

An improved approximation algorithm for maximin shares ☆,☆☆



Jugal Garg^{*}, Setareh Taki

University of Illinois at Urbana-Champaign, United States of America

ARTICLE INFO

Article history:

Received 10 August 2020

Received in revised form 1 June 2021

Accepted 3 June 2021

Available online 8 June 2021

Keywords:

Fair division

Maximin shares

Strongly polynomial algorithm

ABSTRACT

Fair division is a fundamental problem in various multi-agent settings, where the goal is to divide a set of resources among agents in a *fair* manner. We study the case where m indivisible items need to be divided among n agents with additive valuations using the popular fairness notion of *maximin share* (MMS). An MMS allocation provides each agent a bundle worth at least her maximin share. While it is known that such an allocation need not exist [1,2], a series of remarkable work [1,3–6] provided approximation algorithms for a $\frac{2}{3}$ -MMS allocation in which each agent receives a bundle worth at least $\frac{2}{3}$ times her maximin share. More recently, Ghodsi et al. [7] showed the existence of a $\frac{3}{4}$ -MMS allocation and a PTAS to find a $(\frac{3}{4} - \epsilon)$ -MMS allocation for an $\epsilon > 0$. Most of the previous works utilize intricate algorithms and require agents' approximate MMS values, which are computationally expensive to obtain.

In this paper, we develop a new approach that gives a simple algorithm for showing the existence of a $\frac{3}{4}$ -MMS allocation. Furthermore, our approach is powerful enough to be easily extended in two directions: First, we get a strongly polynomial time algorithm to find a $\frac{3}{4}$ -MMS allocation, where we do not need to approximate the MMS values at all. Second, we show that there always exists a $(\frac{3}{4} + \frac{1}{12n})$ -MMS allocation, improving the best previous factor. This improves the approximation guarantee, most notably for small n . We note that $\frac{3}{4}$ was the best factor known for $n > 4$.

© 2021 Elsevier B.V. All rights reserved.

1. Introduction

Fair division is a fundamental problem in various multi-agent settings, where the goal is to divide a set of resources among agents in a *fair* manner. It has been a subject of intense study since the seminal work of Steinhaus [9] where he introduced the cake-cutting problem for $n > 2$ agents: Given a heterogeneous (divisible) cake and a set of agents with different valuation functions, the problem is to find a *fair* allocation. The two most well-studied notions of fairness are: 1) Envy-freeness, where each agent prefers her own share of cake over any other agents' share, and 2) Proportionality, where each agent receives a share that is worth at least $1/n$ of her value for the entire cake.

We study the *discrete* fair division problem where m *indivisible* items need to be divided among n agents with additive valuations. For this setting, no algorithm can provide either envy-freeness or proportionality, in general, e.g., consider

☆ This paper is a participant in the 2020 ACM Conference on Economics and Computation (EC) Forward-to-Journal Program.

☆☆ A two-page abstract of this work appeared in the proceedings of ACM EC'20 [8].

* Corresponding author.

E-mail addresses: jugal@illinois.edu (J. Garg), staki2@illinois.edu (S. Taki).

allocating a single item among $n > 1$ agents. This necessitated an alternate concept of fairness. Budish [10] introduced an intriguing option called *maximin share*, which has attracted a lot of attention [1,2,7,6,3,5,11,4]. The idea is a straightforward generalization of the popular *cut and choose protocol* in the cake-cutting problem and a natural relaxation of proportionality. Suppose we ask an agent i to partition the items into n bundles (one for each agent), with the condition that the other $n - 1$ agents get to choose a bundle before her. In the worst case, i receives the least preferred bundle. Clearly, in such a situation, the agent will choose a partition that maximizes the value of her least preferred bundle. This maximum possible value is called i 's maximin share (MMS) value. In fact, when all agents have the same valuations, i cannot guarantee more than the MMS value.

Each agent's MMS value is a specific objective that gives her an intuitive measure of the fairness of an allocation. For example, Gates et al. [12] showed that in real-life experiments maximin metric is preferred by participating agents over others. This raises a natural question: Is there an allocation where each agent receives a bundle worth at least her MMS value? An allocation satisfying this property is said to be maximin share allocation (MMS allocation), and if it exists, it provides strong fairness guarantees to each individual agent. However, Procaccia and Wang [1], through a clever counter-example, showed that MMS allocation might not exist but a $\frac{2}{3}$ -MMS allocation always exists, i.e., an allocation where each agent receives a bundle worth at least $\frac{2}{3}$ of their MMS value. Later, Ghodsi et al. [7] improved the factor by showing the existence of a $\frac{3}{4}$ -MMS allocation using a sophisticated technique with a very challenging analysis.

We note that these are primarily existential results that do not provide any efficient algorithm to find such an allocation. The main issue in these techniques is the need for agents' MMS values. The problem of finding the MMS value of an agent is NP-hard,¹ but a polynomial-time approximation scheme (PTAS) exists [13]. Theoretically, one can use PTAS to find a $(\frac{3}{4} - \epsilon)$ -MMS allocation for an $\epsilon > 0$ in polynomial time. However, for practical purposes, such algorithms are not very useful for small ϵ . Hence, finding an efficient algorithm to compute a $\frac{3}{4}$ -MMS allocation remained open.

1.1. Our results and techniques

In this paper, we develop a new approach that gives a simple algorithm for showing the existence of a $\frac{3}{4}$ -MMS allocation. Furthermore, our approach is powerful enough to be easily extended in two directions: First, we get a strongly polynomial time algorithm to find a $\frac{3}{4}$ -MMS allocation, where we do not need to use the PTAS in [13] to approximate the MMS values at all. Second, we show that there always exists a $(\frac{3}{4} + \frac{1}{12n})$ -MMS allocation, improving the best previous factor by Ghodsi et al. [7]. This improves the approximation guarantee, most notably for small n . We note that there are works, e.g., [4,7], exploring better approximation factors for a small number of agents, and $\frac{3}{4}$ was the best factor known for $n > 4$.

Our algorithms are extremely simple. We first describe the basic algorithm, given in Section 3, that shows the existence of a $\frac{3}{4}$ -MMS allocation. We assume that MMS values are known for all agents. Since the MMS problem is scale-invariant (shown in Lemma 2.4), we scale valuations to make each agent's MMS value 1. Then, we assign *high-value items* (e.g., a single item that some agent values at least $\frac{3}{4}$) to agents, who value them at least $\frac{3}{4}$, with a simple greedy approach based on the pigeonhole principle. We remove the assigned items and the agents receiving these items from further consideration. This reduces the number of high-value items to be at most $2n'$, where n' is the number of remaining agents. These greedy assignments massively simplify allocation of high-value items, which was the most challenging part of previous algorithms. Next, we prepare n' bags, one for each remaining agent, and put at most two high-value items in each bag. Then, we add low-value items on top of each of these bags one by one using a bag filling procedure until the value of bag for some agent is at least $\frac{3}{4}$. The main technical challenge here is to show that there are enough low-value items to give every agent a bag they value at least $\frac{3}{4}$.

In Section 4, we extend the basic algorithm to compute a $\frac{3}{4}$ -MMS allocation in strongly polynomial time without any need to compute the actual MMS values (using the PTAS in [13]). Here, we define a notion of *tentative assignments* and a novel way for updating the MMS upper bound. For each agent, we use the average value, that is the value of all items divided by the number of agents, as an upper bound of her MMS value. The only change from the basic algorithm is that some of the greedy assignments are tentative, i.e., they are valid only if the current upper bound of the MMS values is tight enough. We show that this can be checked by using the total valuation of low-value items. If the upper bounds are not tight enough for some agents, then we update the MMS value of such an agent and repeat. We show that we do not need to update the MMS upper bounds more than $O(n^3)$ times before we have a good upper bound on all MMS values. Then, we show that the same bag filling procedure, as in the basic algorithm, satisfy every remaining agent. The running time of the entire algorithm is $O(nm(n^4 + \log m))$.

In Section 5, we show that our basic algorithm also yields a better bound of the existence of a $(\frac{3}{4} + \frac{1}{12n})$ -MMS allocation. The entire algorithm remains exactly the same but with an involved analysis. The analysis is tricky in this case, so we add a set of *dummy items* to make proofs easier. We use these items to make up for the extra loss for the remaining agents due to the additional factor, and, of course, these items are not assigned to any agent in the algorithm.

¹ Observe that the partition problem reduces to the MMS value problem with $n = 2$.

1.2. Related work

Maximin share is a popular fairness notion of allocating indivisible items among agents. Bouveret and Lemaître [14] showed that an MMS allocation always exists in some restricted cases, e.g., when there are only two agents or if agents' valuations for items are either 0 or 1, but left the general case as an open problem. As mentioned earlier, Procaccia and Wang [1] showed that MMS allocation might not exist, but a $\frac{2}{3}$ -MMS allocation always exists. They also provided a polynomial time algorithm to find a $\frac{2}{3}$ -MMS allocation when the number of agents n is constant. For the special case of four agents, their algorithm finds a $\frac{3}{4}$ -MMS allocation. Amanatidis et al. [4] improved this result by addressing the requirement for a constant number of agents, obtaining a PTAS that finds a $(\frac{2}{3} - \epsilon)$ -MMS allocation for an arbitrary number of agents; see [3] for an alternate proof. In [4], they also showed that a $\frac{7}{8}$ MMS allocation always exists when there are three agents. This factor was later improved to $\frac{8}{9}$ in [15].

Taking a different approach, Barman and Krishnamurthy [5] obtained a greedy algorithm to find a $\frac{2}{3}$ -MMS allocation. While their algorithm is fairly simple, the analysis is not. More recently, Garg et al. [6] obtained a simple algorithm to find a $\frac{2}{3}$ -MMS allocation that also has a simple analysis.

Ghods et al. [7] improved these results by showing the existence of a $\frac{3}{4}$ -MMS allocation and a PTAS to find a $(\frac{3}{4} - \epsilon)$ MMS allocation.

Maximin share fairness has also been studied in many different setting, e.g., for asymmetric agents (i.e., agents with different entitlements) [11], for group fairness [16,17], beyond additive valuations [5,7,18], in matroids [15], with additional constraints [15,19], for agents with externalities [20,21], with graph constraints [22,23], for allocating chores [24,5,25], and with strategic agents [26–29].

2. The MMS problem and its properties

We consider the fair allocation of a set M of m indivisible items among a set N of n agents with additive valuations, using the popular notion of maximin share (MMS) as our measure of fairness. Let v_{ij} denote agent i 's value for item j , and i 's valuation of any bundle $S \subseteq M$ of items is given by $v_i(S) = \sum_{j \in S} v_{ij}$. Let $V = (v_1, \dots, v_n)$ denote the set of all valuation functions.

An agent's *MMS value* is defined as the maximum value she can guarantee herself if she is allowed to choose a partition of items into n bundles (one for each agent), on the condition that other agents choose their bundles from the partition before her. In the following definition we define it formally.

Definition 2.1 (*MMS value and MMS partition*). Let $\mathcal{I} = \langle N, M, V \rangle$ denote an instance of the fair division problem, and let $\Pi_n(M) = \{P = \{P_1, \dots, P_n\} \mid P_i \cap P_j = \emptyset, \forall i, j; \cup_k P_k = M\}$ be the set of all feasible partitions of M into n bundles (one for each agent). Agent i 's MMS value or $\mu_i^n(M)$ (or simply μ_i when n and M are clear from the context) is defined as

$$\mu_i^n(M) = \max_{P \in \Pi_n(M)} \min_{P_k \in P} v_i(P_k) .$$

We call a partition achieving μ_i , an *MMS partition* of agent i .

Further, let $\mathcal{P}_i^n(M)$ denote the set of partitions achieving $\mu_i^n(M)$, i.e.,

$$\mathcal{P}_i^n(M) = \{P \in \Pi_n(M) : \min_{P_k \in P} v_i(P_k) = \mu_i^n(M)\} .$$

In other words, $\mathcal{P}_i^n(M)$ is set of all MMS partitions of agent i for items in M when there are n agents.

Definition 2.2 (α -MMS allocation and MMS problem). We say an allocation $A = (A_1, \dots, A_n)$ is α -MMS, for $\alpha \in (0, 1]$, if each agent i receives a bundle A_i worth at least α times her MMS value, i.e., $v_i(A_i) \geq \alpha \cdot \mu_i, \forall i \in N$. An MMS allocation is simply 1-MMS allocation.

Given an instance $\mathcal{I} = \langle N, M, V \rangle$ and an approximation factor $\alpha \in (0, 1]$, the MMS problem is to find an α -MMS allocation.

2.1. Properties of maximin share

In this section, we state nice properties of maximin shares that our algorithm exploits. These are standard results appeared in [1,4,30,5–7]. For completeness, we include their proofs in Appendix A.

