Approximating Maximin Share Allocations

Jugal Garg
University of Illinois at Urbana-Champaign
jugal@illionis.edu

Peter McGlaughlin
University of Illinois at Urbana-Champaign
mcglghl2@illionis.edu

Setareh Taki
University of Illinois at Urbana-Champaign
staki2@illionis.edu

Abstract

We study the problem of fair allocation of M indivisible items among N agents using the popular notion of maximin share as our measure of fairness. The maximin share of an agent is the largest value she can guarantee herself if she is allowed to choose a partition of the items into N bundles (one for each agent), on the condition that she receives her least preferred bundle. A maximin share allocation provides each agent a bundle worth at least their maximin share. While it is known that such an allocation need not exist [9, 7], a series of work [9, 8, 1, 2] provided 2/3 approximation algorithms in which each agent receives a bundle worth at least 2/3 times their maximin share. Recently, [6] improved the approximation guarantee to 3/4. Prior works utilize intricate algorithms, with an exception of [2] which is a simple greedy solution but relies on sophisticated analysis techniques. In this paper, we propose an alternative 2/3 maximin share approximation which offers both a simple algorithm and straightforward analysis. In contrast to other algorithms, our approach allows for a simple and intuitive understanding of why it works.

2012 ACM Subject Classification Theory of computation → Algorithmic game theory

Keywords and phrases Fair division, Maximin share, Approximation algorithm

Digital Object Identifier 10.4230/OASIcs.SOSA.2019.20

Funding Work on this paper partly supported by NSF CRII Award 1755619.

1 Introduction

We study the problem of allocating M indivisible items among N agents with additive valuations in a fair way, using the popular notion of maximin share [5] as our measure for fairness. There is an extensive literature for fair allocation of divisible items, starting with the cake cutting problem [10]. Standard notions of fairness include: envy-freeness where every agent prefers their allocation over any other agents' allocation, and proportionality where every agent receives at least a $1/N$ share of all the items.

In the case of indivisible items, a simple counter example shows that no algorithm can provide either envy-freeness or proportionality. Consider allocating a single item between $N > 1$ agents. Clearly, $N - 1$ agents envy the one lucky agent that received the item and there is no way to ensure all agents receive a bundle of items with value at least $1/N$. This motivates the need for an alternate concept of fairness. Recently, Budish [5] introduced an intriguing option, a maximin share. The idea is a natural generalization of the well known cut and choose protocol in the cake cutting problem. Suppose we allow agent i to choose a partition of the items into N bundles (one for each agent), with the caveat that the other

$N-1$ agents get to choose a bundle before her. In the worst case, she receives her least preferred bundle. Clearly, in this situation i will choose a partition that maximizes the value of her least preferred bundle. We call the value of this bundle i's maximin share (MMS). Since all other agents may have the same valuations as her, i's MMS is the most she can guarantee for herself in this scenario. In this paper, we focus on the case of additive valuations. Let us note that, computing any agent's MMS is NP-hard, but a PTAS exists [11].

A maximin share gives an intuitive local measure of fairness of an allocation, that is a specific objective for each agent. This raises the natural question: Is there an allocation where each agent receives a bundle worth at least their MMS? An allocation satisfying this property is said to be maximin share allocation (MMS allocation), and if it exists, an MMS allocation provides strong fairness guarantees to each individual agent. Bouveret and Lemaître [3] show that an MMS allocation always exist in some special settings, e.g., when there are only two agents or if agents' valuations for items are 0 or 1, but leave the general case as an open problem.

Procaccia and Wang [9] obtain the surprising result that MMS allocations might not exist, by means of a clever counter example. However, they show that a 2/3 MMS allocation always exists, that is an allocation where each agent receives a bundle worth at least 2/3 of their maximin share, and they provide a polynomial time algorithm to find a 2/3 MMS allocation when the number of agents N is constant. In the special case where $N \leq 4$, their algorithm finds a 3/4 MMS allocation. Amanatidis et al. [1] improve this result by addressing the requirement for a constant number of agents, obtaining a PTAS which finds a $(2/3 - \epsilon)$ MMS allocation for an arbitrary number of agents; see [8] for an alternate proof. [1] also shows that a 7/8 MMS allocation always exists when there are three agents.

Barman and Murthy [2] take an alternate approach from [9, 1], utilizing key insights from [3] to obtain a greedy algorithm to find a 2/3 MMS allocation. While the algorithm itself is fairly simple, the proof is not.

The recent results of Ghodsi et al. [6] breaks new ground, establishing existence of 3/4 MMS allocation, and, building on the work of [9, 1], provides a PTAS to find a $(3/4 - \epsilon)$ MMS allocation. They also show that when $N = 4$, an 4/5 MMS allocation exists and proposed a algorithm to find it.

