{2}$, $v_{2,3} = 2-\sqrt{3}$. Obviously, $\mathsf{WMMS}_1 = \sqrt{3}-1$ and $\mathsf{WMMS}_2 = 2-\sqrt{3}$.

If agent a_2 get item o_1 or o_2, the approximation ratio is at least $\frac{\sqrt{3}+1}{2}$. While if agent a_2 doesn't get item o_1 and o_2, it means agent a_1 get item o_1 and o_2, the approximation ratio is also at least $\frac{\sqrt{3}+1}{2}$. So the lower bound is $\frac{\sqrt{3}+1}{2}$. □

agent	weight	item o_1	item o_2	item o_3
a_1	$w_1 = \sqrt{3} - 1$	$\sqrt{3} - 1$	$2 - \sqrt{3}$	0
a_2	$w_2 = 2 - \sqrt{3}$	$\frac{\sqrt{3}-1}{2}$	$\frac{\sqrt{3}-1}{2}$	$2 - \sqrt{3}$

Table 2: An instance with two agents and three items.

4.2 Upper Bound

Lemma 8. *When $n = 2$, a $\frac{\sqrt{3}+1}{2}$-WMMS allocation exists.*

Proof. Suppose the agents are $\mathcal{N} = \{a_1, a_2\}$, and without loss of generality, assume agent a_1 has a larger weight of w_1, and thus $\frac{1}{2} \leq w_1 \leq 1$. We determine different allocation schemes based on the value of w_1:

(1) When $w_1 \geq \sqrt{3} - 1$, we allocate all the items $\mathcal{M}$ to a_1, which ensures an approximation ratio of $\frac{1}{w_1} \leq \frac{\sqrt{3}+1}{2}$.

(2) When $\frac{1}{2} \leq w_1 \leq \frac{3}{5}$, we ask agent a_1 to partition all items $\mathcal{M}$ into two parts, S_1 and S_2, with the goal of ensuring that the values of the two parts are as equal as possible based on her own valuation. If one of the two parts has a value greater than WMMS_1 for agent a_1, agent a_1 should divide the items in $\mathcal{M}$ into the WMMS-defining partitions instead of S_1 and S_2. Therefore, both parts S_1 and S_2 induce a WMMS approximation ratio of at most 1 for agent a_1. Next, we turn to agent a_2, and ask her to choose the part with the smaller value between S_1 and S_2 based on her own valuation, i.e., the part with a value less than or equal to $\frac{1}{2}$. The approximation ratio of the chosen bundle for agent a_2 is less than or equal to $\frac{0.5}{w_2} \leq \frac{5}{4} < \frac{\sqrt{3}+1}{2}$. Agent a_1 takes the remaining part, resulting an approximation ratio less than or equal to 1. Thus it suffices to set $A_2 = \arg\min_{S \in \{S_1, S_2\}} v_2(S)$, $A_1 = \mathcal{M} \setminus A_2$.

(3) When $\frac{3}{5} < \alpha < \sqrt{3} - 1$, we find agent a_2's WMMS partition B_1 and B_2. Afterwards, we let agent a_1 partition all items in B_1 into two parts with values as equal as possible, denoted as $B_{1,1}$ and $B_{1,2}$, based on her own valuation. Since $B_1 \subseteq \mathcal{M}$, according to the discussion in the previous paragraph, the values of $B_{1,1}$ and $B_{1,2}$ are not greater than WMMS_1 for agent a_1. Then, agent a_2 selects the part with smaller value according to her own valuation between $B_{1,1}$ and $B_{1,2}$, assuming it is $B_{1,1}$. Therefore,

$$v_2(B_{1,1}) \leq \frac{v_2(B_1)}{2} \leq \frac{\mathsf{WMMS}_2 w_1}{2w_2} \leq \frac{\sqrt{3}+1}{2}\mathsf{WMMS}_2.$$

Now we have that for agent a_2, the WMMS approximation ratios for $B_{1,1}$ and B_2 are both less than $\frac{\sqrt{3}+1}{2}$. As for which one is chosen for agent a_2 in the end, it is determined by agent a_1. Agent a_1 selects the larger one between $B_{1,1}$ and B_2 based on her own valuation and we give this bundle to a_2, such that $A_2 = \arg\max_{S \in \{B_{1,1}, B_2\}} v_1(S)$. Agent a_1 keeps the remaining items in $\mathcal{M}$ as her allocation, i.e $A_1 = \mathcal{M} \setminus A_2$. Let $v_1(B_{1,2}) = \beta$. Then we have

$$\begin{aligned} v_1(A_1) &< v_1(B_{1,2}) + \frac{v_i(B_{1,1}) + v_i(B_2)}{2} \\ &= v_1(B_{1,2}) + \frac{1 - v_1(B_{1,2})}{2} = \beta + \frac{1-\beta}{2}. \end{aligned}$$

Given that $\beta \leq \mathsf{WMMS}_1$ and $\mathsf{WMMS}_1 \geq w_1 \geq \frac{3}{5}$, we have the approximation ratio

$$\frac{v_1(A_1)}{\mathsf{WMMS}_i} \leq \frac{\beta + 1}{2\mathsf{WMMS}_1} \leq \frac{w_1 + 1}{2w_1} \leq \frac{4}{3}.$$

Combining the above three cases, we always have a $\frac{\sqrt{3}+1}{2}$-WMMS allocation for any instance with two agents. □

Algorithm 3 The online greedy algorithm

Input: Online instance $\mathcal{I} = (\mathcal{M}, \mathcal{N}, \mathbf{w}, \mathbf{v})$.
Output: An allocation $\mathbf{A} = (A_1, \ldots, A_n)$.