Lemma 2.3 (AVERAGE UPPER BOUNDS MMS). $\mu_i^n(M) \leq \frac{v_i(M)}{n}, \forall i \in N$.

Lemma 2.4 (SCALE INVARIANCE). Let $A = (A_1, \dots, A_n)$ be an α -MMS allocation for instance $\mathcal{I} = \langle N, M, V \rangle$. If we create an alternate instance $\mathcal{I}' = \langle N, M, V' \rangle$ where valuations of each agent i are scaled by $c_i > 0$, i.e., $v'_{ij} := c_i \cdot v_{ij}, \forall j \in M$, then $\mu'_i = c_i \cdot \mu_i$ and A is an α -MMS allocation for $\mathcal{I}'$.

2.1.1. Ordered instances

We say that an instance $\mathcal{I} = \langle N, M, V \rangle$ is ordered if:

$$v_{i1} \geq v_{i2} \geq \dots \geq v_{im}, \forall i \in N.$$

In words, in an ordered instance, all agents have the same order of preferences over items. Bouveret and Lemaître [30] showed that the ordered instances are the worst case. They provided a reduction from any arbitrary instance $\mathcal{I} = \langle N, M, V \rangle$ to an ordered instance $\mathcal{I}' = \langle N, M, V' \rangle$ with a simple polynomial-time procedure that converts an MMS allocation of $\mathcal{I}'$ into an MMS allocation of $\mathcal{I}$. Later, Barman and Krishnamurthy [5] generalized this result for α -MMS allocations. This property is used in [5,6] to find a $\frac{2}{3}$ -MMS allocation. Observe that the MMS values of an agent in $\mathcal{I}$ and $\mathcal{I}'$ are the same because, by Definition 2.1, it neither depends on the order of the items nor on other agents' valuations.

Lemma 2.5 (ORDERED INSTANCE [30,5]). Without loss of generality, we can assume that agents have the same order of preferences over the items, i.e., $v_{i1} \geq v_{i2} \geq \dots \geq v_{im}, \forall i \in N$.

2.1.2. Bag filling for low value items

Ghods et al. [7] showed that if we normalize the valuation of agents so that $\mu_i^n(M) = 1, \forall i \in N$ and $v_{ij} \leq \beta, \forall i \in N, j \in M$, then we can find a $(1 - \beta)$ -MMS allocation using the following simple *bag filling procedure*: All items are unallocated in the beginning. Start with an empty bag B and keep filling it with unallocated items until some agent i values B at least $(1 - \beta)$. Then, allocate B to i (choose i arbitrarily if there are multiple such agents). Note that any remaining agent values B at most one because before adding the last item to B everyone values it less than $(1 - \beta)$ and adding one item will not increase the value of B more than β . We repeat this process for the set of unallocated items and the set of agents who have not allocated any bundle yet. Since $v_i(M) \geq |N|, \forall i$ using Lemma 2.3, there are enough items to satisfy all the agents with a bag that they value $(1 - \beta)$.

In Sections 3.2 and 4.3, we design a more general bag filling procedure.

2.1.3. Reduction

A useful concept of *valid reduction* is used in [2–4,7,6]. From Definition 2.1, $\mu_i^k(S)$ denote the MMS value of agent i when S is the set of items that needs to be divided among k agents (including i). Recall that for the α -MMS allocation problem for instance $\mathcal{I} = \langle N, M, V \rangle$, we want to partition M into $|N|$ bundles $(A_1, \dots, A_{|N|})$ such that $v_i(A_i) \geq \alpha \cdot \mu_i^{|N|}(M), \forall i$.

Definition 2.6 (Valid reduction). For obtaining an α -MMS allocation, the act of removing a set $S \subseteq M$ of items and an agent i from M and N is called a *valid reduction* if

$$\begin{aligned} v_i(S) &\geq \alpha \cdot \mu_i^{|N|}(M) \\ \mu_{i'}^{|N|-1}(M \setminus S) &\geq \mu_{i'}^{|N|}(M), \quad \forall i' \in N \setminus \{i\}. \end{aligned} \quad (1)$$

In words, valid reduction is the process of reducing the size of the instance $\mathcal{I} = \langle N, M, V \rangle$ by assigning a set of items S to an agent i and getting a smaller instance $\mathcal{I}' = \langle N \setminus \{i\}, M \setminus S, V \setminus \{v_i\} \rangle$ while the two conditions in (1) is satisfied. Clearly, an α -MMS allocation for the smaller instance $\mathcal{I}'$ gives an α -MMS allocation for the original instance. In our algorithms, we use it to remove *high-value items* and get smaller instances.

3. Existence of $\frac{3}{4}$ -MMS allocation

In this section, we present a simple proof of the existence of a $\frac{3}{4}$ -MMS allocation for a given instance $\mathcal{I} = \langle N, M, V \rangle$. We assume that the MMS value μ_i of each agent i is given. Finding the exact μ_i is an NP-Hard problem, however a PTAS exists [13]. This implies a PTAS to compute a $(\frac{3}{4} - \epsilon)$ -MMS allocation for any $\epsilon > 0$. Using the properties stated in Section 2.1, we normalize valuations so that $\mu_i^n(M) = 1, \forall i$ (Lemma 2.4) and assume that $\mathcal{I}$ is an ordered instance, i.e., $v_{i1} \geq \dots \geq v_{im}, \forall i$ (Lemma 2.5). Our proof is algorithmic. Whenever we apply a valid reduction and more than one agent satisfies the conditions (1), we choose one arbitrarily.

For the ease of exposition, we abuse notation and use M and N to respectively denote the set of unallocated items and the set of agents who have not received any bundle yet. We also use $n := |N|$ and $m := |M|$. Further, we use j to denote the j^{th} highest value item in M .

The algorithm is given in Algorithm 1. We use $\alpha = \frac{3}{4}$ in this section. Later, in Section 5, we use the same algorithm for $\alpha = \frac{3}{4} + \frac{1}{12n}$. Algorithm 1 has two main parts: Initial Assignment and Bag Filling. We now describe each part separately in detail.

Algorithm 1: α -MMS allocation.

Input : Ordered Instance $\mathcal{I} = \langle N, M, V \rangle$, i.e., $v_{i1} \geq v_{i2} \geq \dots \geq v_{im}, \forall i \in N$ and α
Output: α -MMS Allocation

```

1 Normalize Valuations                                     // Scale valuations so that  $\mu_i^n(M) = 1, \forall i$ 
2  $(N, M, V) \leftarrow \text{Initial-Assignment}(N, M, V, \alpha)$       // Algorithm 2
3 Bag-Filling( $N, M, V, \alpha$ )                               // Algorithm 3

```

Algorithm 2: Initial-assignment.

Input : Ordered Instance $\mathcal{I} = \langle N, M, V \rangle$, where $\mu_i^n(M) = 1, \forall i \in N$, and an approximation factor α
Output: Reduced Instance

```

1 For any  $S \subseteq M$ , define  $\Gamma(S) := \{i \in N : v_i(S) \geq \alpha\}$ 
2  $n \leftarrow |N|$                                            //  $n$  changes with  $N$ 
3  $S_1 \leftarrow \{1\}; S_2 \leftarrow \{n, n+1\}; S_3 \leftarrow \{2n-1, 2n, 2n+1\}; S_4 \leftarrow \{1, 2n+1\}$  // bundles that can be assigned
4 while  $(\Gamma(S_1) \cup \Gamma(S_2) \cup \Gamma(S_3) \cup \Gamma(S_4)) \neq \emptyset$  do
5    $S \leftarrow$  the lowest index bundle in  $\{S_1, S_2, S_3, S_4\}$  for which  $\Gamma(S) \neq \emptyset$ 
6    $i \leftarrow$  an agent in  $\Gamma(S)$ 
7   Assign  $S$  to agent  $i$                                 // initial assignment
8    $M \leftarrow M \setminus S; N \leftarrow N \setminus \{i\}$ 
9 return  $\langle N, M, V \rangle$ 

```

3.1. Initial assignment

We first assign *high-value items* using Algorithm 2. We note that handling high-value items is the biggest challenge in the MMS problem, e.g., a major part of Ghodsi et al. algorithm [7] is devoted to allocating high-value items. In our algorithms, we make it a simple process of greedy assignment by leveraging the pigeonhole principle to make valid reductions. We define bundles $S_1 := \{1\}$, $S_2 := \{n, n+1\}$ ($\emptyset$, if $m \leq n$), $S_3 := \{2n-1, 2n, 2n+1\}$ ($\emptyset$, if $m \leq 2n$), and $S_4 := \{1, 2n+1\}$ ($\emptyset$, if $m \leq 2n$), where S_1 has the highest value item in M , S_2 has the n^{th} and $(n+1)^{\text{th}}$ highest valued items in M , and so on. We will show that allocating any of these bundles to an agent who values it at least $\frac{3}{4}$ is a valid reduction. Note that these bundles change after every valid reduction.

In Algorithm 2, we keep assigning the lowest index $S \in \{S_1, S_2, S_3, S_4\}$ to agent i , if any, for which $v_i(S) \geq \frac{3}{4}$. Then, we update M and N to reflect the current unallocated items and agents who have not been assigned with any bundle yet. The following lemma extends the ideas that appeared in [7,6].

Lemma 3.1. *Let S be the lowest index bundle in $S \in \{S_1, S_2, S_3, S_4\}$ for which $\Gamma(S) := \{i \in N : v_i(S) \geq \frac{3}{4}\}$ is non-empty. Then, removing S and agent i with $v_i(S) \geq \frac{3}{4}$ is a valid reduction.*

Proof. Clearly, $v_i(S) \geq \frac{3}{4}$. Therefore, we only need to show the second condition in Definition 2.6. We show this separately for each case of $S \in \{S_1, S_2, S_3, S_4\}$. Fix agent $i' \in N \setminus \{i\}$ and $P \in \mathcal{P}_{i'}^n(M)$ (Recall from Definition 2.1 that $\mathcal{P}_{i'}^n(M)$ denote the set of partitions achieving $\mu_{i'}^n(M)$). We show that after removing S , there exists a partition of $M \setminus S$ into $(n-1)$ bundles such that the value of each bundle is at least $\mu_{i'}^n(M)$, i.e., $\mu_{i'}^{n-1}(M \setminus S) \geq \mu_{i'}^n(M)$.

- $S = S_1$. Removal of one item from P affects exactly one bundle and each of the remaining $(n-1)$ bundles has value at least $\mu_{i'}^n(M)$. Therefore, $\mu_{i'}^{n-1}(M \setminus \{1\}) \geq \mu_{i'}^n(M), \forall i' \in N \setminus \{i\}$.
- $S = S_2$. In P , there exists a bundle with two items from $\{1, \dots, n+1\}$ (pigeonhole principle). Let T be a bundle in P that has two items from $\{1, \dots, n+1\}$. Let us exchange these items with items n and $n+1$ in other bundles and distribute any remaining items in T among other bundles arbitrarily. Clearly, the value of other bundles except T does not decrease, and hence $\mu_{i'}^{n-1}(M \setminus \{n, n+1\}) \geq \mu_{i'}^n(M), \forall i' \in N \setminus \{i\}$.
- $S = S_3$. Similar to the proof of Case ' $S = S_2$ '.
- $S = S_4$. In each iteration, the lowest index bundle from $\{S_1, S_2, S_3, S_4\}$ is picked. Therefore, S_4 is only picked when $v_i(S_1), v_i(S_3) < \frac{3}{4}$ for all $i \in N$ which implies that $v_{i'1} < \frac{3}{4}$ and $v_{i'(2n+1)} < \frac{1}{4}$ and hence $v_i(S_4) < 1$ for all $i \in N$. In P , if items 1 and $2n+1$ are in the same bundle, then clearly, removing S_4 and agent i is a valid reduction. For the other case, if 1 and $2n+1$ are in two different bundles, then we can make two new bundles, one with $\{1, 2n+1\}$ and another with all the remaining items of the two bundles. The value of the bundle without $\{1, 2n+1\}$ is at least $2\mu_i - 1 \geq \mu_i$ because $v_i(S_4) < 1$ for all $i \in N$ and $\mu_i \geq 1$. Hence, this is a valid reduction. $\square$

Lemma 3.1 implies the following corollary.

Corollary 3.2. *After the execution of Algorithm 2 is completed, $\mu_i \geq 1$ for all $i \in N$.*

Algorithm 3: Bag filling algorithm for α -MMS allocation.