Our Contribution. We present an algorithm to find a 2/3 MMS allocation for agents with additive valuations. Our approach combines the insights of [3, 2] with the concepts developed in [6] to obtain an algorithm that is both simple to implement and analyze. Like [6], our algorithm consists of two phases: *matching* and *bag filling*. However, unlike [6], our phases are much simpler, and we do not need to compute agents' MMS values. Bag filling is a simple, greedy method to allocate 'low' valued items. We add one item at a time to a bag. If the value of the bag is worth at least 2/3 of any agent's MMS, then we assign the bag to that agent, picking an arbitrary agent when there are more than one satisfying this condition. It is easily shown, in Section 2.2, that bag filling provides a 2/3 MMS allocation as long as no agent values any item more than 1/3. Thus, the real difficultly lies in distributing 'high' value items, i.e., items worth more than 1/3 to some agent. Drawing on the insights of [3, 2], we show a combination of maximum matching and greedy assignment suffices for this purpose. This gives our algorithm the basic structure: repeated maximum matching and greedy assignment to remove high valued items, followed by bag filling to allocate low valued items. Our approach allows for far simpler and more intuitive analysis than [1, 2, 6].

2 Preliminaries

We consider the fair allocation of $M = \{1, \ldots, m\}$ indivisible items among $N = \{1, \ldots, n\}$ agents with additive valuations. That is, v_{ij} is agent i's value for item j, and i's valuation of any bundle of items $S \subseteq M$ is: $v_i(S) = \sum_{j \in S} v_{ij}$. For simplicity, we also use $v_i(j)$ instead of $v_i(\{j\})$. Denote the set of valuation functions, $v_i : 2^M \to \mathbb{R}^+$, as: $V = \{v_1, \ldots, v_n\}$. An allocation $A = \{A_1, \ldots, A_n\}$ is a partition of the items into n bundles (one for each agent). We define fair allocations in terms of maximin shares. Agent i's maximin share (μ_i) is the maximum value she can guarantee herself if she is allowed to choose the allocation A, on the condition that she receives her least preferred bundle. Formally, let A be an allocation and $\mathcal{A} = \left\{A = \{A_1, \ldots, A_n\} : \ A_i \cap A_j = \emptyset, \forall i, j; \ \cup_k A_k = M\right\}$ be the set of all feasible allocations. Agent i's maximin share is:

$$\mu_i = \max_{A \in \mathcal{A}} \min_{A_k \in A} v_i(A_k). \tag{1}$$

We say an allocation A is MMS if each agent i receives a bundle A_i worth at least her maximin share: $v_i(A_i) \geq \mu_i$. An allocation is α approximate MMS (or simplify α-MMS) if each agent i receives a bundle A_i worth at least: $v_i(A_i) \geq \alpha\mu_i$, for some $\alpha \in (0, 1)$.

2.1 Properties of Maximin Share

Our approximation algorithm exploits a few key properties of maximin shares. We note that these are standard results which appear in [1, 6]. We include proofs for sake of completeness.

▶ **Proposition 1** (Scale Invariance). *Let $A = \{A_1, \ldots, A_n\}$ be an α-MMS allocation for the problem instance $I = (N, M, V)$ with additive valuations. For any agent $i \in N$ and any $c \in \mathbb{R}^+$, if we create an alternate instance $I' = (N, M, V')$ where i's valuations are scaled by c, i.e., $v'_{ij} := cv_{ij}, \forall j \in M$, then A is still an α-MMS allocation for (N, M, V').*

Proof. Let μ_i and μ'_i be agent i's MMS in instance I and I' respectively. For any bundle $S \subseteq M$, we have $v'_i(S) = cv_i(S)$. Therefore, $\mu'_i = c\mu_i$. Let $A = \{A_1, \ldots, A_n\}$ be the allocation i selects to create her μ_i. Then, $v'_i(A_k) = cv_i(A_k) \geq c\alpha\mu_i = \alpha\mu'_i, \forall k$. ◀

▶ **Proposition 2** (Normalized Valuation). *For problem instance $I = (N, M, V)$, if agent i's valuation function satisfies:*

$$v_i(M) = \sum_{j \in M} v_{ij} = |N|, \tag{2}$$

then $\mu_i \leq 1$.

Proof. For contradiction, suppose $v_i(M) = |N|$ but $\mu_i > 1$. Let $A = \{A_1, \ldots, A_n\}$ be the allocation i selects to create her μ_i. From the definition of μ_i (1), $v_i(A_k) \geq \mu_i \ \forall A_k \in A$, so $|N| = v_i(M) = \sum_k v_i(A_k) \geq |N|\mu_i > |N|$, a contradiction. ◀

We say agent i's valuation is normalized for $I = (N, M, V)$ when (2) holds, or simply normalized when the underlying problem instance is clear. In view of Proposition 1, normalizing agents' valuations provides a convenient upper bound on μ_i's without affecting performance guarantees. In addition, this removes the problem of comparing the relative value of a bundle of items between agents whose scale of valuations differs in orders of magnitude.