Initialize that $A_i = \emptyset$ for $a_i \in \mathcal{N}$.
Find the set of valid agents

$$\mathcal{N}' = \left\{ a_i \in \mathcal{N} \mid w_i \geq \frac{1}{\alpha} \cdot w_1 \right\}.$$

for each arrived item $o \in \mathcal{M}$ **do**
  $a_i \leftarrow \arg\min_{a_j \in \mathcal{N}'} v_j(o)$.
  $A_i \leftarrow A_i \cup \{o\}$.
  if $v_i(A_i) > \alpha \cdot w_i$ and $|\mathcal{N}'| \neq 1$ **then**
    $\mathcal{N}' \leftarrow \mathcal{N}' \setminus \{a_i\}$.
  end if
end for
return $\mathbf{A} = (A_1, \ldots, A_n)$.

5 The Online Algorithm

In this section, we consider the online version of the problem, and design a simple $O(\sqrt{n})$-competitive algorithm. Similar as [Zhou *et al.*, 2023], we also assume the valuations are normalized, and $w_1 \geq \cdots \geq w_n$. Our algorithm relies on the idea that we first find some agents whose weights are relatively larger than the others, and then only allocate the items to these them. Fix $\alpha = \sqrt{n}+1$, and let $\mathcal{N}' = \{a_1, \ldots, a_k\}$ be the set of agents whose weights are no less than $w_i \geq \frac{1}{\alpha} \cdot w_1$. We call the agents in $\mathcal{N}'$ *valid*. Note that $\mathcal{N}'$ is not empty as agent a_1 always belongs to $\mathcal{N}'$. Our algorithm runs as follows. For each arrived item, we allocate it to one valid agent in $\mathcal{N}'$ who has the smallest value, and when an agent's value becomes larger than α times of her weight, we remove this agent from $\mathcal{N}'$. We can prove that our algorithm can allocate all items, and the formal description is in Algorithm 3. Due to space limit, we omit the proof of Theorem 3.

Theorem 3. *Algorithm 3 returns an $O(\sqrt{n})$-WMMS allocation in polynomial time.*

6 Conclusion

In this paper, we study the weighted maximin share fair allocation of indivisible chores when the agents have different weights on completing them. We improve the previously best known results for both upper and lower bounds. The direct open and intriguing problem is to design the tight approximation algorithm for arbitrary number of agents. Our $O(\log n)$-approximation algorithm does not fully utilize the agents' cardinal valuations, and we suspect a more involved design could further improve the approximation ratio.

Acknowledgements

This work is funded by Hong Kong RGC General Research Fund (No. PolyU 15224823) and Guangdong Basic and Applied Basic Research Foundation (No. 2023A1515010592).

References

[Akrami and Garg, 2024] Hannaneh Akrami and Jugal Garg. Breaking the 3/4 barrier for approximate maximin share. In *SODA*, pages 74–91. SIAM, 2024.

[Amanatidis *et al.*, 2023] Georgios Amanatidis, Haris Aziz, Georgios Birmpas, Aris Filos-Ratsikas, Bo Li, Hervé Moulin, Alexandros A. Voudouris, and Xiaowei Wu. Fair division of indivisible goods: Recent progress and open questions. *Artif. Intell.*, 322:103965, 2023.

[Aziz *et al.*, 2017] Haris Aziz, Gerhard Rauchecker, Guido Schryen, and Toby Walsh. Algorithms for max-min share fair allocation of indivisible chores. In *AAAI*, pages 335–341. AAAI Press, 2017.

[Aziz *et al.*, 2019] Haris Aziz, Hau Chan, and Bo Li. Weighted maxmin fair share allocation of indivisible chores. In *IJCAI*, pages 46–52. ijcai.org, 2019.

[Aziz *et al.*, 2020] Haris Aziz, Hervé Moulin, and Fedor Sandomirskiy. A polynomial-time algorithm for computing a pareto optimal and almost proportional allocation. *Oper. Res. Lett.*, 48(5):573–578, 2020.

[Babaioff *et al.*, 2021] Moshe Babaioff, Tomer Ezra, and Uriel Feige. Fair-share allocations for agents with arbitrary entitlements. In *EC*, page 127. ACM, 2021.

[Barman and Krishnamurthy, 2020] Siddharth Barman and Sanath Kumar Krishnamurthy. Approximation algorithms for maximin fair division. *ACM Trans. Economics and Comput.*, 8(1):5:1–5:28, 2020.