Input : Reduced Instance $\mathcal{I} = \langle N, M, V \rangle$ with $\mu_i^n = 1 \ \forall i$ and an approximation factor α
Output: Allocation $A = (A_1, \dots, A_n)$ where $v_i(A_i) \geq \alpha$

```

1 Initialize Bags  $B = \{B_k\}_{k \in [n]}$  as in (3) // see Fig. 1
2  $R \leftarrow M \setminus J$ 
3 for  $k = 1$  to  $n$  do
4    $T \leftarrow B_k$ 
5   Define  $\Gamma(T) := \{i \in N : v_i(T) \geq \alpha\}$ 
6   while  $\Gamma(T) = \emptyset$  do
7      $j \in R$  // pick one low value item arbitrarily
8      $T \leftarrow T \cup \{j\}; R \leftarrow R \setminus \{j\}$  // add the new item to bag
9    $i \in \Gamma(T); A_i \leftarrow T; N \leftarrow N \setminus \{i\}$  // assign  $T$  to  $i \in \Gamma(T)$ 

```

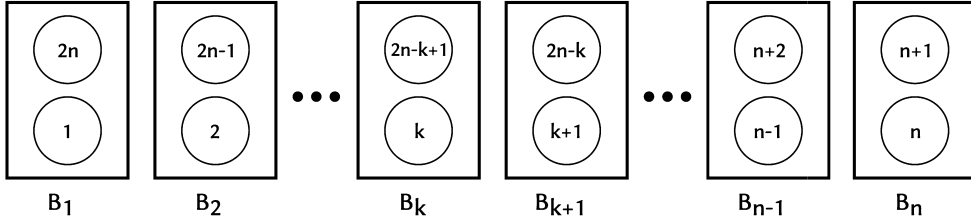


Fig. 1. Setting of the items in the bags.

Note that the original MMS value is 1 for each agent. Although it may increase after a valid reduction, we only need to give each agent a bundle of value at least $\frac{3}{4}$ to achieve a $\frac{3}{4}$ -MMS allocation for the original instance. For this reason, we are interested in partitions where the value of each bundle is at least 1 for an agent. Therefore, we abuse notation to denote by $\mathcal{P}_i^n(M)$ the set of partitions achieving the original MMS value, i.e.,

$$\mathcal{P}_i^n(M) = \{P \in \Pi_n(M) : \min_{P_k \in P} v_i(P_k) \geq 1\} . \quad (2)$$

Observe that $\mathcal{P}_i^n(M)$ contains MMS partitions of agent i for the current M and n as well as the partitions that do not achieve the current μ_i , but the value of each bundle is at least 1.

3.2. Bag filling

We use the bag filling procedure given in Algorithm 3 to satisfy the remaining agents. Let $J_1 := \{1, \dots, n\}$ denote the set of first n items. Similarly, let us define $J_2 := \{n+1, \dots, 2n\}$ and $J := J_1 \cup J_2$. We call J to be the set of high-value items. The following corollary is straightforward.

Corollary 3.3. *If $v_i(S) < \frac{3}{4}$, for all i and for all $S \in \{S_1, S_2, S_3\}$, then (i) $v_{ij} < \frac{3}{4}, \forall j \in J_1$, (ii) $v_{ij} < \frac{3}{8}, \forall j \in J_2$, and $v_{in} < \frac{3}{4} - v_{i(n+1)}$, and (iii) $v_{ij} < \frac{1}{4}, \forall j \in M \setminus J$, for all i .*

Next, we initialize n bags as follows:

$$B = \{B_1, B_2, \dots, B_n\}, \text{ where } B_k = \{k, 2n - k + 1\}, \forall k . \quad (3)$$

Each bag contains one item from J_1 and one item from J_2 such that from B_1 to B_n value of items from J_1 decreases and value of items from J_2 increases (see Fig. 1 for an illustration).

Algorithm 3 has n rounds. In each round k , it starts a new bundle T with $T \leftarrow B_k$. If there is an agent who values T to be at least $\frac{3}{4}$, then assign T to such an agent. Otherwise, keep adding items from $M \setminus J$ to T one by one until an agent with no bundle assigned to her values T at least $\frac{3}{4}$. The algorithm allocates T to that agent, and if there are multiple such agents, it chooses one arbitrarily.

For correctness, we need to show that there are enough items in $M \setminus J$ to add on top of each bag in (3) so that each agent gets a bundle that they value at least $\frac{3}{4}$. For this, we first divide agents into two types:

$$N^1 := \{i \in N \mid v_i(B_k) \leq 1, \forall k\} \text{ and } N^2 := N \setminus N^1 .$$

If N^2 is empty, then it is easy to check that using Corollary 3.3(iii) and the ideas in Section 2.1.2 that Algorithm 3 gives each agent at least $\frac{3}{4}$. We need some more notation to show correctness when N^2 is not empty.

For an agent $i \in N^2$, define

$$\begin{aligned} L_i &:= \{B_k : v_i(B_k) < \frac{3}{4}\}; \quad l_i := |L_i| \\ H_i &:= \{B_k : v_i(B_k) > 1\}; \quad h_i := |H_i| \\ x_i &:= (\frac{3}{4})l_i - \sum_{k: B_k \in L_i} v_i(B_k). \end{aligned} \quad (4)$$

In words, L_i is the set of bags that i values strictly less than $\frac{3}{4}$, H_i is the set of bags that i values strictly more than 1, and l_i and h_i are the number of bags in L_i and H_i respectively. Further, x_i is the least total value needed to make all bundles in L_i at least $\frac{3}{4}$. Agents in N^2 have nice properties that we show in the following lemma.

Lemma 3.4. For an agent $i \in N^2$, (i) $l_i > 0$ and $h_i > 0$, (ii) $v_{i1} > \frac{5}{8}$, (iii) $v_i(B_k) < \frac{9}{8}$, $\forall k$, and (iv) $v_{ij} < \frac{1}{8}$, $\forall j \in M \setminus J$.

Proof. For the first part, if $h_i = 0$ then $i \in N^1$. Further, $l_i = 0$ implies that $v_i(B_n) \geq \frac{3}{4}$, which cannot be true after the execution of Algorithm 2 due to Line 4–8 of the algorithm.

For the second part, $v_{ij} < \frac{3}{8}$, $\forall j \in J_2$ due to Corollary 3.3. Since $h_i > 0$, there exists a $j \in J_1$ such that $v_{ij} > \frac{5}{8}$. Further, since item 1 is the highest value item for every agent, $v_{i1} > \frac{5}{8}$.

For the third part, each item of J_1 has value less than $\frac{3}{4}$ and each item of J_2 has value less than $\frac{3}{8}$ (Corollary 3.3) for any agent. Therefore, $v_i(B_k) < \frac{9}{8}$ for any bundle B_k .

For the fourth part, the value of item 1 for an agent $i \in N^2$ is more than $\frac{5}{8}$ and $v_i(S_4) < \frac{3}{4}$, hence $v_{ij} \leq v_{i(2n+1)} < \frac{1}{8}$, $\forall j \in M \setminus J$. $\square$

In the following lemma, we show that if the total value of the items in $M \setminus J$ for each agent i in N^2 is at least $x_i + \frac{l_i}{8} - \frac{1}{8}$, i.e.,

$$v_i(M \setminus J) \geq x_i + \frac{l_i}{8} - \frac{1}{8},$$

then the bag filling algorithm will assign every agent (in N^1 and N^2) a bundle with value at least $\frac{3}{4}$. In the rest of this section, we show that the bound on the value of $M \setminus J$ actually holds by using the fact that $\mu_i \geq 1$ for all $i \in N$.

Lemma 3.5. If $v_i(M \setminus J) \geq x_i + \frac{l_i}{8} - \frac{1}{8}$, $\forall i \in N^2$, then Algorithm 3 gives every agent a bundle that they value at least $\frac{3}{4}$.

Proof. This is proof by contradiction. Note that, in Algorithm 3, R is the set of unallocated items from $M \setminus J$ and T is the bag that is being filled at a time. Let $R^{(k)}$ and $T^{(k)}$ be respectively R in the beginning and T at the end of round k of the algorithm, i.e., $R^{(1)} = M \setminus J$ and $T^{(1)} \supseteq B_1$. For contradiction, suppose the algorithm stops at round t because there are not enough unallocated items in $R^{(t)}$ to satisfy any remaining agent i , i.e., $v_i(B_t \cup R^{(t)}) < \frac{3}{4}$.

If $i \in N^1$, each removed bundle in rounds $k \in [t-1]$, has value of at most 1 for agent i . Because, if $v_i(B_k) \geq \frac{3}{4}$ for $k \in [t-1]$, agent i is already interested in B_k ($i \in \Gamma(B_k)$) and the algorithm does not enter the while loop in Line 6. Therefore, if $v_i(B_k) \geq \frac{3}{4}$ no more item has been added to $T^{(k)} = B_k$. Also, if $v_i(B_k) < \frac{3}{4}$ for $k \in [t-1]$, before adding the last item (if any) to $T^{(k)}$, the value of $T^{(k)}$ is less than $\frac{3}{4}$ (otherwise, it would be out of loop and allocated to someone). Moreover, from Corollary 3.3, $v_{ij} < \frac{1}{4}$ for $j \in R^{(k)}$ (since $R^{(k)} \subseteq M \setminus J$). Therefore, at the end of the round k , $v_i(T^{(k)}) < 1$. Further, since $v_i(M) \geq n$ and $v_i(B_k) \leq 1$, $\forall k$, we have $v_i(B_t \cup R^{(t)}) \geq 1$, which is a contradiction.

If $i \in N^2$, then since at round t , $v_i(T^{(t)}) < \frac{3}{4}$, we have $B_t \in L_i$ (See (4) for the definition of L_i). Consider a round $k \in [t-1]$. If $B_k \notin L_i$, then $T^{(k)} = B_k$ has been assigned to someone with no additional items added to $T^{(k)}$ from $R^{(k)}$ because $i \in \Gamma(T)$ and the algorithm does not enter the while loop in Line 6. If $B_k \in L_i$, then in round k , before adding the last item (if any) to $T^{(k)}$, the value of i for $T^{(k)}$ is less than $\frac{3}{4}$. Moreover, from Lemma 3.4, each item in $R^{(k)}$ has value of at most $\frac{1}{8}$. Therefore, if $B_k \in L_i$, the value of the assigned bag for i in round k is less than $\frac{7}{8}$. Since $B_t \in L_i$, at most $l_i - 1$ bags from L_i have been assigned up to $t-1$ iterations. Further, since items from $M \setminus J$ are added to bags in L_i only, the total value taken from $M \setminus J$ up to $t-1$ iterations, according to agent i , is at most $x_i - (\frac{3}{4} - B_t) + (l_i - 1)/8$ where $x_i - (3/4 - B_t)$ to make each of $L_i \setminus B_t$ exactly $\frac{3}{4}$ (See (4) for the definition of x_i) and $(l_i - 1)/8$ to add an extra $\frac{1}{8}$ to each. Hence, in the beginning of round t ,

$$v_i(R^{(t)}) \geq \left(x_i + \frac{l_i}{8} - \frac{1}{8}\right) - \left(x_i - (\frac{3}{4} - v_i(B_t)) + (l_i - 1)/8\right) = \frac{3}{4} - v_i(B_t), \quad (5)$$

which is a contradiction. $\square$

Now, we only need to show that for each $i \in N^2$, we have

$$v_i(M \setminus J) \geq x_i + \frac{l_i}{8} - \frac{1}{8}. \quad (6)$$

3.2.1. Showing (6)

In fact, we will show a stronger bound without $-\frac{1}{8}$ in Theorem 3.15. We will use the extra $-\frac{1}{8}$ to improve the bound in Section 5. We start with a few lemmas to show more properties of agents in N^2 , in addition to previous properties shown in Lemma 3.4. Recall from (2) that $\mathcal{P}_i^n(M)$ denote the set of partitions where the value of each bundle is at least 1.

Lemma 3.6. *For an agent $i \in N^2$, there exists a bundle P_k in every partition $P = \{P_1, \dots, P_n\} \in \mathcal{P}_i^n(M)$ such that $v_i(P_k \setminus J) > \frac{1}{4}$.*

Proof. If there exists a bundle P_k with exactly one item j from J , then $v_i(P_k \setminus J) \geq 1 - v_{ij} > \frac{1}{4}$ because the value of every item is less than $\frac{3}{4}$. Otherwise, each bundle has exactly two items from J , which implies that one of the bundles, say P_k , has two items j_1, j_2 from the set $\{n\} \cup J_2$. Since $v_{ij_1} + v_{ij_2} \leq v_{in} + v_{i(n+1)} < \frac{3}{4}$, $v_i(P_k \setminus J) > \frac{1}{4}$. $\square$

Next, we show that there exists a partition in $\mathcal{P}_i^n(M)$ for $i \in N^2$ where all items with value more than $\frac{5}{8}$ are in separate bundles. The intuition of this proof is that the bundle which has two items of value greater than $\frac{5}{8}$ can be merged with another bundle (or possibly two other bundles) and make two (or three) bundles each value at least 1. This basically utilizes the extra $\frac{2}{8}$ value in the first bundle to reshuffle items to obtain the desired partition. We begin with the following claim.