2.2 What Makes Finding Approximate MMS Allocations Hard?

In this section, we build intuition for what exactly makes finding α approximate MMS allocations difficult. We begin with a definition. Let $I = (N, M, V)$ be a problem instance, and $L \subset N$ be subset of agents and $S \subset M$ be subset of items. We say

$$I' = (N', M', V') = (N \setminus L, M \setminus S, V), \tag{3}$$

is a reduced instance of I. In words, we create a reduced instance by removing some subset of agents, and some subset of items. We call the agents $N' = N \setminus L$ the remaining agents of the reduced instance I'. The following simple observation plays an important role in finding approximate MMS allocations.

▶ **Proposition 3.** *Let $I = (N, M, V)$ be a problem instance, and let μ_i be the MMS for agent $i \in N$. If we remove one agent $k \in N$ and one item $j \in M$, then the MMS of all remaining agents in the reduced instance $I' = (N \setminus \{k\}, M \setminus \{j\}, V)$, is at least as large as their MMS in I, i.e., $\mu'_i \geq \mu_i$. In words, removing one agent and one item from a problem instance does not reduce the MMS guarantees for any remaining agent in the reduced instance.*

Proof. Suppose agent $k \in N$ and item $j \in M$ are removed from the instance, and let i be any remaining agent. Consider the allocation $A = \{A_1, \ldots, A_n\}$ she makes to calculate her MMS in the original instance I, and note that $A_l \geq \mu_i$ for all $A_l \in A$ by the definition of MMS. In the reduced instance $I' = (N \setminus \{k\}, M \setminus \{j\}, V)$, agent i needs to make one less bundle but has one less item. Let $A_l \in A$ be the bundle containing the removed item j. Suppose she simply takes the items of $A_l \setminus \{j\}$ and distributes them arbitrarily to the other bundles of A to create a new allocation $\hat{A} = \{\hat{A}_1, \ldots, \hat{A}_{n-1}\}$. Clearly, $\hat{A}$ is a feasible allocation, and $\hat{A}_l \geq \mu_i$ for all $\hat{A}_l \in \hat{A}$. Therefore, $\mu'_i \geq \mu_i$. ◀

Proposition 3 shows that removing one agent and one item does not reduce MMS guarantees for remaining agents in the reduced instance. It is straightforward to generalize the above argument to show that removing k agents and k items does not decrease MMS guarantees in the reduced instance.

▶ **Corollary 4.** *Let $L \subset N$ and $S \subset M$, and μ_i be the MMS for each agent i in an instance $I = (N, M, V)$. If $|L| = |S|$, then the MMS μ'_i of any remaining agent in the reduced instance $I' = (N \setminus L, M \setminus S, V)$ is at least as large as in the original instance, i.e., $\mu'_i \geq \mu_i$.*

Combining Proposition 3 with the simple greedy allocation method *bag filling*, yields an almost trivial 1/2-MMS allocation algorithm. Suppose we seek an α-MMS allocation. The bag filling algorithm is as follows: We add one, arbitrary item at a time to a bag S until an agent k values the bag at least $\alpha\mu_k$, i.e., $v_k(S) \geq \alpha\mu_k$. If another agent k' values the bag at least $\alpha\mu'_k$, then pick one arbitrarily. We assign k the bag S, and remove agent k and the items of S from the instance. We show that bag filling provides an efficient way to allocate low value items. We note that a similar result also appears in [6].

▶ **Proposition 5.** *Assume agents' valuations are normalized as defined in* (2)*, and that no agent values any item more than $0 < \delta < 1/2$: $v_{ij} \leq \delta$ for all $j \in M$, for all $i \in N$. Then, the bag filling algorithm gives a $(1 - \delta)$ MMS allocation.*

Proof. By the definition of normalized valuations and Proposition 2, we have $v_i(M) = |N|$ and $\mu_i \leq 1$ for all $i \in N$. Therefore, it is enough to show each agent receives a bag worth at least $1 - \delta$. Clearly, the agent that receives the bag in each iteration gets at least $1 - \delta$, so the claim amounts to showing remaining agents do not lose too much value when the

bag is assigned. Let j be the last item added to the bag S. Note that before adding j, all agents valued S less than $1-\delta$, i.e., $v_i(S \setminus j) < 1-\delta$ for all $i \in N$. Now, since valuations are additive and $v_{ij} < \delta$ for all agents $i \in N$, we have $v_i(S) \leq 1$. That is, no agent values the bag S more than 1. This means after removing agent k and the items of S, all remaining agents satisfy $v_i(M \setminus S) = v_i(M) - v_i(S) \geq |N| - 1$. Since this condition is an invariant of bag filling algorithm, all agents get at least $(1-\delta)$ of their maximin share. ◀

Combining Propositions 2, 3, and 5 yields a simple greedy algorithm to compute a 1/2 MMS allocation. We start by normalizing valuations as defined in (2). By Proposition 2, this ensures $\mu_i \leq 1$ for all agents $i \in N$. If some agent, say k, has valuation $v_{kj} \geq 1/2$ for some item j, then we assign item j to agent k. If more than one agent satisfies this condition, then pick one arbitrarily. By Proposition 3, the MMS μ'_i of agents in the reduced instance $I' = (N \setminus k, M \setminus j, V)$ is at least as large as their MMS μ_i in the original instance. Next we normalize valuations for the reduced instance I', and repeat the process, greedily assigning one item at a time to any agent who values the item at least 1/2 and then normalizing valuations, until either all agents are removed or $v_{ij} < 1/2$, $\forall j \in M$, $\forall i \in N$. In the later case, Proposition 5 shows that the bag filling algorithm provides all remaining agents a bundle worth at least 1/2 of their maximin share.

