Artificial Intelligence 303 (2022) 103633

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

Fair allocation of indivisible goods: Beyond additive valuations ☆

Mohammad Ghodsi [a,b], MohammadTaghi HajiAghayi [c,1], Masoud Seddighin [b,*], Saeed Seddighin [c], Hadi Yami [c]

[a] Sharif University of Technology, Iran
[b] School of Computer Science, Institute for Research in Fundamental Sciences (IPM), P. O. Box: 19395 - 5746, Tehran, Iran
[c] University of Maryland, College Park, United States of America

ARTICLE INFO

Article history:
Received 6 October 2020
Received in revised form 7 November 2021
Accepted 10 November 2021
Available online 17 November 2021

Keywords:
Fairness
Maximin-share
Approximation
Submodular
XOS
Subadditive

ABSTRACT

We conduct a study on the problem of fair allocation of indivisible goods when *maximin share* [1] is used as the measure of fairness. Most of the current studies on this notion are limited to the case that the valuations are additive. In this paper, we go beyond additive valuations and consider the cases that the valuations are submodular, fractionally subadditive, and subadditive. We give constant approximation guarantees for agents with submodular and XOS valuations, and a logarithmic bound for the case of agents with subadditive valuations. Furthermore, we complement our results by providing close upper bounds for each class of valuation functions. Finally, we present algorithms to find such allocations for submodular and XOS settings in polynomial time.

1. Introduction

Fair division is a fundamental problem that has received significant attention in economics, political science, mathematics, and more recently in computer science [3–5,1,6–10]. In this problem, for a reasonable notion of fairness, the goal is to divide a resource among a set of n agents in a fair manner. Initially, the resource was considered to be a cake (that is, a heterogeneous infinity divisible resource) and the problem was called cake-cutting [4]. To evaluate fairness in a cake-cutting problem, several notions of fairness have been suggested, the most famous of which are *proportionality* [4] and *envy-freeness* [8]. A division is called proportional, if the total value of the allocated pieces to each agent is at least $1/n$ fraction of his total value for the entire cake. In an envy-free division, no agent wishes to exchange his share with another agent, i.e., every agent's valuation for his share is at least as much as his valuation for the other agents' shares.

In the past decade, a new line of research focuses on the case that the resource is a set of indivisible goods. Unfortunately, most of the classic fairness notions are tailored to the cake-cutting problem and none of them can be guaranteed beyond the divisible case. For example, despite many strong positive results for guaranteeing the envy-freeness and proportionality in the cake-cutting problem [11,5,6,12], none of these notions can be either exactly or approximately[2] guaranteed in the

☆ The results in this paper have previously appeared at the nineteenth ACM conference on Economics and Computation (ACM EC'18) [2].
* Corresponding author.
E-mail address: seddighin.masood@gmail.com (M. Seddighin).
1 Supported by the NSF BIGDATA Grant No. 1546108, NSF SPX Grant No. 1822738, and NSF AF Grant No. 2114269.
2 Here by approximation we mean multiplicative approximation.

https://doi.org/10.1016/j.artint.2021.103633
0004-3702/

case of indivisible items. Consider allocating a single indivisible item to n agents. Since this item is indivisible, one agent receives this item and the rest of the agents receive no item. Hence, there is no way to guarantee proportionality or any approximation of it for the agents that receive no item. This led the community to develop more relaxed fairness notions that align with the allocation of indivisible goods. In the past decade, several different criteria are defined for the case of indivisible items, e.g., envy-freeness up to one good (EF1), envy-freeness up to any good (EFX), proportional up to one good (Prop1), and maximin-share (MMS) [13,1,14].

In this paper, we investigate the maximin-share (MMS) notion. This notion is introduced by Budish [1] as a relaxation of proportionality for the case of indivisible goods. Imagine the following cut-and-choose game: we ask an agent a_i to partition the items into n bundles and then ask the rest of the agents to choose a bundle before agent a_i. In the worst-case scenario, the least preferred bundle remains for agent a_i. The maximin share of a_i is the largest value he can guarantee herself in this game in the worst-case scenario.

Formally, for a set $\mathcal{M}$ of goods, agent a_i the maximin-share value of agent a_i, denoted by MMS_i is defined as

$$\mathsf{MMS}_i = \max_{(\pi_1,\pi_2,\ldots,\pi_n)\in\Pi} \min_j v_i(\pi_j),$$

where Π is the set of all partitions of $\mathcal{M}$ into n bundles and $v_i(\pi_j)$ is the valuation of agent a_i for bundle π_j. An allocation is then said to be MMS, if it guarantees each agent a_i a bundle with value at least MMS_i.

Contrary to optimistic views about guaranteeing MMS, a counterexample by Kurokawa, Procaccia, and Wang [9] shows that some instances admit no MMS allocation. On the positive side, it is shown that a 2/3-MMS allocation (allocation that guarantees each agent a_i a bundle with a value at least $2\mathsf{MMS}_i/3$) always exists [9]. Improving the approximation factor for guaranteeing MMS has become an intriguing direction since 2014. The current best known approximation factor for the additive setting is $3/4+o(1)$ by Grag and Taki [15]. From the algorithmic viewpoint, the best polynomial time approximation guarantee for MMS is improved in series of studies to $2/3-\epsilon$ [16], 2/3 [17], $3/4-\epsilon$ [2], and 3/4 [18]. On the negative side, for three agents and nine items, Feige, Sapir, and Tauber [19] design an instance in which at least one agent does not get more than a $\frac{39}{40}$ fraction of her maximin share. For $n \geq 4$ agents, they show examples in which at least one agent does not get more than a $1-\frac{1}{n^4}$ fraction of her maximin share.

Although most early investigations on maximin-share focus on the additive settings, it is very natural to extend the definition to other classes of set functions. For instance, it is quite natural to expect that an agent prefers to receive two items of value 400, rather than receiving 1000 items of value 1. Such a constraint cannot be imposed in the additive setting. However, submodular functions which encompass k-demand valuations are strong tools for modeling these constraints. Such generalizations have been made to many similar problems, including the *Santa Claus max-min fair allocation*, *welfare maximization* and *Nash social welfare maximization* [20–25]. The most common classes of set functions that have been studied before are submodular, XOS, and subadditive functions. In this paper, we consider the fair allocation problem when the agents' valuations are in each of these classes. In contrast to the additive setting in which finding a constant MMS allocation is trivial, the problem becomes much more subtle even when the agents' valuations are *monotone submodular*. Independent of this work, Barman et al. [17] prove the existence of a 1/10-MMS allocation for monotone submodular valuations and provide a polynomial-time 1/30-MMS allocation algorithm for that setting. In this paper, we propose algorithms with the approximation guarantee of 1/3 for submodular, 1/5 for XOS, and logarithmic for subadditive valuations. In Section 2 we review some mathematical background and basic definitions related to our work. Next, In Section 3 we discuss our results and techniques in detail.

2. Preliminaries

Throughout this paper we assume the set of agents is denoted by $\mathcal{N}$ and the set of items is referred to by $\mathcal{M}$. Let $|\mathcal{N}|=n$ and $|\mathcal{M}|=m$, we refer to the agents by a_i and to the items by b_i, i.e., $\mathcal{N}=\{a_1,a_2,\ldots,a_n\}$ and $\mathcal{M}=\{b_1,b_2,\ldots,b_m\}$. We denote the valuation of agent a_i for a set S of items by $v_i(S)$. Our interest is in valuation functions that are monotone and non-negative. More precisely, we assume $v_i(S)\geq 0$ for every agent a_i and set $S\subseteq\mathcal{M}$, and for every two sets S_1 and S_2 we have

$$\forall a_i \in \mathcal{N} \qquad v_i(S_1\cup S_2) \geq \max\{v_i(S_1), v_i(S_2)\}.$$

Due to obvious impossibility results for the general valuation functions,[3] we restrict our attention to three classes of set functions:

- **Submodular**: Given a ground set $\mathcal{G}$, A function $f:2^{\mathcal{G}}\rightarrow\mathbb{R}$ is submodular if for every two sets $S_1,S_2\in\mathcal{G}$,

$$f(S_1)+f(S_2)\geq f(S_1\cup S_2)+f(S_1\cap S_2).$$

[3] If the valuation functions are not restricted, no approximation guarantee can be achieved. For instance consider the case where we have two agents and 4 items. Agent a_1 has value 1 for sets $\{b_1,b_2\}$ and $\{b_3,b_4\}$ and 0 for the rest of the sets. Similarly, agent a_2 has value 1 for sets $\{b_1,b_3\}$ and $\{b_2,b_4\}$ and 0 for the rest of the sets. In this case, no allocation can provide both of the agents with sets which are of non-zero value to them.

- **Fractionally Subadditive** (XOS): Given a ground set $\mathcal{G}$, an XOS function $f : 2^{\mathcal{G}} \to \mathbb{R}$ can be shown via a finite set of additive functions $\{f^1, f^2, \dots, f^\alpha\}$ where $f(S) = \max_{i=1}^{\alpha} f^i(S)$ for any set $S \subseteq \mathcal{G}$.
- **Subadditive**: Given a ground set $\mathcal{G}$, a set function $f : 2^{\mathcal{G}} \to \mathbb{R}$ is subadditive if for every two sets $S_1, S_2 \subseteq \mathcal{G}$,

$$f(S_1) + f(S_2) \geq f(S_1 \cup S_2).$$

For additive functions, it is reasonable to assume that the value of the function for every element is given in the input. However, representing other classes of set functions requires access to oracles. For submodular functions, we assume we have access to *value-query oracle* defined below. For XOS and subadditive settings, we use a stronger oracle which is called *demand-query oracle*. In addition to this, we consider a special oracle for XOS functions which is called XOS *oracle*. Access to query oracles for submodular functions, XOS oracle for XOS functions, and demand oracles for XOS and subadditive functions are quite common and have been very fruitful in the literature [26,23,24,27–30]. In what follows, we formally define the oracles:

- **Value-query oracle:** Given a function f, a value-query oracle $\mathcal{O}^{val}$ is an algorithm that receives a set S as input and computes $f(S)$ in time $O(1)$.
- **Demand-query oracle:** Given a function f, a demand-query oracle $\mathcal{O}^{dem}$ is an algorithm that receives a sequence of prices $p_1, p_2, \dots, p_n$ as input and finds a set S such that

$$f(S) - \sum_{e \in S} p_e$$

  is maximized. We assume the running time of the algorithm is $O(1)$.
- XOS **oracle:** (defined only for an XOS function f) Given a set S of items, it returns the additive representation of the function that is maximized for S. In other words, it reveals the contribution of each item in S to the value of $f(S)$.