Claim 3.7. *If there exists a partition $P = \{P_1, \dots, P_n\} \in \mathcal{P}_i^n(M)$ for an agent $i \in N^2$ where a bundle $P_k \in P$ contains two items with value more than $\frac{5}{8}$ for agent i . Then,*

1. *there exists another bundle $P_{k'} \in P$ for which $\max_{j \in P_{k'}} v_{ij} < \frac{3}{8}$.*
2. *if $v_i(P_k \setminus J) > \frac{1}{4}$, then we can make two new bundles from the items in $P_k \cup P_{k'}$ where each bundle has one item with value more than $\frac{5}{8}$ and each bundle values at least 1.*
3. *if $P_{k'}$ cannot be divided into two parts with value at least $\frac{3}{8}$ each. Then, $v_i(P_{k'} \setminus J) < \frac{1}{4}$.*

Proof. Corollary 3.3 implies that the only items that can have value more than $\frac{5}{8}$ are items in $J_1 = \{1, \dots, n\}$. Therefore, P_k has two items from J_1 . By the pigeonhole principle, there must exist another bundle $P_{k'}$ where all items are from $\{n+1, \dots, m\}$. Corollary 3.3 implies that each item in $P_{k'}$ has value less than $\frac{3}{8}$. This proves the first part.

For the second part, if $v_i(P_k \setminus J) > \frac{1}{4}$ we have $v_i(P_k) > \frac{5}{8} + \frac{5}{8} + \frac{1}{4} = \frac{3}{2}$. We make two bundles by initializing two empty bags and adding items from $P_k \cup P_{k'}$ one by one in decreasing order of value to the bundle which has the lower value. We will get new bundles with values v_1 and v_2 where $v_1 + v_2 > \frac{5}{2}$ and $|v_1 - v_2| < \frac{3}{8}$. The second inequality holds because each item in $P_{k'}$ and $P_k \setminus J$ has value less than $\frac{3}{8}$. This proves the second part.

For the third part, assume $v_i(P_{k'} \setminus J) \geq \frac{1}{4}$ for a contradiction. We initialize two empty bundles and add items from $P_{k'}$ one by one in decreasing order of value to the bundle with lower value. We get two new bundles with value v_1 and v_2 where $v_1 + v_2 = v_i(P_{k'}) \geq 1$ and $|v_1 - v_2| < \frac{1}{8}$, which implies $v_1, v_2 \geq \frac{3}{8}$. The second inequality holds because each item in $P_{k'} \setminus J$ has value less than $\frac{1}{8}$. $\square$

Lemma 3.8. *For every $i \in N^2$, there exists a partition $P = \{P_1, \dots, P_n\} \in \mathcal{P}_i^n(M)$ such that each $P_k \in P$ has at most one item j with $v_{ij} > \frac{5}{8}$.*

Proof. If there are more than two items with value more than $\frac{5}{8}$ in a bundle P_k of $P \in \mathcal{P}_i^n(M)$, then we add one of these items to $P_{k'}$, defined in Claim 3.7. This will ensure that the value of both P_k and $P_{k'}$ is at least 1. By repeating this, we can obtain a $P \in \mathcal{P}_i^n(M)$ that has at most two items with value more than $\frac{5}{8}$.

Next, we show that if there are two items j_1, j_2 each with value more than $\frac{5}{8}$ for an agent i in a bundle P_k of $P \in \mathcal{P}_i^n(M)$, then we can construct another $P' \in \mathcal{P}_i^n(M)$ where this is not true. Let $P_{k'} \in P$ be a bundle for which $\max_{j \in P_{k'}} v_{ij} < \frac{3}{8}$ (see Claim 3.7(1) for the proof of its existence).

Case 1: If $v_i(P_k \setminus J) > \frac{1}{4}$, using Claim 3.7(2), we make two bundles with value at least 1 and exactly one item with value more than $\frac{5}{8}$ in each.

Case 2: If $v_i(P_k \setminus J) \leq \frac{1}{4}$ and there exists a partition $Q_{k'}^1$ and $Q_{k'}^2$ of items in $P_{k'}$ such that value of each $Q_{k'}^1$ and $Q_{k'}^2$ is at least $\frac{3}{8}$, then we can rearrange items in $P_k \cup P_{k'}$ and make two new bundles $(\{j_1\} \cup Q_{k'}^1)$ and $(\{j_2\} \cup Q_{k'}^2)$. Clearly, the value of each bundle is at least 1 and each has exactly one item with value more than $\frac{5}{8}$.

Case 3: Finally, if $v_i(P_k \setminus J) \leq \frac{1}{4}$ and no such $Q_{k'}^1$ and $Q_{k'}^2$ exists (as in Case 2), then we claim that there exists a partition of $P_{k'}$ into three bundles, each with value less than $\frac{3}{8}$. We can find this partition as follows: Initialize three empty bundles, and repeatedly add the highest value item of $P_{k'}$ to the bundle with the lowest value. For a contradiction, suppose one of the three bundles has value more than $\frac{3}{8}$, then the sum of the values of the other two sets must be less than $\frac{3}{8}$ because otherwise, they make a partition of two where each has value more than $\frac{3}{8}$. This means that at least one of the bundles

Table 1
Valuation of agent a for items in J in Example 3.10.

Item	1	2	3	4	5	6	7	8	9	10	11	12
v_{aj}	$\frac{28}{40}$	$\frac{28}{40}$	$\frac{28}{40}$	$\frac{28}{40}$	$\frac{17}{40}$	$\frac{15}{40}$	$\frac{14}{40}$	$\frac{14}{40}$	$\frac{14}{40}$	$\frac{13}{40}$	$\frac{5}{40}$	$\frac{1}{40}$

must have a value less than $\frac{3}{16}$. This implies that the value of the largest bag before adding the last item must be less than $\frac{3}{16}$ and the last item also should value less than $\frac{3}{16}$, which is a contradiction. Therefore, there exists a partition $Q_{k'}^1$, $Q_{k'}^2$ and $Q_{k'}^3$ of $P_{k'}$ such that each has value less than $\frac{3}{8}$.

According to Lemma 3.6, there exists a bundle $P_{\hat{k}} \in \mathcal{P}_i^n(M)$ such that $v_i(P_{\hat{k}} \setminus J) > \frac{1}{4}$. Let $v_i(P_{\hat{k}}) = 1 + \delta$ for some $\delta \geq 0$. $P_{\hat{k}}$ cannot be same as P_k (Case 1 of this proof) and $P_{k'}$ (Claim 3.7(3)). We initialize three bags:

$$\text{bag 1 : } \{j_1\} \cup Q_{k'}^1 \quad \text{bag 2 : } \{j_2\} \cup Q_{k'}^2 \quad \text{bag 3 : } (P_k \setminus \{j_1, j_2\}) \cup (P_{\hat{k}} \cap J) \cup Q_{k'}^3$$

Observe that the value of each of bag 1 and bag 2 is at most $\frac{9}{8}$, and the total value of all items in P_k , $P_{k'}$ and $P_{\hat{k}}$ is at least $3 + \frac{2}{8} + \delta$. We sort the remaining items in decreasing order and add them one by one to a bag with the lowest value. Since the value of the last item added is at most $\frac{1}{8}$ (Lemma 3.4), each bag has a value of at least 1 and it has at most one item with value more than $\frac{5}{8}$. We repeat this process for each bundle with two items of value more than $\frac{5}{8}$ to find a desired partition. $\square$

Let a be an agent in N^2 . For simplicity, until the end of this section, when we use *value* of an item or a bundle, we mean the value for agent a (unless mentioned otherwise). Recall that we need to show (6). Let $P = (P_1, \dots, P_n) \in \mathcal{P}_a^n(M)$ be a partition satisfying the condition of Lemma 3.8, i.e., there is at most one item with value greater than $\frac{5}{8}$ in P_k for all $k \in [n]$.

Using (3), we manipulate P as follows: First, for each $B_i = \{j, j'\}$ with $\frac{3}{4} \leq v_a(B_i) \leq 1$, if $j \in P_k$ and $j' \in P_{k'}$, then we turn P_k and $P_{k'}$ into two new bundles $\{j, j'\}$ and $((P_k \cup P_{k'}) \setminus \{j, j'\})$. Observe that $v_a((P_k \cup P_{k'}) \setminus \{j, j'\}) \geq 1$. Hence, we can assume that for each $B_i \notin L_a \cup H_a$ (as defined in (4)), there exists a $P_k = B_i$ and all other bundles value at least 1. Second, we re-enumerate the bundles in P such that

- $P_1, \dots, P_{t_1}$: each has an item j of value more than $\frac{5}{8}$ and $v_a(B_j) < \frac{3}{4}$.
- $P_{t_1+1}, \dots, P_{t_2}$: each has an item j of value more than $\frac{5}{8}$ and $v_a(B_j) > 1$ (Observe that there are h_a such bundles, so $t_2 - t_1 = h_a$).
- $P_{t_3+1}, \dots, P_n$: each such $P_k = B_{k'}$ for some k' and $\frac{3}{4} \leq v_a(B_{k'}) \leq 1$.
- $P_{t_2+1}, \dots, P_{t_3}$ be the remaining bundles (Observe that $t_3 = l_a + h_a$).

Let $L_a^1 \subseteq L_a$ be the set of bags in L_a , which have one item with value more than $\frac{5}{8}$, and $L_a^2 = L_a \setminus L_a^1$. Clearly, $|L_a^1| = t_1$ and $|L_a^2| = l_a - t_1$.

Observation 3.9.

1. If $v_a(B_j) > 1 \geq v_{aj} > \frac{5}{8}$, then by Corollary 3.3, $v_{a(2n-j+1)} > \frac{1}{4}$.
2. If $v_a(B_j) < \frac{3}{4}$ and $v_{aj} > \frac{5}{8}$, i.e., $B_j \in L_a^1$ then $v_{a(2n-j+1)} < \frac{1}{8}$.
3. Using the two previous observations, if $B_{j_1} \in L_a^1$ and $B_{j_2} \in H_a$, then $j_1 < j_2$ because $v_{a(2n-j_2+1)} > \frac{1}{4} > \frac{1}{8} > v_{a(2n-j_1+1)}$.
4. if $B_{j_1} \in H_a$ and $B_{j_2} \in L_a^2$, then $j_1 < j_2$ because $v_{aj_1} > \frac{5}{8} \geq v_{aj_2}$.
5. Bags are ordered by value of most-valued item in each bag. First, L_a^1 then H_a then L_a^2 (excluding the bags for which $\frac{3}{4} \leq v_a(B_k) \leq 1$).
6. The above observations imply that items with value in $[\frac{1}{8}, \frac{1}{4}]$ (if any) automatically belong to bags B_j 's, which are neither in H_a nor L_a . Hence, there is no such item in $\{P_1, \dots, P_{t_3}\}$.
7. After the termination of Algorithm 2 we have $v_i(S_2) = v_i(B_n) < \frac{3}{4}$. Therefore, $B_n \in L_a$. Further, since $h_a > 0$ (Lemma 3.4), $B_n \in L_a^2$. This implies that $|L_a^2| \geq 1$.

Further, observe that there is no item from $M \setminus J$ in $P_{t_3+1}, \dots, P_n$. We leverage this observation to prove (6) by only considering partitions $\{P_1, \dots, P_{t_3}\}$ that contain all items of $M \setminus J$. Here is a simple example to help understand the construction.

Example 3.10. Consider an example where, after the execution of Algorithm 2, $n = 6$ (hence $|J| = 12$). The valuation of agent a for items in J is shown in Table 1. The construction of bags is shown in Fig. 2 and Table 2.