A natural approach to extending the above algorithm to give a 2/3 MMS allocation requires splitting the items M into three sets based on their value: high valued items for which $v_{ij} \geq 2/3$ for some agent $i \in N$, low valued items for which $v_{ij} < 1/3$ for all agents $i \in N$, and medium valued items for which $v_{ij} \geq 1/3$ for at least one agent $i \in N$ but $v_{ij} < 2/3$ for all $i \in N$. Notice that, similar to the 1/2 MMS algorithm above, we may greedily assign high valued items to give at least 2/3 MMS to the agent receiving the item without decreasing the MMS of any remaining agent in the reduced instance. Also, if all items are low valued $v_{ij} < 1/3$ $\forall i \in N$, then the bag filling algorithm easily yields a 2/3 MMS allocation. The real challenge lies in managing the medium valued items. These items are not valuable enough individually to satisfy $2/3\mu_i$ for an agent i, yet they are too valuable, to some agent, to distribute haphazardly through bag filling. Thus, we seek a simple, efficient means to allocate medium valued items.

2.3 Results of Bouveret and Lemaître, and Barman and Murthy

Our algorithm relies on some results of [4, 2] to obtain the means to properly manage 'medium valued' items as defined at the end of the last section. We start with a definition. A problem instance $I = (N, M, V)$ is ordered if:

$$v_{i1} \geq v_{i2} \geq \cdots \geq v_{im}, \quad \forall i \in N. \tag{4}$$

In words, in an ordered instance all agents have the same order of preference over items. Roughly speaking, this maximizes the competition between agents, and, intuitively, should make it more difficult to provide an MMS allocation. Indeed, Bouveret and Lemaître [4] show ordered instances are worst case. Further, they provide a reduction from any arbitrary instance $I = (N, M, V)$ to an ordered instance $I' = (N, M, V')$, and show that if A' is an MMS allocation for I', then one can find an MMS allocation A for I in polynomial time. Barman and Murthy [2] generalize these results for α approximate MMS allocations.

▶ **Proposition 6** (Section 2.1 of Barman and Murthy [2]). *Given any instance $I = (N, M, V)$, one can find an ordered instance $I' = (N, M, V')$ in polynomial time.*

Algorithm 1: Converting to an Ordered Instance.

Input : Original Instance (N, M, V)
Output : V': Valuations for Ordered Instance

```
1 for j = 1 to m do
2     for i = 1 to n do
3         j* = agent i's jth most valuable item ;
4         v'_ij ← v_i(j*) ;
```

Algorithm 2: α-MMS Allocation for Unordered Instance.

Input : Allocation $A' = (A'_1, \ldots, A'_n)$ for Ordered Normalized Instance $I' = (N, M, V')$ such that $v'_i(A'_i) \geq \alpha$ for all $i \in N$.
Output : Allocation $A = (A_1, \ldots, A_n)$ for Original Normalized Instance $I = (N, M, V)$ such that $v_i(A_i) \geq \alpha$ for all $i \in N$.

```
1 A = (∅, ..., ∅) and R ← M ;
2 for j = 1 to m do
3     a ← i : j ∈ A'_i (pick the agent assigned item j in A') ;
4     g ← arg max_{k∈R} v_ak;
5     A_i ← A_i ∪ {g} and R ← M \ {g};
```

Algorithm 1 gives explicit details for the process of converting any instance I into an ordered instance I'. We call I' constructed this way the ordered instance of I.

▶ **Theorem 7** (Theorem 2 and Corollary 1 of Barman and Murthy [2]). *For any instance I, let I' be its ordered instance. If A' is an α approximate MMS allocation for I', then using A' we can find allocation A which is an α approximate MMS allocation for I in polynomial time.*

Algorithm 2 shows how to obtain an α approximate MMS allocation A for the original instance I given an α approximate MMS allocation for the ordered instance I'. For the sake of completeness, we provide a brief proof of Theorem 7.