Maximin-share As mentioned before, the maximin-share of agent a_i, denoted by MMS_i is defined as follows:

$$\mathsf{MMS}_i = \max_{\pi_1, \pi_2, \dots, \pi_n \in \Pi} \min_{1 \leq j \leq n} v_i(\pi_j),$$

where Π is the set of all partitions of $\mathcal{M}$ into n bundles. Throughout this paper, we suppose without loss of generality that the valuations are scaled so that for every agent a_i we have $\mathsf{MMS}_i = 1$.

An allocation of items to the agents is a collection $\mathcal{A} = \langle A_1, A_2, \dots, A_n \rangle$ where A_i is the bundle allocated to agent a_i. For an allocation A, we have $\bigcup A_i = \mathcal{M}$ and $A_i \cap A_j = \emptyset$ for every two agents $a_i, a_j \in \mathcal{N}$. An allocation $\mathcal{A}$ is α-MMS, if every agent a_i receives a subset of the items whose value to that agent is at least α times MMS_i. Given our assumption that $\mathsf{MMS}_i = 1$ for every agent a_i, we can say an allocation is α-MMS if and only if for every agent $a_i \in \mathcal{N}$,

$$v_i(A_i) \geq \alpha.$$

Lemma 2.1 represents a well-known and very useful structural property of maximin-share notion proved by Amanitidis et al. [16]. Throughout this paper, we frequently use this property.

Lemma 2.1 *([16]). Assuming that the goal is to find an α-MMS allocation, we can suppose without loss of generality that for every item b_j and every agent a_i we have*

$$v_i(\{b_j\}) < \alpha.$$

The reason that Lemma 2.1 holds is that, if an item is worth at least α to an agent, we can allocate it to him and solve the problem for the rest of the items and the rest of the agents. We remark that Lemma 2.1 holds for all the valuation classes, including submodular, XOS, and subadditive valuations.

3. Our results and techniques

We begin with submodular set functions. First, we show that in some instances with submodular valuations, no allocation is better than 3/4-MMS.

Theorem 3.1. *For any $n \geq 2$, there exists an instance of the fair allocation problem with n agents with submodular valuations where no allocation is better than* 3/4-MMS.

We show Theorem 3.1 by a counterexample. In this counterexample, we have n agents and $2n$ items. Moreover, the valuation functions of the first $n-1$ agents are the same, but the last agent has a slightly different valuation function that makes it impossible to find an allocation that is better than 3/4-MMS. The number of agents in this example can be arbitrarily large.

Next, in Section 5, we show that when the valuations of the agents are submodular, guaranteeing a 1/3-MMS is always possible. In addition, we show, given access to *value-query* and *demand-query* oracles, one can find such an allocation in polynomial time. We further complement our result by showing that a 3/4-MMS is the best guarantee that one can hope to achieve in this setting. This is in contrast to the additive setting for which the only upper bound is that $\frac{39}{40}$-MMS allocation is not always possible [19]. We begin by stating an existential proof.

Theorem 3.2. *Every fair allocation problem in which the agents have submodular valuations admits a* 1/3-MMS *allocation.*

To prove Theorem 3.2, first, by Lemma 2.1 we assume without loss of generality that $\mathsf{MMS}_i = 1$ for every agent $a_i \in \mathcal{N}$. Next, for a given a function $f(\cdot)$, we define the *ceiling function* $f^{\{x\}}(\cdot)$ as follows:

$$f^{\{x\}}(S) = \min\{x, f(S)\} \qquad \forall S \subseteq \mathsf{ground}(f).$$

An important property of the ceiling functions is that they preserve submodularity, fractionally subadditivity, and subadditivity (see Lemma 5.2). Accordingly, we define the bounded welfare of an allocation $\mathcal{A}$ as

$$\sum v_i^{\{2/3\}}(A_i).$$

Given that, we show an allocation that maximizes the bounded welfare is 1/3-MMS. To this end, let $\mathcal{A}$ be an allocation with the maximum bounded welfare and suppose for the sake of contradiction that in such an allocation, an agent a_i receives a bundle that is worth less than 1/3 to him. Since $\mathsf{MMS}_i = 1$, agent a_i can divide the items into n sets, where each set is worth at least 1 to him. For a valuation function V, define the contribution of an item b_j in set S ($b_j \in S$) as $v(S) - v(S \setminus \{b_j\})$. Now, we randomly select an element b_j which is *not* allocated to a_i. By the properties of submodular functions, we show that if we allocate b_j to a_i, the expected contribution of b_j to the bounded valuation function of a_i would be more than the currently expected contribution of b_j to the bounded welfare of the allocation. Therefore, there exists an item b_j such that if we allocate that item to agent a_i, the total bounded welfare of the allocation will be increased. This contradicts the optimality of the allocation.

Notice that Theorem 3.2 is only an existential proof. A natural approach to finding such a solution is to start with an arbitrary allocation and iteratively increase its bounded welfare until it becomes 1/3-MMS. The main challenge though is that we do not even know what the MMS values are. Furthermore, unlike the additive setting, we do not have any PTAS algorithm that provides us a close estimate of these values. To overcome this challenge, we propose a combinatorial trick to guess these values without incurring any additional factor to our guarantee. The high-level idea is to start with large numbers as estimates to the MMS values. Every time we run the algorithm on the estimated values, it either finds the desired allocation or reports that the maximin-share value of an agent is misrepresented by at least a multiplicative factor. Given this, we divide the maximin-share value of that agent by that factor and continue with the new estimates. Therefore, at every step of the algorithm, we are guaranteed that our estimates are not less than the actual MMS values. Based on this, we show that the running time of the algorithm is polynomial and that the resulting allocation has the desired properties. The reader can find a detailed discussion in Section 5.

Theorem 3.3. *Given access to value-query oracles, one can find a* 1/3-MMS *allocation for agents with submodular valuations in polynomial time.*

We also study the problem with fractionally subadditive (XOS) agents. Similar to the submodular setting, we provide an upper bound on the quality of any allocation in the XOS setting. We show Theorem 3.4 by a counterexample.

Theorem 3.4. *For any $n > 1$, there exists an instance of the fair allocation problem with n agents with* XOS *valuations where no allocation is better than* 1/2-MMS.

Next, we state the main theorem of this section.

Theorem 3.5. *The fair allocation problem with* XOS *valuations admits a* 1/5-MMS *allocation.*

Our approach for proving Theorem 3.5 is similar to the proof of Theorem 3.2. Again, we scale the valuations to make sure $\mathsf{MMS}_i = 1$ for all the agents and define the notion of bounded welfare, but this time as $\sum v_i^{\{2/5\}}(A_i)$. However, as XOS functions do not adhere to the nice structure of submodular functions, we use a different analysis to prove this theorem. Let $\mathcal{A}$ be an allocation with the maximum bounded welfare. In case all agents receive a value of at least 1/5, the proof is complete. Otherwise, let a_i be an agent that receives a set of items whose value to him is less than 1/5. In contrast to the submodular setting, giving no item alone to a_i can guarantee an increase in the bounded welfare of the allocation. However, this time, we show there exists a set S of items such that if we take them back from their recipients and instead allocate them to agent a_i, the bounded welfare of the allocation increases. The reason this holds is the following: since $\mathsf{MMS}_i = 1$, agent a_i can split the items into $2n$ sets where every set is worth at least $2/5$ to a_i (see Lemma 6.2). Moreover, since the valuation functions are XOS, we show that giving one of these $2n$ sets to a_i will increase the bounded welfare of the allocation. Therefore, if $\mathcal{A}$ is maximal, then $\mathcal{A}$ is also $1/5$-MMS.

Finally, we show that a $1/8$-MMS allocation in the XOS setting can be found in polynomial time. Our algorithm only requires access to demand-query and XOS oracles. Note that this bound is slightly worse than our existential proof due to computational hardness. However, the blueprint of the algorithm is based on the proof of Theorem 3.5.

Theorem 3.6. *Given access to demand-query and* XOS *oracles, there exists a polynomial time algorithm that finds a* $1/8$-MMS *allocation for agents with* XOS *valuations.*

We start with an arbitrary allocation and increase the bounded welfare until the allocation becomes $1/8$-MMS. The catch is that if the allocation is not $1/8$-MMS, then there exists an agent a_i and a set S of items such that if we take back these items from their current recipients and allocate them to agent a_i, the bounded welfare of the allocation increases. To increase the bounded welfare, two computational barriers need to be lifted. First, similar to the submodular setting, we do not have any estimates for the MMS values. Analogously, we resolve the first issue by iteratively guessing the MMS values. The second issue is that in every step of the algorithm, we have to find a set S of items to allocate to an agent a_i that increases the bounded welfare. Such a set S cannot be trivially found in polynomial time. That is where the demand-query and XOS oracles take part. In Section 6.2.1 we show how to find such a set in polynomial time. The high-level idea is the following: first, by accessing the XOS oracles, we determine the contribution of every item to the bounded welfare of the allocation. Next, we set the price of every element equal to three times the contribution of that element to the bounded welfare and run the demand-query oracle to find which subset has the highest profit for agent a_i. We show this subset has a value of at least $1/4$ to a_i. Next, we sort the elements of this set based on the ratio of contribution to the overall value of the set over the price of the item and select a prefix for them that has a value of at least $1/4$ to a_i. Finally, we argue that allocating this set to a_i increases the bounded welfare of the allocation by at least some known lower bound. This, combined with the combinatorial trick to guess the MMS values, gives us a polynomial time algorithm to find a $1/8$-MMS allocation.

Note that, an immediate corollary of Theorems 3.3 and 3.6 is a polynomial time algorithm for approximating the maximin-share value of an agent with a submodular or XOS valuation function within factors $1/3$ and $1/8$, respectively.

Finally, we investigate the problem when the agents are subadditive and present an existential proof based on a well-known reduction to the XOS setting.

Theorem 3.7. *Any fair allocation problem in which the agents have subadditive valuations admits a* $1/10\lceil \log m \rceil$-MMS *allocation.*

4. Upper-bounds

We begin by presenting our impossibility results for submodular and XOS valuations. First, we give an example that shows that the best guarantee that we can achieve for submodular valuations is upper bounded by $3/4$. Next, we modify this example to prove that the best guarantee that we can achieve for XOS valuations is upper bounded by $1/2$. Our counterexample is generic; we show this result for any number of agents.