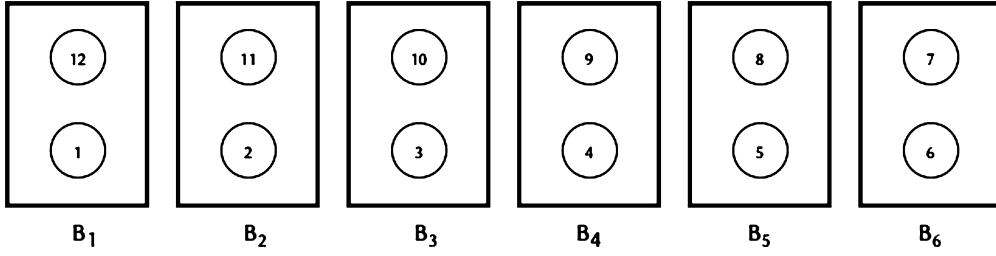
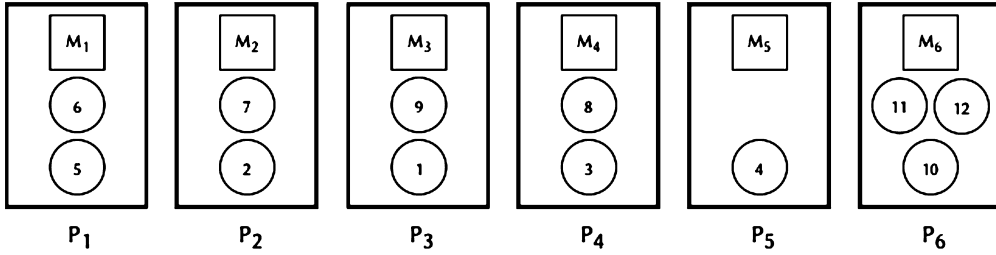
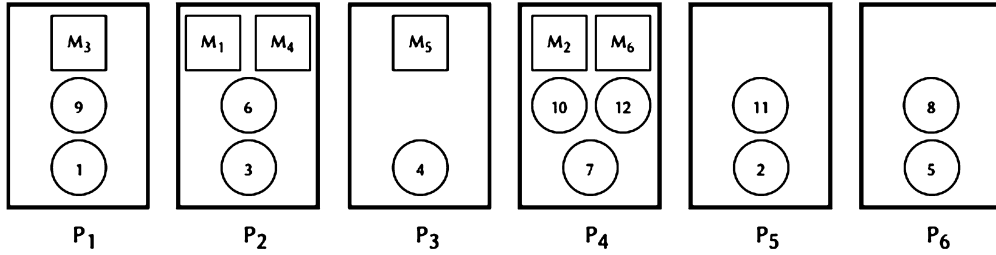


Fig. 2. Setting of the items in the bags in Example 3.10.

Table 2

Bags and their valuations in Example 3.10 where $(H_a \cup L_a)^c$ refers to the bags not in $(H_a \cup L_a)$.

Bag	B_1	B_2	B_3	B_4	B_5	B_6
Setting	{1, 12}	{2, 11}	{3, 10}	{4, 9}	{5, 8}	{6, 7}
$v_a(B_k)$	$\frac{29}{40}$	$\frac{33}{40}$	$\frac{41}{40}$	$\frac{42}{40}$	$\frac{31}{40}$	$\frac{29}{40}$
Bag type	L_a^1	$(H_a \cup L_a)^c$	H_a	H_a	$(H_a \cup L_a)^c$	L_a^2

Fig. 3. An MMS partition (P) of agent a in Example 3.10.Fig. 4. Re-enumerated P in Example 3.10.

Now let P be an MMS partition of agent a satisfying the condition of Lemma 3.8 be as Fig. 3. $M_k \subseteq (M \setminus J)$ is the set of low value items in P_k so $\bigcup_{k \in [6]} M_k = M \setminus J$. M_k 's can be possibly empty. Then, the enumerated P is shown in Fig. 4. Observe that:

- $v_a(B_2) \in [\frac{3}{4}, 1]$. Therefore, items 2 and 11 are put into one bundle (P_5) and all remaining items, which were previously in the same bundle with each of these items, are in another bundle (P_4). This is the case for B_5 and items 5 and 8 too. These bundles are put in the last (P_5 and P_6).
- $B_1 \in L_a^1$ and also $v_{a1} > \frac{5}{8}$. Therefore, we set the bundle containing item 1 to be P_1 . B_1 is the only bag with this property and hence $t_1 = 1$.
- $B_3, B_4 \in H_a$. Therefore, we put the bundles containing item 3 and item 4 next (P_2 and P_3). Therefore, $t_2 = 3$.
- The remaining bundle is set to be next (P_4). It means that $t_3 = 4$.

Definition 3.11. Define $y := v_{ak'}$ where k' is the highest value item in the set $\{j \in J_2 \cap B_k, B_k \in H_a\}$.

Observe that y is the value of the highest valued item from J_2 (items in top row of Fig. 1), which is in a bag with value more than 1. Further, $\frac{2}{8} < y < \frac{3}{8}$ since by definition $y + v_{ij} > 1$ for some $j \in J_1$ and $y = v_{ij'}$ for some $j' \in J_2$. For instance,

in Fig. 1, if all bags in $\{B_{k+1}, \dots, B_n\}$ value at most 1 for agent a and $v_a(B_k) > 1$, i.e., B_k is in H_a , then $y = v_a(2n-k+1)$. This also implies that $v_{a1} \geq \dots \geq v_{ak} > \frac{5}{8}$. Moreover, $y = v_a(2n-k+1) \leq v_a(2n-k) \leq \dots \leq v_a(k+1)$. In Example 3.10, $y = \frac{14}{40}$.

The value y turns out to be crucial, which we use to relate the total value of $\bigcup_{k=1}^{t_3} P_k \cap (M \setminus J)$ to the value needed in bag filling procedure. Intuitively, if y is high, then less value is needed from $M \setminus J$ to make all bundles in $\{P_1, \dots, P_{t_3}\}$ to be at least 1. This is also true for the bag filling argument: If y is high then less value is needed to add to the bags in (3) to make each of them at least $\frac{3}{4}$. Otherwise, when y is low, the value of $M \setminus J$ can be shown to be relatively high. We begin with the following lemma.

Lemma 3.12. For any bundle $B_k = \{j, j'\} \in L_a$ such that $v_{aj}, v_{aj'} < \frac{5}{8}$, we have $v_{aj}, v_{aj'} \geq y$.

Proof. The claim follows from the construction of B_k 's, and the fact that each bundle of H_a has an item with value more than $\frac{5}{8}$. Let k be the index of the bundle in (3) for which the y in Definition 3.11 is achieved. Then

$$v_{a1} \geq \dots \geq v_{ak} > \frac{5}{8} \geq v_{a(k+1)} \geq \dots \geq v_{a(2n-k+1)} = y \geq v_{a(2n-k+2)} \geq \dots \geq v_{a(2n)}.$$

If $v_{aj} < \frac{5}{8}$ for $j \in J_1$, then $j > k$ and $j' < 2n - k + 1$, implying the result. $\square$

First, we only consider the case when bags in L_a satisfy Lemma 3.12. This is equivalent to the case when $t_1 = 0$. Then, we prove the general case in Theorem 3.15.

In the bags, which are in H_a , there are exactly $2h_a$ items from J . However, in $\{P_{t_1+1}, \dots, P_{t_2}\}$, we know that each bundle has one item with value $> \frac{5}{8}$ but may contain more items from J . We define z to reflect the possible difference, i.e., $z := \max\{2h_a - |\bigcup_{t=t_1+1}^{t_2} P_t \cap J|, 0\}$. In Lemma 3.13, we show two things. First we show a bound for the value that is needed with respect to y to make bags in L_a to become at least $\frac{3}{4}$ (see definition of x_i in (4)). Second, we show a bound for $M \setminus J$ with respect to z and y by using the fact that the value of each bundle in $\{P_1, \dots, P_{t_3}\}$ is at least 1. As we discussed before, we will relate these two observations to prove the bound for $v_a(M \setminus J)$ and we show (6) is always correct.

Lemma 3.13. For agent $a \in N^2$, if $t_1 = 0$ then we have

1. $x_a \leq (\frac{3}{4} - 2y)l_a$.
2. $v_a(M \setminus J) \geq \max\{x_a + \frac{l_a}{4} - yz + \frac{z}{4}, \frac{z}{4}\}$, where $z = \max\{2h_a - |\bigcup_{t=t_1+1}^{t_2} P_t \cap J|, 0\}$.

Proof. For the first part, since $t_1 = 0$, we have $v_a(B_k) \geq 2y, \forall B_k \in L_a$ (Lemma 3.12). This implies that $x_a = (\frac{3}{4})l_a - \sum_{k: B_k \in L_a} B_k \leq (\frac{3}{4} - 2y)l_a$.

For the second part, in $\bigcup_{k: B_k \in H_a} B_k$, there are exactly $2h_a$ items from J . If there are at least $2h_a$ items in $\bigcup_{t=t_1+1}^{t_2} P_t \cap J$ then $z = 0$. On the other hand, if there are $2h_a - z$ items in $\bigcup_{t=t_1+1}^{t_2} P_t \cap J$, then it implies that there are at least z bundles in $\{P_{t_1+1}, \dots, P_{t_2}\}$ with exactly one item from J . Each of these bundles need more than $\frac{1}{4}$ value of items from $M \setminus J$ to be 1.

Next, if $z = 0$, it means that there are at least $2h_a$ items from J in $\{P_{t_1+1}, \dots, P_{t_2}\}$ and $2l_a$ items from J in $\{P_{t_2+1}, \dots, P_{t_3}\}$ since $t_1 = 0$. Then all bundles in $\{P_{t_2+1}, \dots, P_{t_3}\}$ need at least $x_a + \frac{l_a}{4}$ value of items from $M \setminus J$ to become 1 because the value of each item in the bundles of L_a is at least y (follows from Lemma 3.12 and Observation 3.9). If $z > 0$, then all bundles in $\{P_{t_2+1}, \dots, P_{t_3}\}$ need at least $x_a + \frac{l_a}{4} - yz$ value of items from $M \setminus J$ to become one because each of the z items has value at most y (Lemma 3.12). Therefore, in total, there should be at least $x_a + \frac{l_a}{4} - yz + \frac{z}{4}$ value of items from $M \setminus J$ in $\{P_{t_1+1}, \dots, P_{t_3}\}$. Letting $Q = P_{t_2+1} \cup \dots \cup P_{t_3}$, we have (using $v(P_t) \geq 1$ and $t_3 - t_2 = l_a$, and abusing notation):

$$\begin{aligned} l_a &\leq v_a(Q) \\ &= v_a(Q \cap J) + v_a(Q \cap (M \setminus J)) \\ &= v_a(Q \cap H_a) + v_a(Q \cap L_a) + v(Q \cap (M \setminus J)) \\ &\leq zy + v_a(L_a) + v_a(Q \cap (M \setminus J)) \\ &= zy + (\frac{3}{4})l_a - x_a + v_a(Q \cap (M \setminus J)), \end{aligned}$$

implying $v_a(Q \cap (M \setminus J)) \geq x_a + \frac{l_a}{4} - yz$, where the second-to-last line uses the fact that $|Q \cap J| = 2l_a + z$, so obviously $v_a(Q \cap J) \leq v_a(L_a) + zy$ (any items from H_a that end up in Q are worth at most y , which is at most the value of any L_a item), and the last line follows from substituting $x_a = (\frac{3}{4})l_a - v_a(L_a)$.

Further, in case z is large such that the value of $x_a + \frac{l_a}{4} - yz + \frac{z}{4}$ is negative, we still have z bundles in $\{P_{t_1+1}, \dots, P_{t_2}\}$ with exactly one item (with value of at most $\frac{3}{4}$) from J in each. Therefore, we need $\frac{z}{4}$ value of items from $M \setminus J$ to make all bundles in $\{P_{t_1+1}, \dots, P_{t_2}\}$ value 1. $\square$

In Lemma 3.13, we showed that $v_a(M \setminus J) \geq \max\{x_a + \frac{l_a}{4} - yz + \frac{z}{4}, \frac{z}{4}\}$. In Theorem 3.14, we prove (6) (for the special case when $t_1 = 0$) by relating the two observations from Lemma 3.13 and using two different cases for $\max\{x_a + \frac{l_a}{4} - yz + \frac{z}{4}, \frac{z}{4}\}$.

Theorem 3.14. For agent $a \in N^2$, if $t_1 = 0$ then

$$v_a(M \setminus J) \geq x_a + \frac{l_a}{8}. \quad (7)$$

Proof. Using Lemma 3.13(2) there are two cases for the maximum of $x_a + \frac{l_a}{4} - yz + \frac{z}{4}$ and $\frac{z}{4}$. We prove the claim for each case separately.

Case 1: Suppose $x_a + \frac{l_a}{4} - yz + \frac{z}{4} \geq z/4$, then $z \leq \frac{x_a + \frac{l_a}{4}}{y}$. Further, Lemma 3.13(1) implies $y \leq \frac{3l_a - 4x_a}{8l_a}$. Using these, we get

$$v_a(M \setminus J) \geq x_a + \frac{l_a}{4} - z(y - \frac{1}{4}) \geq \frac{4x_a + l_a}{16y} \geq \frac{(4x_a + l_a)l_a}{2(3l_a - 4x_a)}.$$

Next, consider

$$\frac{(4x_a + l_a)l_a}{2(3l_a - 4x_a)} - \left(x_a + \frac{l_a}{8}\right) = \frac{28x_a^2 + (2x_a - l_a)^2}{8(3l_a - 4x_a)} \geq 0.$$

The last inequality follows because $(3l_a - 4x_a) > 0$ (Lemma 3.4). This implies $v_a(M \setminus J) \geq x_a + \frac{l_a}{8}$.