Proof. (Theorem 7) Clearly, both Algorithms 1 and 2 run in polynomial time. Notice that Algorithm 2 allocates each item $j \in M$ to at most one agent $i \in N$ and that one item is allocated in each iteration. Let k_j be the item allocated in the jth iteration of Algorithm 2, lines 2 through 5. Consider the agent i assigned $j \in A'_i$, meaning that $k_j \in A_i$. At the beginning of the jth iteration, exactly $j-1$ items have been allocated. Therefore, k_j is among the top j most valuable items for agent i. From the construction of the ordered instance I', it follows that for all $j \in A'_i$, $v_i(k_j) \geq v'_i(j)$. Therefore, $v_i(A_i) = \sum_{j \in A'_i} v_i(k_j) \geq \sum_{j \in A'_i} v'_i(j) = v'_i(A'_i) \geq \alpha$. ◀

Proposition 6 and Theorem 7 show that it suffices to consider ordered instances. A total ordering over the set of items M provides precious information to us as algorithm designers since we know precisely which items are best (favored by all agents). In other words, the ordering over M essentially means all items fall into three categories: low, medium, and high valued, corresponding to low, medium, and high in the ordering respectively. In Section 3, we show that a total ordering over the items M allows for a simple generalization of the 1/2 MMS allocation algorithm described in Section 2.2 to give 2/3 MMS allocations.

3 A 2/3 MMS Approximation

In this section we present an algorithm to find 2/3 approximate MMS allocations. Our method involves a preprocessing step with Proposition 6 to ensure the instance is ordered. We show how to obtain a 2/3 MMS allocation for the ordered instance, then use a post processing step with Theorem 7 to obtain a 2/3 MMS allocation for the original instance. From this point on, we assume the instance is ordered as defined in (4).

The algorithm builds one bundle of items at a time, assigns it to some agent i who values it at least $2/3\mu_i$, and then removes that agent and the bundle from the instance. The basic structure of the algorithm closely resembles the simple 1/2 MMS algorithm discussed in Section 2.2. In fact, the same simple strategies guide the algorithm's design.

Assuming valuations are normalized as defined in (2), our algorithm handles allocation of items based on their value: low, medium or high. For this we use the clustering approach of [6], and define the following sets of items:

$$\begin{aligned} S_H &= \{j \in M : \ \exists i \in N \text{ s.t. } v_{ij} \geq 2/3\} \\ S_M &= \{j \in M : \ \exists i \in N \text{ s.t. } 1/3 \leq v_{ij}, \ v_{ij} < 2/3, \ \forall i \in N\} \\ S_L &= \{j \in M : \ v_{ij} < 1/3, \ \forall i \in N\}, \end{aligned} \tag{5}$$

which correspond to high, medium, and low valued items respectively. Second, for any bundle $S \subseteq M$, we define the set $N(S)$ as the agent's with value at least 2/3 for S:

$$N(S) = \{i : \ i \in N, \ v_i(S) \geq 2/3\}. \tag{6}$$

By using the preprocessing step of Proposition 6, we ensure a total ordering on the items (4). Thus, for any agent i if $v_{ik} > 1/3$ for some item k, then $v_{ij} > 1/3$ for all $j \leq k$. Similarly, if $v_{ik} < 2/3$, then $v_{ij} < 2/3$ for all $j \geq k$.

3.1 2/3 MMS Algorithm

At a high level, our algorithm mirrors the simple 1/2 MMS algorithm, consisting of two phases: matching and bag filling. Like the 1/2 MMS algorithm, we allocate high value items S_H through a maximum matching, and assign all low value items S_L through bag filling. For medium valued items S_M, the total ordering on the items simplifies allocation decisions based on $|S_M|$. If $|S_M|$ is sufficiently large, we greedily assign a bundle containing the two 'least valuable' items of $|S_M|$ to any agent that values it at least 2/3, using a generalization of Proposition 3. Otherwise, we use a modified version of the bag filling algorithm. We make the treatment of medium valued items more precise shortly, but note that, the total ordering of items allows for small adjustments to the matching and bag filling stages of the 1/2 MMS algorithm to improve the approximation guarantees to 2/3 MMS. Further, our approach makes the analysis of each stage nearly as simple as in the 1/2 MMS algorithm. We now explain the algorithm in more detail, see Algorithm 3 for a formal description.

Matching Procedure. The initial phase of the algorithm allocates high value items S_H through a maximum matching we call the Matching Procedure. First, we normalize valuations which ensures $\mu_i \leq 1$ for all agents by Proposition 2. Next, we form a bipartite graph with agents of on the left hand side and items of S_H on the right. We create an edge between agent i and item j, if $v_{ij} \geq 2/3$. In words, the graph's edges connect agents with items they value at least 2/3. Next, we solve a maximum matching T, and assign i bundle $A_i = j$, if $(i, j) \in T$. All matched items and agents are removed from the instance and we normalize valuations for the remaining agents. This process repeats until $|S_H| = 0$, i.e., there are no more high valued items.

Greedy Assignment from S_M. After completing the first phase, all high value items are allocated. Next, we determine how to distribute medium and low value items among the remaining agents. Our preprocessing step with Proposition 6 ensures the instance is ordered (4), meaning there is a least preferred item in any set of items (5). More precisely, $j^* = \arg\max_{j \in S_M} j$ is the least preferred item of S_M (medium value items). When $|S_M| > |N|$, each agent must create at least one bundle containing two or more items of S_M when calculating their μ_i, by pigeon hole principle. Similar to the matching stage of the 1/2 MMS algorithm, we greedily assign the two least preferred items of S_M, $S = \{j^* - 1, j^*\}$, to an arbitrary agent i with valuation $v_i(S) \geq 2/3$. This ensures the i receiving S gets at least $2/3\mu_i$, and the MMS of all remaining agents k in the reduced instance $I' = (N \setminus i, M \setminus S, V)$ satisfy: $\mu'_k \geq \mu_k$. Agent's valuations are then normalized, and process repeats.