Theorem 3.1. *For any $n \geq 2$, there exists an instance of the fair allocation problem with n agents with submodular valuations where no allocation is better than* $3/4$-MMS*.*

Proof. We construct an instance of the problem that does not admit any $3/4+\epsilon$-MMS allocation. To this end, let n be the number of agents and $\mathcal{M} = \{b_1, b_2, \dots, b_m\}$ where $m = 2n$. Furthermore, let $f : 2^{\mathcal{M}} \to \mathbb{R}$ be as follows:

$$f(S) = \begin{cases} 0, & \text{if } |S| = \emptyset \\ 1, & \text{if } |S| = 1 \\ 2, & \text{if } |S| > 2 \\ 2, & \text{if } S = \{b_{2i}, b_{2i-1}\} \text{ for some } 1 \leq i \leq n \\ 3/2, & \text{if } |S| = 2 \text{ and } S \neq \{b_{2i}, b_{2i-1}\} \text{ for any } 1 \leq i \leq n. \end{cases}$$

In what follows we show that f is submodular. To this end, suppose for the sake of contradiction that there exist sets S and S' such that $S \subseteq S'$ and for some element b_i we have:

$$f(S' \cup \{b_i\}) - f(S') > f(S \cup \{b_i\}) - f(S). \tag{1}$$

Since f is monotone and $S' \neq S$, $f(S' \cup \{b_i\}) - f(S') > 0$ holds and thus S' cannot have more than two items. Therefore, S' contains at most two items and thus S is either empty or contains a single element. If S is empty, then adding every element to S has the highest increase in the value of S and thus Inequality (1) doesn't hold. Therefore, S contains a single element and S' contains exactly two elements. Thus, $f(S) = 1$ and $f(S') \geq 3/2$. Therefore, $f(S \cup \{b_i\}) - f(S) \geq 1/2$ and $f(S' \cup \{b_i\}) - f(S') \leq 1/2$ which contradicts Inequality (1).

Now, for agents $a_1, a_2, \ldots, a_{n-1}$ we set $v_i = f$ and for agent a_n we set $v_n = f(\mathsf{inc}(S))$ where b_i is in $\mathsf{inc}(S)$ if and only if either $i > 1$ and $b_{i-1} \in S$ or $i = 1$ and $b_m \in S$. Clearly, for every agent a_i we have $\mathsf{MMS}_i = 2$.

The crux of the argument is that for any allocation of the items to the agents, someone receives a value of at most 3/2. In case an agent receives fewer than two items, his valuation for his items would be at most 1. Similarly, if an agent receives more than two items, someone has to receive fewer than 2 items and the proof is complete. Therefore, the only case to investigate is where everybody receives exactly two items. We show in such cases, $\min v_i(A_i) = 3/2$ for all possible allocations. If all agents $a_1, a_2, \ldots, a_{n-1}$ receive two items whose value for them is exactly equal to 2, then by the construction of f, the value of the remaining items is also equal to 2 to them. Thus, a_n's valuation for the items he receives is equal to 3/2. □

Remark that one could replace function f with an XOS function

$$g(S) = \begin{cases} 0, & \text{if } |S| = \emptyset \\ 1, & \text{if } |S| = 1 \\ 2, & \text{if } |S| > 2 \\ 2, & \text{if } S = \{b_{2i}, b_{2i-1}\} \text{ for some } 1 \leq i \leq n \\ 1, & \text{if } |S| = 2 \text{ and } S \neq \{b_{2i}, b_{2i-1}\} \text{ for any } 1 \leq i \leq n, \end{cases}$$

and make the same argument to achieve a 1/2-MMS upper bound for XOS and subadditive valuations.

Theorem 3.4. *For any $n > 1$, there exists an instance of the fair allocation problem with n agents with* XOS *valuations where no allocation is better than* 1/2-MMS.

5. Submodular valuations

Most of the studies on maximin-share are limited to the agents with additive valuations [15,16,9]. In the real world, however, valuation functions are usually more complex than additive ones. As an example, imagine an agent is interested in at most k items. More precisely, he is indifferent between receiving k items or more than k items. Such a valuation function is called k-demand and cannot be modeled by additive functions. k-demand functions are a subclass of submodular set functions which have been extensively studied in the literature of different contexts, e.g., optimization, mechanism design, and game theory [31–37,30]. To the best of our knowledge, the only result for the maximin-share notion beyond additive valuations is the work of Barman and Krishna Murthy [17] wherein the authors prove the existence of 1/10-MMS allocation for agents with submodular valuations.[4]

In this section we provide an existential proof to a 1/3-MMS allocation when the valuations are submodular. Due to the algorithmic nature of the proof, we show in Section 5.1 that such an allocation can be computed in time $\mathsf{poly}(n, m)$. We emphasize that we scale the valuation functions to ensure $\mathsf{MMS}_i = 1$ for every agent a_i.

We begin by introducing the ceiling functions.

Definition 5.1. *Given a set function $f : \mathsf{ground}(f) \to \mathbb{R}^+$, we define $f^{\{x\}}(\cdot)$ as follows:*

$$f^{\{x\}}(S) = \begin{cases} f(S), & \text{if } f(S) \leq x \\ x, & \text{if } f(S) > x. \end{cases}$$

Observation 5.1. *$f^{\{x\}}(S) \leq x$ for every given S.*

Observation 5.2. *$f^{\{x\}}(S) \leq f(S)$ for every given S.*

An important property of the ceiling functions is that they preserve submodularity, fractionally subadditivity, and subadditivity as we show in Lemma 5.2.

Lemma 5.2. *For any real number $x \geq 0$, we have:*

1. *Given a submodular set function $f(\cdot)$, $f^{\{x\}}(\cdot)$ is submodular.*

[4] In the journal version of this paper, this factor is improved to 0.21.

2. *Given an* XOS *set function* $f(\cdot)$, $f^{\{x\}}(\cdot)$ *is* XOS.
3. *Given an subadditive set function* $f(\cdot)$, $f^{\{x\}}(\cdot)$ *is also subadditive.*

Proof. First Claim: By definition of submodular functions, for given sets A and B we have:

$$f(A \cup B) \leq f(A) + f(B) - f(A \cap B).$$

We prove that $f^{\{x\}}(\cdot)$ is a submodular function in three different cases:

First Case: let both $f(A)$ and $f(B)$ be at least x. According to Observation 5.1, $f^{\{x\}}(A \cup B)$ and $f^{\{x\}}(A \cap B)$ are bounded by x. Therefore, $f^{\{x\}}(A \cup B) + f^{\{x\}}(A \cap B) \leq 2x$, which yields:

$$f^{\{x\}}(A \cup B) + f^{\{x\}}(A \cap B) \leq f^{\{x\}}(A) + f^{\{x\}}(B).$$

Second Case: in this case one of $f(A)$ and $f(B)$ is at least x. We have $f(A \cup B) \geq x$ and $f(A \cap B)$ is no more than max $\{f(A), f(B)\}$. As a result $f^{\{x\}}(A \cup B)$ and one of $f^{\{x\}}(A)$ or $f^{\{x\}}(B)$ are equal to x which yields:

$$f^{\{x\}}(A \cup B) + f^{\{x\}}(A \cap B) \leq f^{\{x\}}(A) + f^{\{x\}}(B)$$

Third Case: in this case both $f(A)$ and $f(B)$ are less than x, and $f(A \cap B)$ is less than x too. Since $f^{\{x\}}(A) = f(A)$, $f^{\{x\}}(B) = f(B)$, $f^{\{x\}}(A \cap B) = f(A \cap B)$, according to Observation 5.2, $f^{\{x\}}(A \cup B) \leq f(A \cup B)$ holds. Since $f(\cdot)$ is a submodular function, we conclude that:

$$f^{\{x\}}(A \cup B) \leq f^{\{x\}}(A) + f^{\{x\}}(B) - f^{\{x\}}(A \cap B).$$

Second Claim: Since $f(\cdot)$ is an XOS set function, by definition, there exists a finite set of additive functions $\{f_1, f_2, \ldots, f_\alpha\}$ such that

$$f(S) = \max_{i=1}^{\alpha} f_i(S)$$

for any set $S \subseteq \text{ground}(f)$. With that in hand, for a given real number x, we define an XOS set function $g(\cdot)$, and show $g(\cdot)$ is equal to $f^{\{x\}}(\cdot)$.

We define $g(\cdot)$ on the same domain as $f(\cdot)$. Moreover, based on $\{f_1, f_2, \ldots, f_\alpha\}$, we define a finite set of additive functions $\{g_1, g_2, \ldots, g_\beta\}$ that describe g. More precisely, for each set S in domain of $f(\cdot)$ we define a new additive function like g_γ in $g(\cdot)$ as follows: Without loss of generality let f_δ be the function which maximizes $f(S)$. For each $b_i \notin S$ let $g_\gamma(b_i) = 0$. Furthermore, for each $b_i \in S$ if $f(S) \leq x$ let $g_\gamma(b_i) = f_\delta(b_i)$, and otherwise let $g_\gamma(b_i) = \frac{x}{f(S)} f_\delta(b_i)$.

We claim that $g(\cdot)$ is equivalent to $f^{\{x\}}(\cdot)$, which implies $f^{\{x\}}(\cdot)$ is an XOS function. $g(\cdot)$ and $f^{\{x\}}(\cdot)$ are two functions which have equal domains. First, we prove that $g(S) \leq f(S)$ for any given set S. According to construction of $g(\cdot)$, for each additive function in $g(\cdot)$ such g_γ, there is at least one additive function in $f(\cdot)$ such f_δ where $g_\gamma(b_i) \leq f_\delta(b_i)$ for each $b_i \in \mathcal{M}$. Therefore, for any given set S we have:

$$g(S) \leq f(S) \tag{2}$$

Now, according to the construction of $g(\cdot)$, for any given set S where $f(S) \leq x$, we have a function $g_\gamma(S) = f(S)$, and where $f(S) > x$, we have a function $g_\gamma(S) = x$. Therefore, we can conclude that:

$$g(S) \geq f^{\{x\}}(S) \tag{3}$$

For any given set S where $f(S) \leq x$, according to the definition of $f^{\{x\}}(\cdot)$, $f(S) = f^{\{x\}}(S)$, and using Inequalities (2) and (3) we argue that $f^{\{x\}}(S) = g(S)$. Moreover, according to the construction of $g(\cdot)$, $g(S) \leq x$ for any given set S. Therefore, for any given set S where $f(S) > x$, according to the definition of $f^{\{x\}}(\cdot)$ and Inequality (3), $f^{\{x\}}(S) = g(S) = x$. As a result, by considering these two cases we argue that $f^{\{x\}}(\cdot)$ and $g(\cdot)$ are equivalent, which shows $f^{\{x\}}(\cdot)$ is an XOS function.

Third Claim: In this claim, we use a similar argument to the first claim. By definition of subadditive functions for any given sets A and B, we have:

$$f(A \cup B) \leq f(A) + f(B).$$

We prove that $f^{\{x\}}(\cdot)$ meets the definition of subadditive functions by considering two different cases. In the first case at least one of $f(A)$ and $f(B)$ is at least x, and in the second case both $f(A)$ and $f(B)$ is less than x.