Case 2: Suppose $x_a + \frac{l_a}{4} - yz + \frac{z}{4} \leq \frac{z}{4}$. Then, using Lemma 3.13(1), we have

$$v_a(M \setminus J) \geq \frac{z}{4} \geq \frac{4x_a + l_a}{16y} \geq \frac{(4x_a + l_a)l_a}{2(3l_a - 4x_a)} \geq x_a + \frac{l_a}{8}. \quad \square$$

Next, we handle the general case when $t_1 > 0$. Let $J_{\min} = \bigcup_{B_k \in L_a^1} B_k \cap J_2$ (recall that L_a^1 denote the set of bags in L_a that have one item with value more than $\frac{5}{8}$). Then, we have $v_{aj} < \frac{1}{8}, \forall j \in J_{\min}$ because these items are bagged with an item value more than $\frac{5}{8}$ and together they have value less than $\frac{3}{4}$. Therefore, $v_a(J_{\min}) < \frac{t_1}{8}$. Let us call items in $(M \setminus J) \cup J_{\min}$ as *filler* items (each filler item, by Lemma 3.4(iv), has value $< \frac{1}{8}$) and items in $J \setminus J_{\min}$ as *base* items.

Theorem 3.15. For agent $a \in N^2$, we have $v_a(M \setminus J) \geq x_a + \frac{l_a}{8}$.

Proof. In this proof we only consider the items in $\{P_1, \dots, P_{t_3}\}$ because, as discussed before, these bundles contain all items from $M \setminus J$. Let $x'_a := \frac{3}{4}t_1 - \sum_{B_k \in L_a^1} v_a(B_k)$, and $x''_a := x_a - x'_a$. For agent a , we can treat items of $J_{\min}$ as low-value items so we will prove:

$$v_a(M \setminus J) + v_a(J_{\min}) \geq x'_a + \frac{t_1}{4} + x''_a + (l_a - t_1)/8. \quad (8)$$

Since $v_a(J_{\min})$ is at most $\frac{t_1}{8}$, (8) directly implies the theorem.

In $\{P_1, \dots, P_{t_3}\}$, recall that each of $\{P_1, \dots, P_{t_1}\}$ contains a base item whose value is at least the maximum of all the items in the remaining bundles. Further, if there are two base items in any bundle P_k in $\{P_1, \dots, P_{t_1}\}$, then $v_a(P_k \cap J) > 1$ (using Observation 3.9). If each bundle in $\{P_1, \dots, P_{t_1}\}$ has exactly one base item (t_1 bundles in total) then items of value at least $x'_a + \frac{t_1}{4}$ are needed to make all these bundles at least 1. This, together with Theorem 3.14, proves the bound. On the other hand, if there are more than t_1 base items in $\{P_1, \dots, P_{t_1}\}$, by Observation 3.9(vi), each base item in $\{P_1, \dots, P_{t_3}\}$ has value $> \frac{1}{4}$. Therefore, for each extra base item j , there are two cases. If j comes from $B_k \in L_a^2$ then clearly, there should be enough items in $M \setminus J$ to compensate for v_{aj} in $\{P_{t_2+1}, \dots, P_{t_3}\}$ since the value item j is adding to a bag in L_a^1 is more than what it was considered in the right side of (8). For the other case, if j comes from $B_k \in H_a$ then it will make $B_k \setminus \{j\}$ to need more low-value items than the bundle in $\{P_1, \dots, P_{t_1}\}$ who gets j to become one since if $B_k \in H_a$ and $B_{k'} \in L_a^1$ then $v_{ak} < v_{ak'}$ (Observation 3.9(v)). This completes the proof. $\square$

Algorithm 4: $\frac{3}{4}$ -MMS allocation.

Input : Ordered Instance $\langle N, M, V \rangle$, i.e., $v_{i1} \geq \dots \geq v_{im}, \forall i \in N$
Output: $\frac{3}{4}$ -MMS Allocation

```

1 Normalize Valuations (i.e., scale valuations so that  $v_i(M) = n, \forall i \in N$ ) //  $n = |N|$ 
2  $(N, M, V) \leftarrow \text{Fixed-Assignment}(N, M, V)$  // Algorithm 5
3  $(N^t, M^t, V^t) \leftarrow \text{Tentative-Assignment}(N, M, V)$  // Algorithm 6
4 while  $N^{21} \neq \emptyset$  do
5    $i \leftarrow$  the lowest index agent in  $N^{21}$  //  $N^{21}$  is defined in (9)
6   Undo Tentative-Assignments // go back to  $(N, M, V)$ 
7   Update  $i$ 's MMS bound & normalize valuations // Theorem 4.5
8    $(N, M, V) \leftarrow \text{Fixed-Assignment}(N, M, V)$  // these assignments are final
9    $(N^t, M^t, V^t) \leftarrow \text{Tentative-Assignment}(N, M, V)$ 
10 Make all tentative assignments final:  $(N, M, V) \leftarrow (N^t, M^t, V^t)$ 
11 Bag-Filling( $N, M, V, \alpha$ ) for  $\alpha = \frac{3}{4}$  // Section 4.3 and Algorithm 3

```

4. Algorithm for $\frac{3}{4}$ -MMS allocation

The existence proof of $\frac{3}{4}$ -MMS allocation in Section 3 requires the knowledge of the exact MMS value μ_i 's of all agents, e.g., the proof of valid reduction for bundle S_4 in Lemma 3.1 needs this assumption. Finding an exact μ_i of an agent i is an NP-Hard problem, however a PTAS exists [13]. This implies a PTAS to compute a $(\frac{3}{4} - \epsilon)$ -MMS allocation for an $\epsilon > 0$. However, for small ϵ , this PTAS is computationally very expensive and may not be practical. In this section, we show that our algorithmic technique in Section 3 is powerful enough to be modified into a strongly polynomial time algorithm to find an exact $\frac{3}{4}$ -MMS allocation for a given instance $\mathcal{I} = \langle N, M, V \rangle$. The key idea is to use the average as an upper-bound for the MMS value; see Lemma 2.3.

As it is shown in Section 2.1, we assume $\mathcal{I}$ to be an ordered instance, i.e., $v_{i1} \geq v_{i2} \geq \dots \geq v_{im}$, for all $i \in N$. We fix the order of agents in N such that at each step of the algorithm if more than one agent satisfies the conditions of valid reduction, defined in Section 2.1.3, we choose one with the lowest index. We maintain the same order among remaining agents after every valid reduction. For the ease of exposition, as in Section 3, we abuse notation and use M and N to denote the set of unallocated items and the set of agents who have not received any bundle yet, respectively. Moreover, we use $n := |N|$ and $m := |M|$. Further, we use j to denote the j^{th} highest value item in M , and i to denote the i^{th} agent in N as per the fixed order.

In this section, unlike what we did in Section 3 (normalizing the valuation of items based on actual $\mu_i^n(M)$), we normalize v_{ij} 's using the total value of all items.

Definition 4.1. We call an instance $\mathcal{I} = \langle N, M, V \rangle$ *normalized* if $v_i(M) = n$ for all $i \in N$.

Using the scale invariance property (Lemma 2.4), we can without loss of generality work with the normalized valuations so that $v_i(M) = n$ for all agents $i \in N$. The proof of following corollary is straightforward.

Corollary 4.2. For a normalized instance $\mathcal{I} = \langle N, M, V \rangle$, $\mu_i^n(M) \leq 1 \ \forall i$.

The algorithm to compute a $\frac{3}{4}$ -MMS allocation is given in Algorithm 4. It has three main parts: Initial Assignment, Update Upper Bound, and Bag Filling. Initial Assignment is further divided into two parts: Fixed and Tentative.

4.1. Initial assignment

This section is very similar to Section 3.1 where we assign S_1, S_2, S_3 , and S_4 to agents in order to reduce the number of high-value items to at most $2n$. The main difference for this section is assigning S_4 . Since assigning S_4 is a *valid reduction* only if $\mu_i^n(M) = 1$ (Lemma 3.1), therefore, when S_1, S_2 , and S_3 value less than $\frac{3}{4}$ for all agents we check for S_4 and if the first S_4 is allocated, then this allocation and all initial allocations after that are called *tentative assignments*. Tentative assignment are finalized only if allocation of any S_4 did not take away too much valuation for any agent in N .

It is crucial for this step to assign bundles to agents according to an order. Therefore, for S_1, S_2, S_3 , and S_4 the bundle with lower index has more priority and if there are more than one agent satisfied with $S \in \{S_1, S_2, S_3, S_4\}$ we choose the one with lower index based on the original order of N . This ordering will be useful when we have to undo tentative assignments.

Update N, M, V After every assignment, say bundle S to agent i , we update N, M, V as follows:

$$M \leftarrow M \setminus S; \ N \leftarrow N \setminus \{i\}; \ v_{i'j} \leftarrow v_{i'j} \cdot \frac{|N|}{v_{i'}(M)}, \forall i' \in N, j \in M.$$

Note that this maintains $\mu_i^n(M) \leq 1, \forall i \in N$ after normalization (Definition 4.1 and Lemma 2.3).

Algorithm 5: Fixed-assignment.

Input : Ordered Instance (N, M, V) (i.e., $v_{i1} \geq \dots \geq v_{im}, \forall i \in N$), where $\mu_i^n(M) \leq 1, \forall i \in N$
Output: Fixed Assignments and Reduced Instance

```

1 For any  $S \subseteq M$ , define  $\Gamma(S) = \{i \in N : v_i(S) \geq \frac{3}{4}\}$ 
2  $S_1 := \{1\}; S_2 := \{n, n+1\}; S_3 := \{2n-1, 2n, 2n+1\}$  // bundles that can be assigned
3  $T = \Gamma(S_1) \cup \Gamma(S_2) \cup \Gamma(S_3)$ 
4 while  $T \neq \emptyset$  do
5    $S \leftarrow$  the lowest index bundle in  $\{S_1, S_2, S_3\}$  for which  $\Gamma(S) \neq \emptyset$ 
6    $i \leftarrow$  the lowest index agent in  $\Gamma(S)$ 
7   Assign  $S$  to agent  $i$  // final assignment
8   Update  $N, M, V, T$ 
9 return  $N, M, V$ 

```

Algorithm 6: Tentative-assignment.

Input : Ordered Instance (N, M, V) that satisfies Conditions 1-3 in Corollary 3.3
Output: Tentative Assignments and Reduced Instance

```

1 For any  $S \subseteq M$ , define  $\Gamma(S) = \{i \in N : v_i(S) \geq \frac{3}{4}\}$ 
2  $S_1 := \{1\}; S_2 := \{n, n+1\}; S_3 := \{2n-1, 2n, 2n+1\}; S_4 := \{1, 2n+1\}$ 
3  $T = \Gamma(S_1) \cup \Gamma(S_2) \cup \Gamma(S_3) \cup \Gamma(S_4)$ 
4 while  $T \neq \emptyset$  do
5    $S \leftarrow$  the lowest index bundle in  $\{S_1, S_2, S_3, S_4\}$  for which  $\Gamma(S) \neq \emptyset$ 
6    $i \leftarrow$  the lowest index agent in  $\Gamma(S)$ 
7   Assign  $S$  to agent  $i$  // tentative assignment
8   Update  $N, M, V, T$ 
9 return  $N, M, V$ 

```

4.1.1. Fixed assignment

In this part, we allocate high-value items to agents using Algorithm 5. We assign the bundle $S \in \{S_1, S_2, S_3\}$ to the lowest index agent $i \in N$ for which $v_i(S) \geq \frac{3}{4}$. Then, we update N, M, V and repeat this step until no such agent exists. Lemma 3.1 is applicable here to show that allocating $S \in \{S_1, S_2, S_3\}$ to agent i and removing them from M and N is a *valid reduction*.

As in Section 3, let $J_1 := \{1, \dots, n\}$ denote the set of first n items. Similarly, define $J_2 := \{n+1, \dots, 2n\}$ and $J := J_1 \cup J_2$. Note that Corollary 3.3 is also applicable here.

4.1.2. Tentative assignment

This step is a continuation of Fixed Assignment in Section 4.1.1. It starts when Algorithm 5 terminates and the first S_4 bundle is being assigned. In this part, we assume that $\mu_i^n(M) = 1, \forall i \in N$, i.e., the actual MMS value for each agent is equal to her current MMS upper bound, and we allocate S_1, S_2, S_3, S_4 using this assumption. In the next step, either this assumption works fine and we make all tentative assignments final and move to the next stage of bag filling or we detect an agent i for whom $\mu_i^n(M)$ is significantly lower than 1. In the latter case, we update i 's MMS upper bound. In particular, we update the MMS upper bound by updating the valuations so that the new MMS upper bound remains 1 for every agent.