Modified Bag Filling. After allocating the bulk of medium value items $|S_M| \leq |N|$, we create bundles for the remaining agents through a slightly modified version of bag filling. Here, we simply initialize the bag S using one, arbitrary item from S_M, and then fill the bag with items from S_L (low valued items) until some agent i values S at least $v_i(S) \geq 2/3$. Once $|S_M| = 0$, we use the standard bag filling algorithm.

Recall that the challenge of improving the approximation guarantees of the 1/2 MMS algorithm requires proper management of the medium valued items S_M. Our approach, using Proposition 6 and Theorem 7 to ensure the instance is ordered, enables a simple and natural extension of the straight-forward 1/2 MMS algorithm to provide improved 2/3 MMS guarantees.

The algorithm consists of two phases, matching and bag filling. The phases use different allocation procedures based on $|S_H|$ and $|S_M|$ respectively. We consider these procedures separately, starting with the matching procedure.

▶ **Lemma 8.** *Let $I = (N, M, V)$ be a problem instance where agents' valuations are normalized as defined in (2), and let μ_i be agent i's MMS. Suppose $|S_H| > 0$, as defined in (5), and that the Matching Procedure is used to create a maximum matching T. Let L be the agents of T and S be the items of T. Then,*

- **(i)** *$|L| = |S| > 0$.*
- **(ii)** *All removed agents i receive at least $2/3\mu_i$.*
- **(iii)** *Let μ'_i be the MMS of remaining agent i in the reduced instance $I' = (N \setminus L, M \setminus S, V)$. Then, $\mu'_i \geq \mu_i$.*

Proof. Recall the Matching Procedure creates a bipartite graph $G = (V, E)$ where the vertices V consist of agents on the left side and items on the right. An edge $e \in E$ is created between agent i and item j if $v_{ij} \geq 2/3$. Finally, a maximum matching T is determined. By definition of S_H and the fact $|S_H| > 0$, the set of edges E of G is non-empty. Since T is a maximum matching, part i) is obvious. Next, recall that Proposition 2 shows that $\mu_i \leq 1$ for all $i \in N$ since valuations are normalized. Part ii) then follows by the construction of G. Finally, since $|L| = |S|$, Corollary 4 guarantees $\mu'_i \geq \mu_i$ for all remaining agents i. ◀

We now consider the second procedure of the algorithm's matching phase.

▶ **Lemma 9.** *Let $I = (N, M, V)$ be an ordered problem instance with normalized valuations, and let μ_i be agent i's MMS. Suppose that $|S_H| = 0$ and $|S_M| > |N|$. Define $j^* = \arg\max_{j \in S_M} j$, and let $S = \{j^*, j^* - 1\}$ be the two least preferred items of S_M. Suppose bundle S is assigned an arbitrary agent k satisfying $v_k(S) \geq 2/3$. Then,*

- **(i)** *$v_k(S) \geq 2/3\mu_k$.*
- **(ii)** *Let μ'_i be the MMS of any remaining agent i in the reduced instance $I' = (N \setminus k, M \setminus S, V)$. Then, $\mu'_i \geq \mu_i$.*

Algorithm 3: 2/3-MMS Allocation.

Input : Ordered Instance $\langle N, M, V \rangle$
Output : 2/3 Approximate Maximin Share Allocation

while $|N| > 0$ **do**
  Normalize Valuations ;
  if $|S_H| > 0$ **then**
    Matching Procedure ;
  else if $|S_M| > |N|$ **then**
    $j^* \leftarrow \max_{j \in S_M} j$; // lowest value item of S_M
    $N(j^*) \leftarrow \{i : \ i \in N, \ v_i(j^*, j^* - 1) \geq 2/3\}$;
    $i \in N(j^*)$; $A_i \leftarrow \{j^*, j^* - 1\}$; // assign i the bundle $\{j^*, j^* - 1\}$
    $N \leftarrow N \setminus i$; $M \leftarrow M \setminus \{j^*, j^* - 1\}$;
  else
    while $|N| \geq |S_M|$ **do**
      if $|S_M| > 0$ **then**
        $S \leftarrow j \in S_M$; // create a bag with arbitrary item of S_M
      else
        $S \leftarrow j \in S_L$; // create a bag with arbitrary item of S_L
      $N(S) = \{i : i \in N, v_i(S) \geq 2/3\}$; // $N(S)$ changes with S
      while $|N(S)| = 0$ **do**
        $j \in S_L$; $S \leftarrow S \cup j$; // add arbitrary low value item to the bag
      $i \in N(S)$; $A_i \leftarrow S$; // assign i the bundle S
      $N \leftarrow N \setminus i$; $M \leftarrow M \setminus S$;