First Case: In this case $f^{\{x\}}(A) + f^{\{x\}}(B)$ is at least x. Since $f^{\{x\}}(S) \leq x$ for any given set S, $f^{\{x\}}(A \cup B) \leq x$. Therefore,

$$f^{\{x\}}(A \cup B) \leq f^{\{x\}}(A) + f^{\{x\}}(B).$$

Second Case: Since $f^{\{x\}}(A \cup B) \le f(A \cup B)$, $f(A \cup B) \le f(A) + f(B)$, $f(A) = f^{\{x\}}(A)$, and $f(B) = f^{\{x\}}(B)$, we have:

$$f^{\{x\}}(A \cup B) \le f^{\{x\}}(A) + f^{\{x\}}(B). \quad \square$$

The idea behind the existence of a 1/3-MMS allocation is simple: let $\mathcal{A} = \langle A_1, A_2, \dots, A_n \rangle$ be an allocation of items to the agents that maximizes the following expression:

$$\sum_{a_i \in \mathcal{N}} v_i^{\{2/3\}}(A_i). \tag{4}$$

We refer to Expression (4) by $\mathsf{ex}^{\{2/3\}}(\mathcal{A})$. We prove $v_i(A_i) \ge 1/3$ for every agent $a_i \in \mathcal{N}$. The main ingredients of the proof are Lemmas 2.1, 5.3 and 5.4.

Lemma 5.3. *Let $S_1, S_2, \dots, S_k$ be k disjoint sets and $f_1, f_2, \dots, f_k$ be k submodular functions. We remove an element e from $\bigcup S_i$ uniformly at random to obtain sets $S_1^* = S_1 \setminus \{e\}, S_2^* = S_2 \setminus \{e\}, \dots, S_k^* = S_k \setminus \{e\}$. In this case we have*

$$\mathbb{E}[\sum f_i(S_i^*)] \ge \sum f_i(S_i) \frac{|\bigcup S_i| - 1}{|\bigcup S_i|}.$$

Proof. Since $f(\cdot)$ is submodular, according to the definition of submodular functions, for every given sets X and Y in domain of $f(\cdot)$ with $X \subseteq Y$ and every $x \in \mathcal{M} \setminus Y$ we have:

$$f(X \cup \{x\}) - f(X) \ge f(Y \cup \{x\}) - f(Y) \tag{5}$$

Let $S_i = \{e_1, e_2, \dots, e_\alpha\}$, $T_0 = \emptyset$, and $T_j = \{e_1, e_2, \dots, e_j\}$, for every $1 \le j \le \alpha$. Since $T_j \subseteq S_i$ for each $0 \le j \le \alpha$ and f_i is a submodular function, according to Inequality (5) we have:

$$\sum_{1 \le j \le \alpha} f_i(S_i \setminus T_{j-1}) - f_i(S_i \setminus T_j) \ge \sum_{1 \le j \le \alpha} f_i(S_i) - f_i(S_i - e_j). \tag{6}$$

Since $f_i(S_i) = \sum_{1 \le j \le \alpha} f_i(S_i \setminus T_{j-1}) - f_i(S_i \setminus T_j)$, we can rewrite Inequality (6) for every $1 \le i \le k$ as follows:

$$f_i(S_i) \ge \sum_{e \in S_i} f_i(S_i) - f_i(S_i - e). \tag{7}$$

For every $1 \le i \le k$ we can rewrite Inequality (7) as follows:

$$\sum_{e \in s_i} f_i(S_i - e) \ge (|S_i| - 1) f_i(S_i) \tag{8}$$

By adding $(|\bigcup S_i| - |S_i|) f_i(S_i)$ to the both sides of Inequality (8), we have:

$$\begin{aligned}(|\bigcup S_i| - |S_i|) f_i(S_i) + \sum_{e \in S_i} f_i(S_i - e) &= \sum_{e \in \bigcup S_i} f_i(S_i \setminus \{e\}) \\ &\ge (|\bigcup S_i| - 1) f_i(S_i)\end{aligned} \tag{9}$$

Since Inequality (9) holds for every $1 \le i \le k$, we can sum up both sides of Inequality (9) as follows:

$$\sum_{1 \le i \le k} \sum_{e \in \bigcup S_i} f_i(S_i - e) \ge \sum_{1 \le i \le k} (|\bigcup S_i| - 1) f_i(S_i) \tag{10}$$

By dividing both sides of Inequality (10) over $1/|\bigcup S_i|$ we obtain:

$$\begin{aligned}\frac{1}{|\bigcup S_i|}(\sum_{e \in \bigcup S_i} \sum_{1 \le i \le k} f_i(S_i - e)) &= \mathbb{E}[\sum_{1 \le i \le k} f_i(S_i^*)] \\ &\ge \sum_{1 \le i \le k} f_i(S_i) \frac{|\bigcup S_i| - 1}{|\bigcup S_i|}. \quad \square\end{aligned} \tag{11}$$

Lemma 5.4. *Let f be a submodular function and $S_1, S_2, \dots, S_k$ be k disjoint sets such that $f(S_i) \ge 1$ for every set S_i. Moreover, let $S \subseteq \bigcup S_i$ be a set such that $f(S) < 1/3$. If we pick an element $\{e\}$ of $\bigcup S_i \setminus S$ uniformly at random, we have:*

$$\mathbb{E}[f(S \cup \{e\}) - f(S)] \ge \frac{2k/3}{|\bigcup S_i \setminus S|}.$$

Proof. Similar to the proof of Lemma 5.3, we use Inequality (5) as a definition of submodular functions. Let $S'_i = S_i \setminus S = \{e_1, e_2, \dots, e_\alpha\}$, $T_0 = S$, and $T_j = S \cup \{e_1, e_2, \dots, e_j\}$ for $1 \le j \le \alpha$. According to $f(S) < 1/3$, $f(S \cup S'_i) \ge 1$, and Inequality (5) as a definition of submodular functions, we have:

$$\begin{aligned} 2/3 &< f(S \cup S') - f(S) \\ &= \sum_{1 \le j \le \alpha} f(T_{j-1} \cup \{e_j\}) - f(T_{j-1}) \\ &\le \sum_{e \in S'_i} f(S \cup \{e\}) - f(S). \end{aligned} \tag{12}$$

Similar to Inequality (10), we can rewrite Inequality (12) with a summation, since Inequality (12) holds for any $1 \le i \le k$.

$$2k/3 < \sum_{1 \le i \le k} \sum_{e \in S'_i} f(S \cup \{e\}) - f(S) \tag{13}$$

By dividing both sides of Inequality (13) over $1/|\bigcup S_i \setminus S|$ we have:

$$\begin{aligned} \frac{2k/3}{|\bigcup S_i \setminus S|} &< \frac{1}{|\bigcup S_i \setminus S|}\Big(\sum_{1 \le i \le k} \sum_{e \in S'_i} f(S \cup \{e\}) - f(S)\Big) \\ &= \mathbb{E}[f(S \cup \{e\}) - f(S)] \quad \square \end{aligned} \tag{14}$$

Next, we show the fair allocation problem with submodular valuations admits a 1/3-MMS allocation.

Theorem 3.2. *Every fair allocation problem in which the agents have submodular valuations admits a* 1/3-MMS *allocation.*

Proof. Since our goal is to prove 1/3 approximation guarantee, by Lemma 2.1 we can assume that the value of each item for each agent is less than 1/3. Let $\mathcal{A}$ be an allocation that maximizes $\mathsf{ex}^{\{2/3\}}$. Suppose for the sake of contradiction that $v_i(A_i) < 1/3$ for some agent a_i. In this case we select an item b_r from $\mathcal{M} \setminus A_i$ uniformly at random to create a new allocation $\mathcal{A}^r$ as follows:

$$A^r_j = \begin{cases} A_j \setminus \{b_r\}, & \text{if } i \ne j \\ A_j \cup \{b_r\} & \text{if } i = j. \end{cases}$$

In the rest we show $\mathbb{E}[\mathsf{ex}^{\{2/3\}}(\mathcal{A}^r)] > \mathsf{ex}^{\{2/3\}}(\mathcal{A})$ which contradicts the maximality of $\mathcal{A}$. Note that by Lemma 5.3 the following inequality holds:

$$\mathbb{E}[\sum_{j \ne i} v_j^{\{2/3\}}(A^r_j)] \ge \sum_{j \ne i} v_j^{\{2/3\}}(A_j)\frac{|\mathcal{M} \setminus A_i| - 1}{|\mathcal{M} \setminus A_i|}. \tag{15}$$

Moreover, by Lemma 5.4 we have

$$\mathbb{E}[v_i(A^r_i) - v_i(A_i)] \ge \frac{2n/3}{|\mathcal{M} \setminus A_i|}. \tag{16}$$

Inequality (15) along with Inequality (16) shows

$$\begin{aligned} \mathbb{E}[\mathsf{ex}^{\{2/3\}}(\mathcal{A}^r)] &= \mathbb{E}[\sum_{j \ne i} v_j^{\{2/3\}}(A^r_j)] + \mathbb{E}[v_i(A^r_i)] \\ &\ge \sum_{j \ne i} v_j^{\{2/3\}}(A_j)\frac{|\mathcal{M} \setminus A_i| - 1}{|\mathcal{M} \setminus A_i|} + \mathbb{E}[v_i(A^r_i)] \\ &\ge \sum_{j \ne i} v_j^{\{2/3\}}(A_j)\frac{|\mathcal{M} \setminus A_i| - 1}{|\mathcal{M} \setminus A_i|} + \frac{2n/3}{|\mathcal{M} \setminus A_i|} + v_i(A_i) \\ &\ge \sum_{j \ne i} v_j^{\{2/3\}}(A_j)\frac{|\mathcal{M} \setminus A_i| - 1}{|\mathcal{M} \setminus A_i|} + \frac{2n/3}{|\mathcal{M} \setminus A_i|} + v_i^{\{2/3\}}(A_i) \\ &\ge \sum_{j \ne i} v_j^{\{2/3\}}(A_j)\frac{|\mathcal{M} \setminus A_i| - 1}{|\mathcal{M} \setminus A_i|} + \frac{2n/3}{|\mathcal{M} \setminus A_i|} + v_i^{\{2/3\}}(A_i)\frac{|\mathcal{M} \setminus A_i| - 1}{|\mathcal{M} \setminus A_i|} \\ &= \mathsf{ex}^{\{2/3\}}(\mathcal{A})\frac{|\mathcal{M} \setminus A_i| - 1}{|\mathcal{M} \setminus A_i|} + \frac{2n/3}{|\mathcal{M} \setminus A_i|}. \end{aligned} \tag{17}$$

Recall that by Lemma 2.1, the value of agent a_i for any item alone is bounded by $1/3$ and thus $\mathbb{E}[v_i(A_i^r) - v_i(A_i)] = \mathbb{E}[v_i^{\{2/3\}}(A_i^r) - v_i^{\{2/3\}}(A_i)]$. Notice that by the definition, $v_j^{\{2/3\}}$ is always bounded by $2/3$ and also $v_i(A_i) < 1/3$, therefore, $\mathsf{ex}^{\{2/3\}}(\mathcal{A}) \le 2n/3 - 1/3$ and thus

$$\begin{aligned} \mathbb{E}[\mathsf{ex}^{\{2/3\}}(\mathcal{A}^r)] &\ge \mathsf{ex}^{\{2/3\}}(\mathcal{A})\frac{|\mathcal{M} \setminus A_i| - 1}{|\mathcal{M} \setminus A_i|} + \frac{2n/3}{|\mathcal{M} \setminus A_i|} \\ &\ge \mathsf{ex}^{\{2/3\}}(\mathcal{A}) + \frac{1/3}{|\mathcal{M} \setminus A_i|} \\ &\ge \mathsf{ex}^{\{2/3\}}(\mathcal{A}) + 1/3m. \quad \square \end{aligned} \tag{18}$$

5.1. Algorithm

In this section we give an algorithm to find a $1/3$-MMS allocation for submodular valuations. We show our algorithm runs in time $\mathsf{poly}(n, m)$.