In Algorithm 6, for $S \in \{S_1, S_2, S_3, S_4\}$ we check whether there exists an $i \in N$ with $v_i(S) \geq \frac{3}{4}$. If true, then we *tentatively* assign the lowest index such a bundle S to the lowest index such an agent i and we *tentatively* update M, N, V as we did in Section 4.1.1. Choosing the lowest index S_k makes sure that when S_4 is assigned, then none of S_1, S_2, S_3 satisfies the condition. This is essential in proving that assigning S_4 to an agent is a valid reduction in Lemma 3.1. Further, note that when S_4 is assigned, the value of each bundle in $\{S_1, S_2, S_3\}$ for every agent is strictly less than $\frac{3}{4}$. However, after removing an agent i with S_4 , it may later trigger valid reductions with $\{S_1, S_2, S_3\}$.

4.2. Updating MMS upper bound

The goal here is first to detect the agents for whom the MMS upper bound is overestimated and second to update this bound for them. To detect these agents we continue as in Section 3 by first initializing n bags as in (3) (see Fig. 1), and divide agents into two types N^1 and N^2 according to their valuations for these bags. Recall that N^1 is the set of agents whose value for each bag is at most 1, and N^2 is the set of remaining agents.

In this section, we need to analyze agents in N^2 more thoroughly. We further partition agents in N^2 into two sub-types as follows:

$$N^{21} := \{i \in N^2 : h_i > l_i \text{ and } v_i(M \setminus J) < x_i + l_i/8\}; \quad N^{22} := N^2 \setminus N^{21}. \quad (9)$$

Lemma 4.3. *If $h_i \leq l_i$ for $i \in N^2$ we have $v_i(M \setminus J) \geq x_i + \frac{l_i}{8}$.*

Proof. The value of any bag in H_i is less than $\frac{9}{8}$ for i (Lemma 3.4). By (4), the value of any bag which is not in $H_i \cup L_i$ is at most 1 and the total value of bags in L_i is at most

$$\frac{3}{4}l_i - x_i.$$

The total value of the items in M , in a normalized instance, is n so we have

$$\begin{aligned} v_i(M \setminus J) &\geq n - \left[\left(\frac{9}{8}h_i \right) + (n - h_i - l_i) + \left(\frac{3}{4}l_i - x_i \right) \right] \\ &= -\frac{h_i}{8} + \frac{l_i}{4} + x_i \geq x_i + \frac{l_i}{8}. \end{aligned} \quad \square$$

Lemma 4.3 shows that (6) holds for all $i \in N^{22}$. Hence, by Lemma 3.5, if N^{21} is empty, then we do not need to update the MMS upper bounds, and we can proceed to the next stage of bag filling. Note that Theorem 3.15 is still applicable for all agents in N^2 , which implies the following corollary.

Corollary 4.4. For any agent $i \in N^{21}$, $\mu_i < 1$.

The next theorem gives a new upper bound for these agents.

Theorem 4.5. For any $i \in N^{21}$, $\mu_i \leq \alpha^* = \max\{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5\}$, where

$$\begin{aligned} \alpha_1 &= \frac{4}{3}v_{i1} & \alpha_2 &= \frac{4}{3}(v_{in} + v_{i(n+1)}) \\ \alpha_3 &= \frac{4}{3}(v_{i(2n-1)} + v_{i(2n)} + v_{i(2n+1)}) & \alpha_4 &= \frac{4}{3}(v_{ik} + v_{ik'}) \\ \alpha_5 &= \max \left\{ \alpha : \frac{v_i(M \setminus J)}{\alpha} \geq \sum_{B_k: \frac{v_i(B_k)}{\alpha} < \frac{3}{4}} \left(\frac{3}{4} - \frac{v_i(B_k)}{\alpha} \right) + \frac{1}{8} \left| \left\{ B_k : \frac{v_i(B_k)}{\alpha} < \frac{3}{4} \right\} \right| \right\}, \end{aligned}$$

where k and k' are respectively the maximum value items from J and $M \setminus J$ that were not tentatively assigned, and $\langle N, M, V \rangle$ refers to the instance after we undo the tentative assignments in Line 6 of Algorithm 4.

Proof. For a contradiction, suppose $\mu_i = \beta$ and $\beta > \alpha^*$. We have

$$\begin{aligned} \beta > \alpha_1 &\implies v_{i1} < \frac{3}{4}\beta \\ \beta > \alpha_2 &\implies (v_{in} + v_{i(n+1)}) < \frac{3}{4}\beta \\ \beta > \alpha_3 &\implies (v_{i(2n-1)} + v_{i(2n)} + v_{i(2n+1)}) < \frac{3}{4}\beta \\ \beta > \alpha_4 &\implies (v_{ik} + v_{ik'}) < \frac{3}{4}\beta. \end{aligned}$$

From this, we can conclude that $v_i(S) < (\frac{3}{4})\mu_i$ for $S \in \{S_1, S_2, S_3\}$. Therefore, agent i will not be satisfied in Fixed-Assignment step. Further, since each bundle in $\{S_1, S_2, S_3\}$ is worth strictly less than $(\frac{3}{4})\mu_i$ to agent i , all S_4 bundles, which are tentatively assigned, (see Algorithm 6) have value at most μ_i , and hence all tentative assignments are valid reductions (Lemma 3.1). Also, since $\beta > \alpha_4$, no new pair that form S_4 will satisfy i . Furthermore, by normalizing v_{ij} 's (i.e., scaling them by $1/\beta$), the value of the bags in (3) will increase and consequently h_i will not decrease therefore, h_i will remain greater than 0. Consequently, since $h_i > 0$ and $\beta > \alpha_5$, agent i will remain in N^{21} even after we normalize her valuation with respect to β . This is because, by definition of α_5 , this is the maximum value that we can normalize the valuation such that agent i will not satisfy the definition of N^{21} in (9). Note that by normalizing the valuations, x_i and l_i will be affected that we have taken into account in definition of α_5 .

It implies that $\mu_i < \beta$ (Corollary 4.4), which is a contradiction. $\square$

Remark 4.6. It is easy to see from (9) that if we scale the valuations by $\frac{1}{\alpha}$ for $\alpha = v(M \setminus J)/(x_i + \frac{l_i}{8})$, then agent i will not remain in N^{21} . However, α_5 can be more than α because if we scale the valuations by $\frac{1}{\alpha}$ for $0 < \alpha < 1$ both x_i and l_i will decrease (see (4)).

Remark 4.7. α_5 can be computed in $O(n)$ time using a simple procedure. Sort the value of bags in L_i . Let u^k be the value of the k^{th} highest value bag in L_i . If $\alpha \in (1, u^1/(\frac{3}{4}))$ and we scale the valuations by $\frac{1}{\alpha}$ then all bags, which were in L_i will remain in L_i . Similarly, if $\alpha \in [u^k/(\frac{3}{4}), u^{k+1}/(\frac{3}{4})]$ then l_i will be reduced by k . After figuring out which bags remain in L_i and which bags do not, we can compute x_i with respect to α . Then, we can check whether there exists an α in the range for each k , starting with $k = 1$, that holds the definition of α_5 . If not, then we move to the next k .

Table 3Valuation of agent i for items in J in Example 4.8.

Item	1 – 27	28 – 30	31	32	33 – 60
v_{ij}	$\frac{29}{40}$	$\frac{15}{40}$	$\frac{14}{40}$	$\frac{13}{40}$	$\frac{12}{40}$

Table 4

The value of the bags in Example 4.8.

Bag	$B_1 - B_{27}$	B_{28}	B_{29}	B_{30}
$v_i(B_k)$	$\frac{41}{40}$	$\frac{27}{40}$	$\frac{28}{40}$	$\frac{29}{40}$
Bag type	H_i	L_i	L_i	L_i

Example 4.8. Consider an example where, after the execution of Algorithm 6, $n = 30$ (hence $|J| = 60$). The valuation of some agent $i \in N^{21}$ for items in J is shown in Table 3 and her valuation for bags is shown in Table 4. In this example, we show how to find α_5 .

Let $v_i(M \setminus J) = \frac{9}{40}$. First, observe that $h_i = 27, l_i = 3, x_i = \frac{6}{40}$, and $v_i(M \setminus J) = \frac{9}{40} < \frac{21}{40} = \frac{l_i}{8} + x_i$. It shows that agent i belongs to N^{21} .

In order to obtain α_5 , we check how the value of l_i would change after scaling the valuations of agent i by $\frac{1}{\alpha_5}$.

- Let $\alpha_5 > \frac{29}{40}$ then $l_i = 3$. We check if there exists an α_5 in this range for which the following bound holds

$$\frac{1}{\alpha_5} \times v_i(M \setminus J) \geq \frac{3}{8} + \left(\frac{3}{4} - \frac{v_i(B_{28})}{\alpha_5}\right) + \left(\frac{3}{4} - \frac{v_i(B_{29})}{\alpha_5}\right) + \left(\frac{3}{4} - \frac{v_i(B_{30})}{\alpha_5}\right). \quad (10)$$

- If not, let $\frac{28}{40} < \alpha_5 \leq \frac{29}{40}$ then $l_i = 2$ (B_{30} will not belong to L_i anymore). We check if there exists an α_5 in this range for which the following bound holds

$$\frac{1}{\alpha_5} \times v_i(M \setminus J) \geq \frac{2}{8} + \left(\frac{3}{4} - \frac{v_i(B_{28})}{\alpha_5}\right) + \left(\frac{3}{4} - \frac{v_i(B_{29})}{\alpha_5}\right). \quad (11)$$

- If not, $\frac{27}{40} < \alpha_5 \leq \frac{28}{40}$ then $l_i = 1$. We check if there exists an α_5 in this range for which the following bound holds

$$\frac{1}{\alpha_5} \times v_i(M \setminus J) \geq \frac{1}{8} + \left(\frac{3}{4} - \frac{v_i(B_{28})}{\alpha_5}\right). \quad (12)$$

- Else, $\alpha_5 = \frac{27}{40}$ ($l_i = 0$).

Note that in (10), (11), and (12) we have obtained the sum of $\frac{l_i}{8}$ and the new x_i after scaling the valuations. It takes at most l_i steps (3 steps in this example) to find the maximum α_5 for which one of the bounds above holds.

We pick an agent i with the lowest index in N^{21} and update i 's valuation as $v_{ij} \leftarrow \frac{v_{ij}}{\alpha}$, $\forall j \in M$ and repeat. In Theorem 4.9, we show that the number of updates is at most n^3 in the entire run of algorithm.

4.3. Bag filling and running time

Note that we reach the bag filling stage only when N^{21} (see (9)) is empty. Here, we use Algorithm 3 of Section 3.2. Since N^{21} is empty, Lemma 3.5 is applicable, which implies a $\frac{3}{4}$ -MMS allocation. Next, we show that the entire algorithm runs in strongly polynomial time.

Theorem 4.9. *The entire algorithm runs in $O(n^5m)$ time for an ordered instance, and in $O(nm(n^4 + \log m))$ time for any instance.*

Proof. In Algorithm 4, normalization takes $O(mn)$ arithmetic operations. Both Fixed-Assignment and Tentative-Assignment procedures take at most $O(n^2m)$ arithmetic operations. The bag filling procedure takes at most $O(n^2m)$ time. Each iteration of the while loop takes $O(n^2)$ arithmetic operations to find the set N^{21} of agents, $O(m)$ arithmetic operations to update valuations of an agent $i \in N^{21}$, and $O(n^2m)$ arithmetic operations to run Fixed-Assignment and Tentative-Assignment procedures. Therefore, each iteration of the while loop takes at most $O(n^2m)$ arithmetic operations.

To bound the number of iterations of the while loop, we upper bound the number of times it is run for a particular agent, say a , in N^{21} . Consider the first iteration when a is the lowest index agent, then if we update a 's valuation due to $\alpha \in \{\alpha_1, \alpha_2, \alpha_3\}$ (see Section 4.2), then a gets a fixed assignment after this iteration, and we will not see her again. If we update a 's valuation due to $\alpha = \alpha_4$, then in the future iterations, agent a will not be in N^{21} unless the instance is reduced due to a fixed assignment. This could happen only when another agent in N^{21} ends her iteration due to $\alpha \in \{\alpha_1, \alpha_2, \alpha_3\}$. Clearly, this can occur at most $O(n)$ time. Finally, if we update a 's valuation due to $\alpha = \alpha_5$, then agent a will not be in N^{21} again unless the instance is reduced due to either fixed or tentative assignment, which can affect the set of agents in N^{21} .

This could happen only when another agent of N^{21} ends her iteration due to $\alpha \in \{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$. Clearly, this can occur at most $O(n^2)$ time. Therefore, the maximum number of iterations of the while loop is at most $O(n^3)$. Since each iteration takes at most $O(n^2m)$ arithmetic operations, Algorithm 4 takes $O(n^5m)$ time. Note that this is for an ordered instance.