Proof. The argument is a simple generalization of Proposition 3. From normalized valuations and Proposition 2, $\mu_i \leq 1$ for all $i \in N$. By definition of the set S_M, for all items $j \in S_M$ there exists an agent $k \in N$ so that $v_{kj} \geq 1/3$. Since the instance is ordered, if $v_{kj} \geq 1/3$, then $v_{kj'} \geq 1/3$ for all $j' \leq j$. Since $|S_M| > |N| > 0$ and $j^* \in S_M$, there exists at least one agent $k \in N$ so that $v_k(S) \geq 2/3\mu_k$, showing part i).

We now show part ii). For any remaining agent i in the reduced instance I', consider the bundles $A = \{A_1, \ldots, A_n\}$ she makes while computing her μ_i in the original instance I. Note that $v_i(A_j) \geq \mu_i$ for all $A_j \in A$. In the reduced instance I', agent i must create one less bundle, but has two fewer items, specifically j^* and $j^* - 1$. We show how to construct a feasible allocation $A' = \{A'_1, \ldots, A'_{n-1}\}$ so that $v_i(A'_j) \geq \mu_i$ for all $A'_j \in A'$. Notice that the condition $|S_M| > |N|$ guarantees that at least one bundle, say A_k, must contain at least two items, say $u, v \in S_M$, by the pigeon hole principle. Wlog we may assume $v_i(u) \leq v_i(v)$. Since we take the two lowest valued items of S_M, $S = \{j^*, j^* - 1\}$, then $v_i(j^*) \leq v_i(u)$ and $v_i(j^* - 1) \leq v_i(v)$. Let A_{j^*} and A_{j^*-1} be the bundles of A containing items j^* and $j^* - 1$ respectively. Suppose agent i swaps item $u \in A_k$ with item $j^* \in A_{j^*}$ and swaps item $v \in A_k$ with item $j^* - 1 \in A_{j^*-1}$ to create A'_k, A'_{j^*}, and A'_{j^*-1}. Finally, i distributes the items of $A'_k \setminus S$ to other bundles arbitrarily to create a new set of bundles $A' = \{A'_1, \ldots, A'_{n-1}\}$. It is clear that A' is a feasible allocation and that $v_i(A'_j) \geq \mu_i$. Therefore, agent i's MMS μ'_i in the reduced instance I' satisfies $\mu'_i \geq \mu_i$. ◀

We now consider the algorithm's second phase, bag filling.

▶ **Lemma 10.** *Let $I = (N, M, V)$ be an ordered problem instance with normalized valuations. Suppose that $|S_H| = 0$ and $0 < |S_M| \leq |N|$. Then, the modified bag filling algorithm ensures all agents receive a bundle worth at least $2/3$ of their maximin share.*

Proof. This argument is a simple generalization of Proposition 5. In modified bag filling, we simply initialize the bag S to an arbitrary item $j \in S_M$. Notice that, this initialization ensures the condition $|S_M| \leq |N|$ holds in each iteration since we always remove one agent and one item of S_M. As valuations are normalized, it is enough to show all agents receive a bundle worth at least $2/3$.

First, note that $|S_L| > 0$, since from normalized valuations and the fact that $v_{ij} < 2/3$ $\forall j \in S_M$, we see that $\forall i \in N$: $v_i(S_L) = v_i(M) - v_i(S_M) \geq |N| - 2/3|S_M| \geq |N|/3 > 0$. We now show that some agent i eventually values the bag $v_i(S) \geq 2/3$. Let $j \in S_M$ be the initial item of the bag. If there exists an agent $i \in N$ such that $v_{ik} < 1/3$ for all $k \in M$, then, clearly there exists some $S' \subset S_L$ so that $v_i(j \cup S') \geq 2/3$. Suppose that no such agent exists. Note that from the definition of S_M, there exists some agent i such that $v_{ij} \geq 1/3$. Given that $v_i(S_L) \geq |N|/3$, we see that $v_i(j \cup S_L) \geq 1/3 + |N|/3 \geq 2/3$ for $|N| \geq 1$. Therefore, there exists $S' \subset S_L$ so that $v_i(j \cup S') \geq 2/3$. This establishes the bag is eventually assigned to an agent who values it at least $2/3$.

Let k be the agent assigned the bag S. Now, we show that $v_i(S) \leq 1$ for all other agents $i \in N \setminus k$. Before adding the final item of the bag $j' \in S$, $v_i(S \setminus j') < 2/3$ for all $i \in N$. The final item added to the bag comes from S_L so $v_i(j') < 1/3$ for all $i \in N$. Therefore, $v_i(S) < 1$ for all $i \in N$. This means that for each agent i, $v_i(M) \geq |N|$ and $v_i(S_L) \geq |N|/3$ are invariants of the algorithm. Then, it is easy to see all agents receive a bundle worth $2/3$. Finally, when $|S_M| = 0$, all agents receive a bundle worth at least $2/3$ by Proposition 5. ◀

From Lemmas 8, 9, and 10, we get the following theorem.