For simplicity, we assume for every agent a_i, MMS_i is given as input to the algorithm. However, computing MMS_i alone is an NP-hard problem. That said, we show in Section 6.2.2 that such a computational barrier can be lifted by a combinatorial trick. We refer the reader to Section 6.2.2 for a more detailed discussion. The procedure is illustrated in Algorithm 1.

Algorithm 1: Finding a $1/3$-MMS allocation for submodular valuations.

Data: $\mathcal{N}, \mathcal{M}, \langle v_1, v_2, \dots, v_n\rangle, \langle \mathsf{MMS}_1, \mathsf{MMS}_2, \dots, \mathsf{MMS}_n\rangle$

For every a_j, scale v_j to ensure $\mathsf{MMS}_j = 1$;
while *there exist an agent a_i and an item b_j such that $v_i(\{b_j\}) \ge 1/3$* **do**
  Allocate $\{b_j\}$ to a_i;
  $\mathcal{M} = \mathcal{M} \setminus b_j$;
  $\mathcal{N} = \mathcal{N} \setminus a_i$;
$\mathcal{A} =$ an arbitrary allocation of the items to the agents;
while $\min v_j^{\{2/3\}}(A_j) < 1/3$ **do**
  $i =$ the agent who receives the lowest value in allocation $\mathcal{A}$;
  Find an item b_e such that: $\mathsf{ex}(\langle A_1 \setminus \{b_e\}, A_2 \setminus \{b_e\}, \dots, A_{i-1} \setminus \{b_e\}, A_i \cup \{b_e\}, A_{i+1} \setminus \{b_e\}, \dots, A_n \setminus \{b_e\}\rangle) \ge \mathsf{ex}(\mathcal{A}) + 1/3m$;
  $\mathcal{A} = \langle A_1 \setminus \{b_e\}, A_2 \setminus \{b_e\}, \dots, A_{i-1} \setminus \{b_e\}, A_i \cup \{b_e\}, A_{i+1} \setminus \{b_e\}, \dots, A_n \setminus \{b_e\}\rangle$;
For every $a_i \in \mathcal{N}$ allocate A_i to a_i;

Based on Theorem 3.2, one can show that in every iteration of the algorithm value of $\mathsf{ex}^{\{2/3\}}(\mathcal{A})$ is increased by at least $1/3m$. Moreover, such an element b_e can be easily found by iterating over all items in time $O(m)$. Furthermore, the number of iterations of the algorithm is bounded by $2nm$, since $\mathsf{ex}^{\{2/3\}}(\mathcal{A})$ is bounded by $2n/3$. Therefore, Algorithm 1 finds a $1/3$-MMS allocation in time $\mathsf{poly}(n, m)$.

Theorem 3.3. *Given access to value-query oracles, one can find a* $1/3$-MMS *allocation for agents with submodular valuations in polynomial time.*

As a corollary of Theorem 3.3, one can show that the problem of finding the maximin value of a submodular function admits a 3 approximation algorithm.

Corollary 5.5. *For a given submodular valuation function v_i, we can in polynomial time split the elements of ground set into n disjoint sets $S_1, S_2, \dots, S_n$ such that*

$$v_i(S_j) \ge \mathsf{MMS}_i/3$$

for every $1 \le j \le n$.

6. XOS valuations

The class of fractionally subadditive (XOS) set functions is a superclass of submodular functions. These functions too, have been the subject of many studies in recent years [38,39,23,40–43,28,44]. Similar to submodular functions, in this section, we show a $1/5$-MMS allocation is possible when all agents have XOS valuations. Furthermore, we complement our proof by providing a polynomial algorithm to find a $1/8$-MMS allocation in Section 6.2.

6.1. Existential proof

In this section we show every instance of the fair allocation problem with XOS valuations admits a $1/5$-MMS allocation. Without loss of generality, we assume $\mathsf{MMS}_i = 1$ for every agent a_i. Recall the definition of ceiling functions. As stated in Lemma 5.2, for every XOS function f and every real number $x \geq 0$, f^x is also XOS. The proof of this section is similar to the result of Section 5. However, the details are different since XOS functions do not adhere to the special structure of submodular functions. For every allocation $\mathcal{A}$, we define $\mathsf{ex}^{\{2/5\}}(\mathcal{A})$ as follows:

$$\mathsf{ex}^{\{2/5\}}(\mathcal{A}) = \sum_{a_i \in \mathcal{N}} v_i^{\{2/5\}}(A_i).$$

Now let $\mathcal{A} = \langle A_1, A_2, \ldots, A_n \rangle$ be an allocation of items to the agents that maximizes $\mathsf{ex}^{\{2/5\}}$. Provided that the problem is $1/5$-irreducible, we show $\mathcal{A}$ is a $1/5$-MMS allocation. Before we proceed to the main proof, we state Lemmas 6.1, and 6.2 as auxiliary observations.

Lemma 6.1. *Let $f(\cdot)$ be an* XOS *set function and let $S \subseteq \mathsf{ground}(f)$. If we divide S into k (possibly empty) sets $S_1, S_2, \ldots, S_k$ then*

$$\sum_{i=1}^{k} \Big(f(S) - f(S \setminus S_i) \Big) \leq f(S).$$

Proof. According to the definition of XOS function, $f(\cdot)$ is an XOS function with a finite set of additive functions $\{g_1, g_2, \ldots, g_\alpha\}$ where $f(S) = \max_{i=1}^{\alpha} g_i(S)$ for any set $S \in \mathsf{ground}(f)$. Let $g_j(\cdot)$ be the additive function which maximizes S. Let $g_j(S_1) = \alpha_1, g_j(S_2) = \alpha_2, \ldots, g_j(S_k) = \alpha_k$, which yields $f(S) = \sum \alpha_i$. Since $g_j(S_i) = \alpha_i$, $f(S \setminus S_i) \geq f(S) - \alpha_i$. Therefore, we have:

$$\begin{aligned} \sum f(S) - f(S \setminus S_i) &\leq \sum f(S) - (f(S) - \alpha_i) \\ &= f(S). \quad \square \end{aligned} \tag{19}$$

By Lemma 2.1, to prove the approximation ratio of $1/5$, we can assume without loss of generality that the value of every item for an agent is bounded by $1/5$. For XOS functions, in Lemma 6.2 we prove another important property.

Lemma 6.2. *Assume that the value of each item for each agent is less than $1/5$. Then, for a given agent a_i we can divide the items into $2n$ sets $S_1, S_2, \ldots, S_{2n}$ such that*

$$v_i(S_j) \geq 2/5$$

for every $1 \leq j \leq 2n$.

Proof. According to the definition of MMS, we know that a_i can divide items to n sets $\mathcal{P} = \langle P_1, P_2, \ldots, P_n \rangle$ such that $V_i(P_j) \geq 1$ for any P_j. The catch is that a_i can divide each of these n sets to two disjoint sets such that the value of each of these new sets be at least $2/5$ to him. Let $T = \{b_1, b_2, \ldots, b_\gamma\}$ be one of these n sets, and $g_j(\cdot)$ be an additive function which maximizes $V_i(T)$. Let $T_k = \{b_1, b_2, \ldots, b_k\}$ for any $1 \leq k \leq \gamma$. Since the value of any item is less than $1/5$ to a_i, there is a set T_k among T_1 to T_γ where $2/5 \leq g_j(T_k) < 3/5$. Since $g_j(\cdot)$ is one of additive functions of XOS function V_i, we have $V_i(T_k) \geq 2/5$. Moreover, since $g_j(T_k) < 3/5$, $g_j(T \setminus T_k) \geq 2/5$, which yields $V_i(T \setminus T_k) \geq 2/5$. As a conclusion, we can divide each of n sets to two disjoint sets with at least $2/5$ value to a_i. $\square$

Next we prove the main theorem of this section.

Theorem 3.5. *The fair allocation problem with* XOS *valuations admits a* $1/5$-MMS *allocation.*

Proof. Consider an allocation $\mathcal{A} = \langle A_1, A_2, \ldots, A_n \rangle$ of items to the agents that maximizes $\mathsf{ex}^{\{2/5\}}$. We show that such an allocation is $1/5$-MMS. Suppose for the sake of contradiction that there exists an agent a_i who receives a set of items which are together worth less than $1/5$ to him. More precisely,

$$v_i^{\{2/5\}}(A_i) = v_i(A_i) < 1/5.$$

Recall that by Lemma 2.1, the value of each item to each agent is less than $1/5$, and therefore by Lemma 6.2, we can divide the items into $2n$ sets $S_1, S_2, \ldots, S_{2n}$ such that $v_i(S_j) \geq 2/5$ for every $1 \leq j \leq 2n$. Note that in this case, $v_i^{\{2/5\}}(S_j) = 2/5$ follows from the definition. Moreover by monotonicity, $v_i^{\{2/5\}}(S_j \cup A_i) = 2/5$ holds for every j.