[Budish, 2010] Eric Budish. The combinatorial assignment problem: approximate competitive equilibrium from equal incomes. In *BQGT*, page 74:1. ACM, 2010.

[Caragiannis *et al.*, 2019] Ioannis Caragiannis, David Kurokawa, Hervé Moulin, Ariel D. Procaccia, Nisarg Shah, and Junxing Wang. The unreasonable fairness of maximum nash welfare. *ACM Trans. Economics and Comput.*, 7(3):12:1–12:32, 2019.

[Conitzer *et al.*, 2017] Vincent Conitzer, Rupert Freeman, and Nisarg Shah. Fair public decision making. In *EC*, pages 629–646. ACM, 2017.

[Farhadi *et al.*, 2019] Alireza Farhadi, Mohammad Ghodsi, Mohammad Taghi Hajiaghayi, Sébastien Lahaie, David M. Pennock, Masoud Seddighin, Saeed Seddighin, and Hadi Yami. Fair allocation of indivisible goods to asymmetric agents. *J. Artif. Intell. Res.*, 64:1–20, 2019.

[Feige and Huang, 2023] Uriel Feige and Xin Huang. On picking sequences for chores. In *EC*, pages 626–655. ACM, 2023.

[Feige *et al.*, 2021] Uriel Feige, Ariel Sapir, and Laliv Tauber. A tight negative example for MMS fair allocations. In *WINE*, volume 13112 of *Lecture Notes in Computer Science*, pages 355–372. Springer, 2021.

[Foley, 1967] Duncan Karl Foley. Resource allocation and the public sector. *Yale Economics Essays*, 7(1):45–98, 1967.

[Garg and Taki, 2020] Jugal Garg and Setareh Taki. An improved approximation algorithm for maximin shares. In *EC*, pages 379–380. ACM, 2020.

[Ghodsi *et al.*, 2018] Mohammad Ghodsi, Mohammad Taghi Hajiaghayi, Masoud Seddighin, Saeed Seddighin, and Hadi Yami. Fair allocation of indivisible goods: Improvements and generalizations. In *EC*, pages 539–556. ACM, 2018.

[Gourvès *et al.*, 2014] Laurent Gourvès, Jérôme Monnot, and Lydia Tlilane. Near fairness in matroids. In *ECAI*, volume 263 of *Frontiers in Artificial Intelligence and Applications*, pages 393–398. IOS Press, 2014.

[Hajiaghayi *et al.*, 2023] MohammadTaghi Hajiaghayi, Max Springer, and Hadi Yami. Almost envy-free allocations of indivisible goods or chores with entitlements. *CoRR*, abs/2305.16081, 2023.

[Huang and Lu, 2021] Xin Huang and Pinyan Lu. An algorithmic framework for approximating maximin share allocation of chores. In *EC*, pages 630–631. ACM, 2021.

[Huang and Segal-Halevi, 2023] Xin Huang and Erel Segal-Halevi. A reduction from chores allocation to job scheduling. In *EC*, page 908. ACM, 2023.

[Kurokawa *et al.*, 2018] David Kurokawa, Ariel D. Procaccia, and Junxing Wang. Fair enough: Guaranteeing approximate maximin shares. *J. ACM*, 65(2):8:1–8:27, 2018.

[Li *et al.*, 2022] Bo Li, Yingkai Li, and Xiaowei Wu. Almost (weighted) proportional allocations for indivisible chores. In *WWW*, pages 122–131. ACM, 2022.

[Li *et al.*, 2023] Bo Li, Fangxiao Wang, and Yu Zhou. Fair allocation of indivisible chores: Beyond additive costs. In *NeurIPS*, 2023.

[Lipton *et al.*, 2004] Richard J. Lipton, Evangelos Markakis, Elchanan Mossel, and Amin Saberi. On approximately fair allocations of indivisible goods. In *EC*, pages 125–131. ACM, 2004.

[Robertson and Webb, 1998] Jack M. Robertson and William A. Webb. *Cake-cutting algorithms - be fair if you can*. A K Peters, 1998.

[Steinhaus, 1948] Hugo Steinhaus. The problem of fair division. *Econometrica*, 16(1):101–104, 1948.

[Varian, 1973] Hal R Varian. Equity, envy, and efficiency. 1973.

[Wei *et al.*, 2023] Tianze Wei, Bo Li, and Minming Li. Maximin-aware allocations of indivisible chores with symmetric and asymmetric agents. In *IJCAI*, pages 2897–2905. ijcai.org, 2023.

[Wu *et al.*, 2023] Xiaowei Wu, Cong Zhang, and Shengwei Zhou. Weighted EF1 allocations for indivisible chores. In *EC*, page 1155. ACM, 2023.

[Zhou *et al.*, 2023] Shengwei Zhou, Rufan Bai, and Xiaowei Wu. Multi-agent online scheduling: MMS allocations for indivisible items. In *ICML*, volume 202, pages 42506–42516. PMLR, 2023.