▶ **Theorem 11.** *Algorithm 3 provides a 2/3 approximate MMS allocation.*

▶ Remark. Lemmas 9 and 10 are really just simple generalizations of Propositions 3 and 5 (respectively) designed to manage medium valued items S_M. In this sense, our algorithm is natural generalization of the simple $1/2$ MMS algorithm of Section 2.2 which improves performance guarantees to $2/3$ MMS.

4 Discussion

In this paper we investigate fair division of indivisible items using maximin share as our measure of fairness of an allocation. We propose a simple greedy approximation algorithm to obtain a $2/3$ MMS allocation. Further, we show that our algorithm can be seen as a natural extension of the $1/2$ MMS algorithm discussed in Section 2.2. This allows for a far simpler, and more intuitive analysis as compared to other existing $2/3$ MMS approximations.

Our approach does not seem to generalize to provide better performance guarantees. Consider designing an algorithm to give a $3/4$ MMS allocation. Suppose we naively create three clusters of items: high $v_{ij} \geq 3/4$ for at least one agent i, medium $v_{ij} \geq 1/4$ for at least one agent but $v_{ij} < 3/4$ for all agents, and low $v_{ij} < 1/4$ for all agents. Similar to the $2/3$ case, we allocate high valued items through maximum matching, and if all items are low valued, then bag filling suffices to distribute all remaining items. Notice that, we must assign two or three medium valued items to ensure an agent receives at bundle worth at least $3/4$. If $|S_M| > 2|N|$, then we can guarantee each agent must create at least one bundle containing three items from S_M when computing their MMS, and therefore, may justify

greedily assigning a bundle containing the three lowest valued items of S_M to any agent who values it at least 3/4. When $2|N| \geq |S_M| > |N|$, the situation is less clear. We know each agent creates at least one bundle containing two items from S_M when computing their MMS, but we can't guarantee that some agent will value a bundle containing only the two lowest valued items of S_M at least 3/4. Further, we can't guarantee that we may start bag filling where we initialize the bag to the two least valuable items of S_M since some agent might value some set of two 'better' (more valuable) items of S_M more than 1. If we initialize the bag to only the lowest valued item of S_M, then we might 'run' out of low valued items, leaving only medium valued items and no way to ensure each remaining agent receives at least 3/4.

Attempting a finer partitioning of S_M significantly complicates analysis as it creates numerous special cases based on the number of items within each sub-cluster of medium valued items. Further, it is not clear that a simple allocation decision exists for all possible special cases. For these reasons, it seems the approach presented in this paper is only capable of producing a 2/3 MMS allocation. However, as our algorithm is closely related to the simple 1/2 MMS approximation, we find our approach more intuitive than other existing 2/3 MMS algorithms.

References

1. Georgios Amanatidis, Evangelos Markakis, Afshin Nikzad, and Amin Saberi. Approximation algorithms for computing maximin share allocations. *ACM Transactions on Algorithms (TALG)*, 13(4):52, 2017.
2. Siddharth Barman and Sanath Kumar Krishna Murthy. Approximation algorithms for maximin fair division. In *Proceedings of the 2017 ACM Conference on Economics and Computation*, pages 647–664. ACM, 2017.
3. Sylvain Bouveret and Michel Lemaître. Characterizing conflicts in fair division of indivisible goods using a scale of criteria. *Autonomous Agents and Multi-Agent Systems*, 30(2):259–290, 2016.
4. Sylvain Bouveret and Michel Lemaître. Efficiency and sequenceability in fair division of indivisible goods with additive preferences. *arXiv preprint arXiv:1604.01734*, 2016.
5. Eric Budish. The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes. *Journal of Political Economy*, 119(6):1061–1103, 2011.
6. Mohammad Ghodsi, MohammadTaghi HajiAghayi, Masoud Seddighin, Saeed Seddighin, and Hadi Yami. Fair allocation of indivisible goods: Improvement and generalization. In *EC*, 2018.
7. David Kurokawa, Ariel D Procaccia, and Junxing Wang. When can the maximin share guarantee be guaranteed? In *AAAI*, volume 16, pages 523–529, 2016.
8. David Kurokawa, Ariel D. Procaccia, and Junxing Wang. Fair Enough: Guaranteeing Approximate Maximin Shares. *J. ACM*, 65(2):8:1–8:27, 2018.
9. Ariel D Procaccia and Junxing Wang. Fair enough: Guaranteeing approximate maximin shares. In *Proceedings of the fifteenth ACM conference on Economics and computation*, pages 675–692. ACM, 2014.
10. Hugo Steinhaus. The problem of fair division. *Econometrica*, 16:101–104, 1948.
11. Gerhard J Woeginger. A polynomial-time approximation scheme for maximizing the minimum machine completion time. *Operations Research Letters*, 20(4):149–154, 1997.