Now consider $2n$ allocations $\mathcal{A}^1, \mathcal{A}^2, \dots, \mathcal{A}^{2n}$ such that

$$\mathcal{A}^j = \langle A_1^j, A_2^j \dots, A_n^j \rangle$$

for every $1 \le j \le 2n$ where

$$A_k^j = \begin{cases} A_k \cup S_j, & \text{if } k = i \\ A_k \setminus S_j, & \text{if } k \neq i. \end{cases}$$

We show at least one of these allocations has a higher value for $\mathsf{ex}^{\{2/5\}}$ than $\mathcal{A}$. Since $v_i^{\{2/5\}}$ is XOS, by Lemma 6.1 we have

$$\sum_{j=1}^{2n} \left(v_k^{\{2/5\}}(A_k) - v_k^{\{2/5\}}(A_k \setminus S_j) \right) \le v_k^{\{2/5\}}(A_j)$$

for every $a_k \neq a_i$ and thus

$$\begin{aligned} \sum_{j=1}^{2n} v_k^{\{2/5\}}(A_k^j) &= \sum_{j=1}^{2n} v_k^{\{2/5\}}(A_j \setminus S_j) \\ &\ge 2n v_k^{\{2/5\}}(A_k) - v_k^{\{2/5\}}(A_k) \\ &= (2n-1) v_k^{\{2/5\}}(A_k). \end{aligned} \tag{20}$$

Moreover, since $v_i^{\{2/5\}}(A_i) < 1/5$, we have

$$\begin{aligned} \sum_{a_j \neq a_i} v_j^{\{2/5\}}(A_j) &> \sum_{a_j \in \mathcal{N}} v_j^{\{2/5\}}(A_j) - 1/5 \\ &= \mathsf{ex}^{\{2/5\}}(\mathcal{A}) - 1/5. \end{aligned} \tag{21}$$

Furthermore, since $v_i^{\{2/5\}}(S_j \cup A_i) = 2/5$ for every $1 \le j \le 2n$, we have

$$\begin{aligned} \sum_{a_k \neq a_i} v_k^{\{2/5\}}(A_k^j) &= \sum_{a_k \in \mathcal{N}} v_k^{\{2/5\}}(A_k^j) - 2/5 \\ &= \mathsf{ex}^{\{2/5\}}(\mathcal{A}^j) - 2/5. \end{aligned} \tag{22}$$

Finally, by combining Inequalities (20), (21), and (22) we have

$$\begin{aligned} \sum_{j=1}^{2n} \mathsf{ex}^{\{2/5\}}(\mathcal{A}^j) &= \sum_{j=1}^{2n} (2/5 + \sum_{a_k \neq a_i} v_k^{\{2/5\}}(A_k^j)) \\ &= 4n/5 + \sum_{j=1}^{2n} \sum_{a_k \neq a_i} v_k^{\{2/5\}}(A_k^j) \\ &\ge 4n/5 + \sum_{a_k \neq a_i} (2n-1) v_k^{\{2/5\}}(A_k) \\ &\ge 4n/5 + (2n-1)(\mathsf{ex}^{\{2/5\}}(\mathcal{A}) - 1/5) \\ &\ge 2n \cdot \mathsf{ex}^{\{2/5\}}(\mathcal{A}) + (4n - 2n + 1)/5 - \mathsf{ex}^{\{2/5\}}(\mathcal{A}) \\ &\ge 2n \cdot \mathsf{ex}^{\{2/5\}}(\mathcal{A}) + (2n+1)/5 - \mathsf{ex}^{\{2/5\}}(\mathcal{A}). \end{aligned}$$

Now notice that since $v_k^{\{2/5\}}(A_k) \le 2/5$, we have

$$\begin{aligned} \mathsf{ex}^{\{2/5\}}(\mathcal{A}) &= \sum_{k=1}^{n} v_k^{\{2/5\}}(A_k) \\ &\le \sum_{k=1}^{n} 2/5 \\ &\le 2n/5 \end{aligned}$$

and thus

$$\sum_{j=1}^{2n} \mathsf{ex}^{\{2/5\}}(\mathcal{A}^j) \geq 2n \cdot \mathsf{ex}^{\{2/5\}}(\mathcal{A}) + (2n+1)/5 - \mathsf{ex}^{\{2/5\}}(\mathcal{A})$$
$$\geq 2n \cdot \mathsf{ex}^{\{2/5\}}(\mathcal{A}) + (2n+1)/5 - 2n/5$$
$$\geq 2n \cdot \mathsf{ex}^{\{2/5\}}(\mathcal{A}) + 1/5.$$

Therefore, $\mathsf{ex}^{\{2/5\}}(\mathcal{A}^j) > \mathsf{ex}^{\{2/5\}}(\mathcal{A}) + 1/10n$ holds for at least one $\mathcal{A}^j$ which contradicts the maximality of $\mathcal{A}$. □

6.2. Algorithm

In this section, we provide a polynomial time algorithm for finding a 1/8-MMS allocation for the fair allocation problem with XOS valuations. The algorithm is based on a similar idea that we argued for the proof of Theorem 3.5. Remark that our algorithm only requires access to demand-query and XOS oracles. It does *not* have any additional information about the maximin values. This makes the problem computationally harder since computing the maximin values is NP-hard [45]. We begin by giving a high-level intuition of the algorithm and show that we can overcome computational obstacles by combinatorial tricks. Consider the pseudo-code described in Algorithm 2.

Algorithm 2: Algorithm for finding a 1/8-MMS allocation.

Data: $\mathcal{N}, \mathcal{M}, \langle v_1, v_2, \dots, v_n \rangle$
1 For every a_j, scale v_j to ensure $\mathsf{MMS}_j = 1$;
2 **while** *there exist an agent a_i and an item b_j such that $v_i(\{b_j\}) \geq 1/8$* **do**
3   Allocate $\{b_j\}$ to a_i;
4     $\mathcal{M} = \mathcal{M} \setminus b_j$;
5     $\mathcal{N} = \mathcal{N} \setminus a_i$;
6 $\mathcal{A} =$ an arbitrary allocation of the items to the agents;
7 **while** $\min v_j^{\{1/4\}}(A_j) < 1/8$ **do**
8     $i =$ the agent who receives the lowest value in allocation $\mathcal{A}$;
9   Find a set S such that: $\mathsf{ex}^{\{1/4\}}(\langle A_1 \setminus S, A_2 \setminus S, \dots, A_{i-1} \setminus S, A_i \cup S, A_{i+1} \setminus S, \dots, A_n \setminus S \rangle) \geq \mathsf{ex}^{\{1/4\}}(\mathcal{A}) + 1/12n$;
10     $\mathcal{A} = \langle A_1 \setminus S, A_2 \setminus S, \dots, A_{i-1} \setminus S, A_i \cup S, A_{i+1} \setminus S, \dots, A_n \setminus S \rangle$;
11 For every $a_i \in \mathcal{N}$ allocate A_i to a_i;

As we show in Section 6.2.1, Command 9 of the algorithm is always doable. More precisely, there always exists a set S that holds in the condition of Command 9. Notice that in every step of the algorithm, $\mathsf{ex}^{\{1/4\}}(\mathcal{A})$ is increased by at least $1/12n$ and this value is bounded by $1/4 \cdot n = n/4$. Therefore the algorithm terminates after at most $3n^2$ steps and the allocation is guaranteed to be 1/8-MMS.

That said, there are two major computational obstacles in the way of running Algorithm 2. Firstly, finding a set S that satisfies the condition of Command 9 can not be trivially done in polynomial time. Second, scaling the valuation functions to ensure $\mathsf{MMS}_i = 1$ for all agents is NP-hard and cannot be done in polynomial time unless P=NP. To overcome the former, in Section 6.2.1 we provide an algorithm for finding such a set S in polynomial time. Next, in Section 6.2.2, we present a combinatorial trick to run the algorithm in polynomial time without having to deal with NP-hardness of scaling the valuation functions.

6.2.1. Executing command 9 in polynomial time

In this section we present an algorithm to execute Command 9 of Algorithm 2. We show that such a procedure can be implemented via demand-query oracles.

Let for every $b_j \notin A_i$, c_j be the amount of contribution that b_j makes to $\mathsf{ex}^{\{1/4\}}(\mathcal{A})$. We set $p_e = 3(n/(n-1))c_e$ and ask the demand-query oracle of v_i to find a set S that maximizes $v_i(S) - \sum_{b_j \in S} p_j$. Via a trivial calculation, one can show that $v_i(S) - \sum_{b_j \in S} p_j \geq 1/4$ holds for at least one set of items. The reason this is correct is that one can divide the items into n partitions where each is worth at least 1 to a_i. Moreover, the summation of prices for the items is bounded by $3n/(n-1) \cdot (\sum_{j \neq i} v_j^{\{1/4\}}(A_j)) \leq 3n/4$. Therefore, for at least one of those partitions $v_i(S) - \sum_{b_j \in S} p_j$ is at least 1/4. Thus, the set that the oracle reports is worth at least 1/4 to a_i.

Now, let S^* be the set that the oracle reports and for every $b_j \in S^*$, c_j^* be the contribution of b_j to $v_i(S^*)$. We sort the items of S^* based on $c_j^* - p_j$ in non-increasing order. Next, we start with an empty bag and add the items in their order to the bag until the total value of the items in the bag to a_i reaches 1/4. Since the value of every item alone is bounded by 1/8, the total value of the items in the bag to a_i is bounded by 3/8. Thus the contribution of those items to $\mathsf{ex}^{\{1/4\}}(\mathcal{A})$ is at most $(3/8)/(3n/(n-1)) \leq 1/8 - 1/(10n)$. Therefore, removing items of the bag from other allocations and adding them to A_i, increases $\mathsf{ex}^{\{1/4\}}(\mathcal{A})$ by at least $1/10n$.

We remark that one can use the same argument to prove this even if $\mathsf{MMS}_i \geq 1/(1 + 1/10n)$.

6.2.2. Running Algorithm 2 in polynomial time

As mentioned above, scaling valuation functions to ensure $\mathsf{MMS}_i = 1$ for every agent a_i is an NP-hard problem since determining the maximin values is hard even for additive agents [9]. Therefore, unlike Section 6.2.1, in this section we massage the algorithm to make it executable in polynomial time.

Suppose an oracle gives us the maximin values of the agents. Provided that we can run Command 9 of Algorithm 2 in polynomial time, we can find a 1/8-MMS allocation in polynomial time. Therefore, in case the oracle reports the actual maximin values, the solution is trivial. However, what if the oracle has an error in its calculations? There are two possibilities: (i) Algorithm 2 terminates and finds an allocation which is 1/8-MMS with respect to the reported maximin values. (ii) The algorithm fails to execute Command 9, since no such set S holds in the condition of Command 9. The intellectual merit of this section boils down to investigation of the case when algorithm fails to execute Command 9. We show that this only happens due to an overly high misrepresentation of the maximin value for agent a_i. Note that a_i is the agent who receives the lowest value in the last cycle of the execution.

Observation 6.1. *Given $\langle d_1, d_2, \ldots, d_n\rangle$ as an estimate for the maximin values, if Algorithm 2 fails to execute Command 9 for an agent a_i, then we have*

$$d_i \geq (1 + 1/10n)\mathsf{MMS}_i.$$

Proof of Observation 6.1 follows from the argument of Section 6.2.1. More precisely, as mentioned in Section 6.2.1, such a set S exists, if $\mathsf{MMS}_i \geq 1/(1 + 1/10n)$. Thus, given that the procedure explained in Section 6.2.1 fails to find such a set, one can conclude that the reported value for MMS_i is at least $(1/(1 + 1/10n))$ times its actual value. Based on Observation 6.1, we propose Algorithm 3 for implementing a maximin oracle.

Algorithm 3: Implementing a maximin oracle.

Data: $\mathcal{N}, \mathcal{M}, \langle v_1, v_2, \ldots, v_n\rangle$
for *every* $a_i \in \mathcal{N}$ **do**
    $d_i \leftarrow v_i(\mathcal{M})$;
while *true* **do**
    Run Algorithm 2 assuming maximin values are $d_1, d_2, \ldots, d_n$;
    if *the Algorithm fails to run Command 9 for an agent* a_i **then**
        $d_i \leftarrow d_i/(1 + 1/10n)$;
    else
        Report the allocation and terminate the algorithm;

Note that in the beginning of the algorithm, we set $d_i = v_i(\mathcal{M})$ which is indeed greater than or equal to MMS_i. By Lemma 6.1, every time we decrease the value of d_i for an agent a_i, we preserve the condition $d_i \geq \mathsf{MMS}_i$ for that agent. Therefore, in every step of the algorithm, we have $d_i \geq \mathsf{MMS}_i$ and thus the reported allocation which is 1/8-MMS with respect to d_i's is also 1/8-MMS with respect to true maximin values. Thus, the algorithm provides a correct 1/8-MMS allocation in the end. All that remains is to show the running time of the algorithm is polynomial.

Notice that every time we decrease d_i for an agent a_i, we multiply this value by $1/(1+1/10n)$, hence the number of such iterations is polynomial in n, unless the valuations are super-exponential in n. Since we always assume the input numbers are represented by $\mathsf{poly}(n)$ bits, the number of iterations is bounded by $\mathsf{poly}(n)$ and hence the algorithm terminates after a polynomial number of steps.

Theorem 3.6. *Given access to demand-query and* XOS *oracles, there exists a polynomial time algorithm that finds a* 1/8-MMS *allocation for agents with* XOS *valuations.*

A consequence of Theorem 3.6 is an 8-approximation algorithm for determining the maximin value of an agent with XOS valuation.

Corollary 6.3. *For a given* XOS *valuation function v_i, we can in polynomial time split the elements of ground set into n disjoint sets $S_1, S_2, \ldots, S_n$ such that*

$$v_i(S_j) \geq \mathsf{MMS}_i/8$$

Proof. We construct an instance of the fair allocation problem with n agents, all of whom have a valuation function equal to f. We find a 1/8-MMS allocation of the items to the agents in polynomial time and report the minimum value that an agent receives as output.

The 1/8 guarantee follows from the fact that every agent receives a subset of values that are worth $1/8\text{-}\mathsf{MMS}_i$ to him. □

Remark 6.4. *A similar procedure can also be used to overcome the challenge of computing the maximin values for the algorithm described in Section 5.1.*

7. Subadditive valuations

In this section, we present an existential proof based on a well-known reduction from subadditive setting to the XOS setting (see [39] for example). More precisely, we show for every subadditive set function $f(\cdot)$, there exists an XOS function $g(\cdot)$, where g is dominated by f but the maximin value of g is within a logarithmic factor of the maximin value of f. We begin with an observation. Suppose we are given a subadditive function f on set $\text{ground}(f)$, and we wish to approximate f with an additive function g which is dominated by f. In other words, we wish to find an additive function g such that

$$\forall S \subseteq \text{ground}(f) \qquad g(S) \le f(S)$$

and $g(\text{ground}(f))$ is maximized. One way to formulate g is via a linear program. Suppose $\text{ground}(f) = \{b_1, b_2, \dots, b_m\}$ and let $g_1, g_2, \dots, g_m$ be m variables that describe g in the following way:

$$\forall S \subseteq \text{ground}(f) \qquad g(S) = \sum_{b_i \in S} g_i.$$

Based on this formulation, we can find the optimal additive function g by LP (23).

$$\begin{aligned} \text{maximize:} \quad & \sum_{b_i \in \text{ground}(f)} g_i \\ \text{subject to:} \quad & \sum_{b_i \in S} g_i \le f(S) \qquad \forall S \subseteq \text{ground}(f) \\ & g_i \ge 0 \qquad \forall b_i \in \text{ground}(f) \end{aligned} \tag{23}$$

We show the objective function of LP (23) is lower bounded by $f(\text{ground}(f))/\log m$. The basic idea is to first write the dual program and then based on a probabilistic method, lower bound the optimal value of the dual program by $f(\text{ground}(f))/\log m$.

Lemma 7.1. *The optimal solution of LP (23) is at least $f(\text{ground}(f))/\log m$.*

Proof. To prove the lemma, we write the dual of LP (23) as follows:

$$\begin{aligned} \text{minimize:} \quad & \sum_{S \subseteq \text{ground}(f)} \alpha_S f(S) \\ \text{subject to:} \quad & \sum_{S \ni b_i} \alpha_S \ge 1 \qquad \forall b_i \in \text{ground}(f) \\ & \alpha_S \ge 0 \qquad \forall S \subseteq \text{ground}(f) \end{aligned} \tag{24}$$

By the strong duality theorem, the optimal solutions of LP (23) and LP (24) are equal. Next, based on the optimal solution of LP (24), we define a randomized procedure to draw a set of elements: We start with an empty set S^* and for every set $S \subseteq \text{ground}(f)$ we add *all* elements of S to S^* with probability α_S. Since f is subadditive, the marginal increase of $f(S^*)$ by adding elements of a set S to S^* is bounded by $f(S)$ and thus the expected value of $f(S^*)$ is bounded by the objective of LP (24). In other words:

$$\mathbb{E}[f(S^*)] \le \sum_{S \subseteq \text{ground}(f)} \alpha_S f(S) \tag{25}$$

We remark that we repeat this procedure for all subsets of $\text{ground}(S)$ independently and thus for every $b_i \in \text{ground}(f)$, $\sum_{S \ni b_i} \alpha_S \ge 1$ holds, we have

$$\text{PR}[b_i \in S^*] \ge 1 - 1/e \simeq 0.632121 > 1/2 \tag{26}$$

for every element $b_i \in \text{ground}(S)$. Now, with the same procedure, we draw $\lceil \log m \rceil + 2$ sets $S^*_1, S^*_2, \dots, S^*_{\lceil \log m \rceil + 2}$ *independently*. We define $\hat{S} = \bigcup S^*_i$. By Inequality (26) and the union bound we show

$$\begin{aligned}
\mathsf{PR}[\hat{S} = \mathsf{ground}(f)] &\geq 1 - \sum_{b_i \in \mathsf{ground}(i)} \mathsf{PR}[b_i \notin \hat{S}] \\
&= 1 - \sum_{b_i \in \mathsf{ground}(i)} \mathsf{PR}[b_i \notin S_1^* \text{ and } b_i \notin S_1^* \text{ and } \ldots \text{ and } b_i \notin S^*_{\lceil \log m \rceil + 2}] \\
&= 1 - \sum_{b_i \in \mathsf{ground}(i)} \prod_{j=1}^{\lceil \log m \rceil + 2} \mathsf{PR}[b_i \notin S_j^*] \\
&\geq 1 - \sum_{b_i \in \mathsf{ground}(i)} \prod_{j=1}^{\lceil \log m \rceil + 2} 1/2 \\
&= 1 - \sum_{b_i \in \mathsf{ground}(i)} \prod_{j=1}^{\lceil \log m \rceil + 2} \mathsf{PR}[b_i \notin S_j^*] \\
&\geq 1 - \sum_{b_i \in \mathsf{ground}(i)} 1/4m \\
&= 1 - 1/4 \\
&= 3/4
\end{aligned}$$

and thus $\mathbb{E}[f(\hat{S})] \geq 3/4 f(\mathsf{ground}(f))$. On the other hand, by the linearity of expectation and the fact that f is subadditive we have:

$$\begin{aligned}
\mathbb{E}[f(\hat{S})] &= \mathbb{E}[f(\bigcup S_i^*)] \\
&\leq \mathbb{E}[\sum f(S_i^*)] \\
&\leq (\lceil \log m \rceil + 2)(\sum_{S \subseteq \mathsf{ground}(f)} \alpha_S f(S)).
\end{aligned}$$

Therefore $\sum_{S \subseteq \mathsf{ground}(f)} \alpha_S f(S) \geq 3/4 f(\mathsf{ground}(f))/(\lceil \log m \rceil + 2)$, which means

$$\sum_{S \subseteq \mathsf{ground}(f)} \alpha_S f(S) \geq f(\mathsf{ground}(f))/(2\lceil \log m \rceil)$$

for a big enough m. This shows the optimal solution of LP (23) is lower bounded by $f(\mathsf{ground}(f))/(2\lceil \log m \rceil)$ and the proof is complete. □

In what follows, based on Lemma 7.1, we provide a reduction from subadditive valuations to XOS valuations. An immediate corollary of Lemma 7.1 is the following:

Corollary 7.2 *(of Lemma 7.1). For any subadditive valuation function V_i and integer number n, there exists an* XOS *function V_i' such that*

$$V_i'(S) \leq V_i(S) \qquad \forall S \subseteq \mathcal{M}$$

and

$$\mathsf{MMS}_i' \geq \mathsf{MMS}_i/2\lceil \log m \rceil$$

Based on Theorem 3.5 and Corollary 7.2 one can show that a $1/10\lceil \log m \rceil$-MMS allocation is always possible for subadditive valuations.

Theorem 3.7. *Any fair allocation problem in which the agents have subadditive valuations admits a $1/10\lceil \log m \rceil$-MMS allocation.*

8. Related work

As mentioned, maximin-share was introduced by Budish [1] and since then has been the subject of many studies [2, 46–52,16,9]. Other than the results we mentioned in the introduction, there are results that consider maximin-share for different allocation scenarios. For example, Kurokawa et al. [48] show that when the valuations are drawn at random, an allocation with maximin-share guarantee exists with a high probability, and it can be found in polynomial time. Bouveret and Lemaître in [53] show that for the restricted cases when the valuations of the agents come from $\{0,1\}$, or when the number of items is smaller than $n+3$, an MMS allocation is guaranteed to exist. For the case of 3 agents, the approximation factor is improved from 2/3 in series of work to 3/4 [9], 7/8 [16], and 8/9 [47].

There are other studies that extend maximin-share to more general settings. Similar to classic fairness notions, the weighted version of MMS (WMMS) is also considered [49]. The current best approximation guarantee for WMMS is 1/2 by Farhadi et al. [49]. Gourvès and Monnot [47] also extend the maximin share problem to the case that the goods collectively received by the agents satisfy a matroidal constraint. They prove that for this case, a 1/2-MMS allocation is always possible.

It is worth mentioning that other than maximin share, there are other fairness criteria that attracted considerable attention, especially in recent years: envy-free up to one good (EF1) and envy-free up to any good (EFX). In these settings, we seek to find allocations with limited (but not necessarily zero) envy between the agents [54,55,14]. Also, recent studies have established a connection between *Nash Social Welfare* (NSW) and these fairness criteria [55,14]. NSW is defined as the geometric mean of the agents' utilities. Maximizing NSW has been the subject of many recent studies [55–58].

Subsequent work

To the best of our knowledge, all the results in this paper are currently the best approximation guarantees for MMS beyond the additive setting. For a special case of XOS setting where the valuations form hereditary (or downward closed) set system, Li and Vetta improved the approximation factor to 11/30. For additive setting, the recently best approximation guarantee is the work of Garg and Taki [18] that prove the approximation guarantee of $3/4+o(1)$ for the additive setting.

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.

References

[1] E. Budish, The combinatorial assignment problem: approximate competitive equilibrium from equal incomes, J. Polit. Econ. 119 (6) (2011) 1061–1103.
[2] M. Ghodsi, M. HajiAghayi, M. Seddighin, S. Seddighin, H. Yami, Fair allocation of indivisible goods: improvements and generalizations, in: Proceedings of the 2018 ACM Conference on Economics and Computation, ACM, 2018, pp. 539–556.
[3] S.J. Brams, A.D. Taylor, An envy-free cake division protocol, Am. Math. Mon. (1995) 9–18.
[4] H. Steinhaus, The problem of fair division, Econometrica 16 (1) (1948).
[5] S.J. Brams, A.D. Taylor, Fair Division: From Cake-Cutting to Dispute Resolution, Cambridge University Press, 1996.
[6] L.E. Dubins, E.H. Spanier, How to cut a cake fairly, Am. Math. Mon. (1961) 1–17.
[7] I. Bezáková, V. Dani, Allocating indivisible goods, ACM SIGecom Exch. 5 (3) (2005) 11–18.
[8] D.K. Foley, Resource allocation and the public sector, Yale Economic Essays.
[9] D. Kurokawa, A.D. Procaccia, J. Wang, Fair enough: guaranteeing approximate maximin shares, J. ACM 65 (2) (2018) 8.
[10] R.J. Lipton, E. Markakis, E. Mossel, A. Saberi, On approximately fair allocations of indivisible goods, in: Proceedings of the 5th ACM Conference on Electronic Commerce, ACM, 2004, pp. 125–131.
[11] W. Stromquist, How to cut a cake fairly, Am. Math. Mon. 87 (8) (1980) 640–644.
[12] S. Even, A. Paz, A note on cake cutting, Discrete Appl. Math. 7 (3) (1984) 285–296.
[13] V. Conitzer, R. Freeman, N. Shah, Fair public decision making, in: Proceedings of the 2017 ACM Conference on Economics and Computation, 2017.
[14] I. Caragiannis, D. Kurokawa, H. Moulin, A.D. Procaccia, N. Shah, J. Wang, The unreasonable fairness of maximum Nash welfare, in: Proceedings of the 2016 ACM Conference on Economics and Computation, ACM, 2016, pp. 305–322.
[15] J. Garg, S. Taki, An improved approximation algorithm for maximin shares, Artif. Intell. (2021) 103547.
[16] G. Amanatidis, E. Markakis, A. Nikzad, A. Saberi, Approximation algorithms for computing maximin share allocations, ACM Trans. Algorithms 13 (4) (2017) 52.
[17] S. Barman, S.K. Krishna Murthy, Approximation algorithms for maximin fair division, in: Proceedings of the 2017 ACM Conference on Economics and Computation, ACM, 2017, pp. 647–664.
[18] J. Garg, S. Taki, An improved approximation algorithm for maximin shares, in: Proceedings of the 21st ACM Conference on Economics and Computation, 2020, pp. 379–380.
[19] U. Feige, A. Sapir, L. Tauber, A tight negative example for mms fair allocations, arXiv preprint, arXiv:2104.04977.
[20] J. Garg, P. Kulkarni, R. Kulkarni, Approximating Nash social welfare under submodular valuations through (un)matchings, in: Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, 2020, pp. 2673–2687.
[21] W. Li, J. Vondrák, A constant-factor approximation algorithm for Nash social welfare with submodular valuations, arXiv preprint, arXiv:2103.10536.
[22] M. Bateni, M. Hajiaghayi, M. Zadimoghaddam, Submodular secretary problem and extensions, ACM Trans. Algorithms 9 (4) (2013) 32.
[23] U. Feige, On maximizing welfare when utility functions are subadditive, SIAM J. Comput. 39 (1) (2009) 122–142.
[24] U. Feige, V.S. Mirrokni, J. Vondrak, Maximizing non-monotone submodular functions, in: Foundations of Computer Science, 2007, FOCS'07, 48th Annual IEEE Symposium, IEEE, 2007, pp. 461–471.
[25] D. Golovin, Max-min fair allocation of indivisible goods.
[26] S. Dobzinski, N. Nisan, M. Schapira, Approximation algorithms for combinatorial auctions with complement-free bidders, in: Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing, ACM, 2005, pp. 610–618.
[27] U. Feige, J. Vondrak, Approximation algorithms for allocation problems: improving the factor of 1-1/e, in: Foundations of Computer Science, 2006, FOCS'06, 47th Annual IEEE Symposium, IEEE, 2006, pp. 667–676.
[28] M. Feldman, N. Gravin, B. Lucier, Combinatorial auctions via posted prices, in: Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics, 2015, pp. 123–135.
[29] R.P. Leme, Gross substitutability: an algorithmic survey, Games Econ. Behav. 106 (2017) 294–316.
[30] J. Vondrák, Optimal approximation for the submodular welfare problem in the value oracle model, in: Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing, ACM, 2008, pp. 67–74.

[31] N. Buchbinder, M. Feldman, J. Naor, R. Schwartz, A tight linear time (1/2)-approximation for unconstrained submodular maximization, in: Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium, IEEE, 2012, pp. 649–658.
[32] S. Fujishige, Submodular Functions and Optimization, vol. 58, Elsevier, 2005.
[33] A. Gupta, A. Roth, G. Schoenebeck, K. Talwar, Constrained non-monotone submodular maximization: offline and secretary algorithms, in: International Workshop on Internet and Network Economics, Springer, 2010, pp. 246–257.
[34] G. Kim, E.P. Xing, L. Fei-Fei, T. Kanade, Distributed cosegmentation via submodular optimization on anisotropic diffusion, in: Computer Vision (ICCV), 2011 IEEE International Conference, IEEE, 2011, pp. 169–176.
[35] A. Krause, Sfo: a toolbox for submodular function optimization, J. Mach. Learn. Res. 11 (2010) 1141–1144.
[36] J. Lee, V.S. Mirrokni, V. Nagarajan, M. Sviridenko, Non-monotone submodular maximization under matroid and knapsack constraints, in: Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing, ACM, 2009, pp. 323–332.
[37] M. Minoux, Accelerated greedy algorithms for maximizing submodular set functions, in: Optimization Techniques, Springer, 1978, pp. 234–243.
[38] G. Christodoulou, A. Kovács, M. Schapira, Bayesian combinatorial auctions, in: International Colloquium on Automata, Languages, and Programming, Springer, 2008, pp. 820–832.
[39] K. Bhawalkar, T. Roughgarden, Welfare guarantees for combinatorial auctions with item bidding, in: Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics, 2011, pp. 700–709.
[40] L. Blumrosen, S. Dobzinski, Welfare maximization in congestion games, in: Proceedings of the 7th ACM Conference on Electronic Commerce, ACM, 2006, pp. 52–61.
[41] V. Syrgkanis, Bayesian games and the smoothness framework, arXiv preprint, arXiv:1203.5155.
[42] M. Feldman, H. Fu, N. Gravin, B. Lucier, Simultaneous auctions are (almost) efficient, in: Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing, 2013, pp. 201–210.
[43] H. Fu, R. Kleinberg, R. Lavi, Conditional equilibrium outcomes via ascending price processes with applications to combinatorial auctions with item bidding, in: ACM EC, 2012.
[44] I. Milchtaich, Congestion games with player-specific payoff functions, Games Econ. Behav. 13 (1) (1996) 111–124.
[45] L. Epstein, A. Levin, An efficient polynomial time approximation scheme for load balancing on uniformly related machines, Math. Program. 147 (1–2) (2014) 1–23.
[46] T. Heinen, N.-T. Nguyen, T.T. Nguyen, J. Rothe, Approximation and complexity of the optimization and existence problems for maximin share, proportional share, and minimax share allocation of indivisible goods, Auton. Agents Multi-Agent Syst. 32 (6) (2018) 741–778.
[47] L. Gourvès, J. Monnot, Approximate maximin share allocations in matroids, in: International Conference on Algorithms and Complexity, Springer, 2017, pp. 310–321.
[48] D. Kurokawa, A.D. Procaccia, J. Wang, When can the maximin share guarantee be guaranteed?, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 30, 2016.
[49] A. Farhadi, M. Hajiaghayi, M. Ghodsi, S. Lahaie, D. Pennock, M. Seddighin, S. Seddighin, H. Yami, Fair allocation of indivisible goods to asymmetric agents, in: Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, International Foundation for Autonomous Agents and Multiagent Systems, 2017, pp. 1535–1537.
[50] H. Aziz, G. Rauchecker, G. Schryen, T. Walsh, Algorithms for max-min share fair allocation of indivisible chores, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 31, 2017.
[51] G. Amanatidis, G. Birmpas, G. Christodoulou, E. Markakis, Truthful allocation mechanisms without payments: characterization and implications on fairness, in: Proceedings of the 2017 ACM Conference on Economics and Computation, ACM, 2017, pp. 545–562.
[52] G. Amanatidis, G. Birmpas, E. Markakis, On truthful mechanisms for maximin share allocations, in: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, AAAI Press, 2016, pp. 31–37.
[53] S. Bouveret, M. Lemaître, Characterizing conflicts in fair division of indivisible goods using a scale of criteria, Auton. Agents Multi-Agent Syst. 30 (2) (2016) 259–290.
[54] B. Plaut, T. Roughgarden, Almost envy-freeness with general valuations, in: Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, 2018, pp. 2584–2603.
[55] S. Barman, S.K. Krishnamurthy, R. Vaish, Finding fair and efficient allocations, in: Proceedings of the 2018 ACM Conference on Economics and Computation, 2018, pp. 557–574.
[56] S. Barman, S.K. Krishnamurthy, R. Vaish, Greedy algorithms for maximizing Nash social welfare, in: Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, 2018, pp. 7–13.
[57] N. Anari, T. Mai, S.O. Gharan, V.V. Vazirani, Nash social welfare for indivisible items under separable, piecewise-linear concave utilities, in: Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, 2018, pp. 2274–2290.
[58] R. Cole, N. Devanur, V. Gkatzelis, K. Jain, T. Mai, V.V. Vazirani, S. Yazdanbod, Convex program duality, Fisher markets, and Nash social welfare, in: Proceedings of the 2017 ACM Conference on Economics and Computation, ACM, 2017, pp. 459–460.