The reduction from the general instance to ordered instance is given in Appendix A. Algorithm 7 takes $O(mn \log m)$ time to make the original instance ordered by creating a sorted list of items for each agent. Next, we show that Algorithm 8 can be implemented in $O(mn)$ time. Initialize a binary array A of size m to all ones. $A[j]$ indicates whether item j is available or not; 1 means available. Each agent has a sorted list of items from the Algorithm 7. From the solution of the ordered instance using Algorithm 4, we can create an array B of size m , where $B[j]$ stores the agent who is assigned the item j . Observe that B can be constructed in $O(m)$ time. Then, for each item j from 1 to m , we get the agent $B[j]$ who is assigned j , and then we try to give $B[j]$ the highest item in her list. If her highest item is not available, which we can check from A , then we delete this item from $B[j]$'s list, and move down to the next highest item and repeat until we reach at an available item. We give this item to $B[j]$ and mark it assigned in A . Since the size of each agent's list is m , the total time is $O(mn)$. Using Algorithms 7 and 8 together with Algorithm 4, the running time of the entire procedure is $O(nm(n^4 + \log m))$. $\square$

5. Existence of $(\frac{3}{4} + \frac{1}{12n})$ -MMS allocation

In this section, we show that our approach in Section 3 can be extended to obtain the existence of a $(\frac{3}{4} + \gamma)$ -MMS allocation for any given instance $\mathcal{I} = \langle N, M, V \rangle$ where $\gamma = \frac{1}{12n}$. We note that γ is a constant for the given instance, where $n := |N|$. We assume that the MMS value μ_i of each agent i is given. Finding an exact μ_i is an NP-Hard problem, however a PTAS exists [13]. This implies a PTAS to compute a $(\frac{3}{4} + \gamma - \epsilon)$ -MMS allocation for any $\epsilon > 0$. Using the properties shown in Section 2.1, we normalize valuations so that $\mu_1 = 1, \forall i$ and assume that $\mathcal{I}$ is an ordered instance, i.e., $v_{i1} \geq \dots \geq v_{i|M|}, \forall i$. Our proof is algorithmic. If more than one agent satisfies the conditions (1) of valid reduction, then we choose one arbitrarily.

For the ease of exposition, we abuse notation and use M and N to denote the set of unallocated items and the set of agents who have not received any bundle yet, respectively. Further, we use j to denote the j^{th} highest value item in M . Moreover, we use $n := |N|$ and $m := |M|$. This is the reason we use γ , which is a constant for a given instance, to denote the approximation factor.

The approach is identical to Section 3. Here, we run Algorithm 1 with $\alpha = \frac{3}{4} + \gamma$. The analysis is also almost same. Here, we crucially use the extra $\frac{1}{8}$ (see (6) and the paragraph after it) to obtain a better factor. To avoid repetition, we only highlight the differences.

5.1. Initial assignment

In Algorithm 2, we keep assigning a bundle $S \in \{S_1, S_2, S_3, S_4\}$ to agent i , if any, for which $v_i(S) \geq \frac{3}{4} + \gamma$. Then, we update M and N to respectively reflect the current unallocated items and agents who are not assigned any bundle yet. Observe that Lemma 3.1 is applicable here to show that assigning a bundle $S \in \{S_1, S_2, S_3\}$ is a *valid reduction*. However, it only applies for S_4 when $v_i(S_4) \leq 1, \forall i$.

As in Section 3, let $J_1 := \{1, \dots, n\}$ denote the set of first n items. Similarly, define $J_2 := \{n+1, \dots, 2n\}$ and $J := J_1 \cup J_2$. The following corollary is straightforward.

Corollary 5.1. *If $v_i(S) < \frac{3}{4} + \gamma, \forall i$ and $\forall S \in \{S_1, S_2, S_3\}$, then (i) $v_{ij} < \frac{3}{4} + \gamma, \forall j \in J_1$, (ii) $v_{ij} < \frac{3}{8} + \frac{\gamma}{2}, \forall j \in J_2$, and $v_{in} < \frac{3}{4} + \gamma - v_{i(n+1)}$, and (iii) $v_{ij} < \frac{1}{4} + \frac{\gamma}{3}, \forall j \in M \setminus J$, for all i .*

However, the value of S_4 might be strictly greater than 1 for an agent i . Corollary 5.1 implies that, $v_i(S_4) < \frac{3}{4} + \gamma + \frac{1}{4} + \frac{\gamma}{3} = 1 + \frac{4\gamma}{3}$. We use dummy items to fix the issue of extra $\frac{4\gamma}{3}$ lost in assigning S_4 .

Dummy items (D_1): For each removed S_4 in Algorithm 2, we add one dummy item d_j such that

$$v_i(\{d_j\}) = \frac{4\gamma}{3} \quad \forall i \in N. \quad (13)$$

Let D_1 denote the set of all dummy items after the termination of Algorithm 2. We note that dummy items will not be assigned to any agent. They are defined to make proofs easier. Later, we introduce two more sets D_2 and D_3 of dummy items in Section 5.2. Let $D := D_1 \cup D_2 \cup D_3$ with $D_2 = D_3 = \emptyset$ currently. The proof of the following corollary easily follows from Lemma 3.1.

Corollary 5.2. *For any remaining agent $i \in N$, $\mu_i^n(M \cup D) \geq 1$.*

The proof of the following lemma easily follows using (13) and Corollary 5.2.

Lemma 5.3. When Algorithm 2 terminates, we have $\mu_i^n(M) \geq 1 - \frac{4\gamma r}{3}$ and $v_i(M) \geq n - \frac{4\gamma r}{3}$, $\forall i$, where r is the total number of rounds in Algorithm 2 when S_4 is assigned.

5.2. Bag filling

The overall approach is same as in Section 3.2. We use the bag filling procedure given in Algorithm 3 to satisfy the remaining agent by setting $\alpha = \frac{3}{4} + \gamma$. We initialize the bags with items in J as in (3) (see Fig. 1). Algorithm 3 has n rounds. In each round k , it starts a new bundle T with $T \leftarrow B_k$. If there is an agent who values T to be at least $\frac{3}{4} + \gamma$, then assign T to such an agent. Otherwise, keep adding items from $M \setminus J$ to T one by one until someone values T at least $\frac{3}{4} + \gamma$.

For correctness, we need to show that there are enough items in $M \setminus J$ to add on top of each bag in (3) so that each agent gets a bundle that they value at least $\frac{3}{4} + \gamma$. For this, we first divide agents into two types: $N^1 := \{i \in N \mid v_i(B_k) \leq 1 + \frac{3\gamma}{2}, \forall k\}$ and $N^2 := N \setminus N^1$. Next, we update the notations of (4) to reflect the improved bound. For an agent $i \in N^2$, define

$$\begin{aligned} L_i &:= \{B_k : v_i(B_k) < \frac{3}{4} + \gamma\}; & l_i &:= |L_i| \\ H_i &:= \{B_k : v_i(B_k) > 1 + \frac{3\gamma}{2}\}; & h_i &:= |H_i| \\ x_i &:= (\frac{3}{4} + \gamma)l_i - \sum_{k: B_k \in L_i} v_i(B_k). \end{aligned} \quad (14)$$

The proof of the following lemma is an easy extension of Lemma 3.4, hence omitted.

Lemma 5.4. For an agent $i \in N^2$, (i) $l_i > 0$ and $h_i > 0$, (ii) $v_{i1} > \frac{5}{8} + \gamma$, (iii) $v_i(B_k) < \frac{9}{8} + \frac{3\gamma}{2}, \forall k$, and (iv) $v_{ij} < \frac{1}{8}, \forall j \in M \setminus J$.

The proof of the following lemma is an extension of the proof of Lemma 3.5, and is in Appendix A.

Lemma 5.5. If $v_i(M \setminus J) \geq x_i + \frac{l_i}{8} - \frac{1}{8}, \forall i \in N^2$, then Algorithm 3 with $\alpha = \frac{3}{4} + \gamma$ gives every agent a bundle that they value at least $\frac{3}{4} + \gamma$.

Now, we only need to show that for each $i \in N^2$, we have

$$v_i(M \setminus J) \geq x_i + \frac{l_i}{8} - \frac{1}{8} \quad (15)$$

We start with a few lemmas. Recall from (2) that $\mathcal{P}_i^n(M)$ denote the set of partitions where the value of each bundle is at least 1. The proofs of the following two lemmas are extensions of the corresponding Lemmas 3.6 and 3.8 respectively, and are given in Appendix A.

Lemma 5.6. For an agent $i \in N^2$, there exists a bundle P_k in every partition $P = \{P_1, \dots, P_n\} \in \mathcal{P}_i^n(M \cup D)$ such that $v_i(P_k \setminus J) > \frac{1}{4} - \gamma$.

Lemma 5.7. For every agent i , there exists a partition $P = \{P_1, \dots, P_n\} \in \mathcal{P}_i^n(M \cup D)$ such that each $P_k \in P$ has at most one item j with $v_{ij} > \frac{5}{8} + \gamma$.

Fix an agent, say a , in N^2 . For simplicity, until the end of this section, when we use *value* of an item or a bundle, we mean their value for agent a (unless mentioned otherwise). Recall that we need to show (15). Let $P = \{P_1, \dots, P_n\} \in \mathcal{P}_a^n(M \cup D)$ be a partition satisfying the condition of Lemma 5.7.

We manipulate P as follows: First, for each $B_i = \{j, j'\}$ with $\frac{3}{4} + \gamma \leq v_a(B_i) \leq 1 + \frac{3\gamma}{2}$, if $j \in P_k$ and $j' \in P_{k'}$, then we turn P_k and $P_{k'}$ into two new bundles $\{j, j'\}$ and $(P_k \cup P_{k'} \cup \{d_j\} \setminus \{j, j'\})$, where d_j is a new dummy item with $v_a(d_j) := \frac{3\gamma}{2}$.

Dummy items (D_2): Let D_2 denote the set of all dummy items added during the manipulation of P . Observe that $|D_2| = n - h_a - l_a$. Each $d_j \in D_2$ values $\frac{3\gamma}{2}$ for agent a . Therefore,

$$v_a(D_2) = \frac{3\gamma|D_2|}{2}. \quad (16)$$

Later we introduce one more set D_3 of dummy items. Recall that $D = D_1 \cup D_2 \cup D_3$ where D_1 is defined in (13) and $D_3 = \emptyset$ currently. Observe that $v_a(P_k \cup P_{k'} \cup \{d_j\} \setminus \{j, j'\}) \geq 1$. Hence, we can assume that for each $B_i \notin L_a \cup H_a$ (as defined in (14)), there exists a $P_k = B_i$ and all other bundles value at least 1. Second, we re-enumerate the bundles in P such that

- $P_1, \dots, P_{t_1}$: each has an item j of value more than $\frac{5}{8} + \gamma$ and $v_a(B_j) < \frac{3}{4} + \gamma$.

- $P_{t_1+1}, \dots, P_{t_2}$: each has an item j of value more than $\frac{5}{8} + \gamma$ and $v_a(B_j) > 1$ (Observe that there are h_a such bundles, so $t_2 - t_1 = h_a$).
- $P_{t_3+1}, \dots, P_n$: each such $P_k = B_{k'}$ for some k' and $\frac{3}{4} + \gamma \leq v_a(B_{k'}) \leq 1 + \frac{3\gamma}{2}$.
- $P_{t_2+1}, \dots, P_{t_3}$ be the remaining bundles (Observe that $t_3 = l_a + h_a$).

Definition 5.8. Define $y := v_{ak'}$ where k' is the highest value item in the set $\{j \in J_2 \cap B_k, B_k \in H_a\}$.

Observe that $\frac{1}{4} + \frac{\gamma}{2} < y < \frac{3}{8} + \frac{\gamma}{2}$. For instance, in Fig. 1, if no bag from $\{B_{k+1}, \dots, B_n\}$ is in H_a but $B_k \in H_a$, then $y = v_{a(2n-k+1)}$. This also implies that $v_{a1} \geq \dots \geq v_{ak} > \frac{5}{8} + \gamma$. Moreover, $y = v_{a(2n-k+1)} \leq v_{a(2n-k)} \leq \dots \leq v_{a(k+1)}$. The proof of the following lemma is an easy extension of the proof of Lemma 3.12, and hence omitted.

Lemma 5.9. For any bundle $B_k = \{j, j'\} \in L_a$ such that $v_{aj}, v_{aj'} < \frac{5}{8} + \gamma$, we have $v_{aj}, v_{aj'} \geq y$.

Next, we show the extension of Lemma 3.13, whose proof is given in Appendix A.

Lemma 5.10. For agent $a \in N^2$, if $t_1 = 0$ then we have

1. $x_a \leq (\frac{3}{4} + \gamma - 2\gamma)l_a$
2. $v_a((M \cup D) \setminus J) \geq \max\{x_a + l_a(\frac{1}{4} - \gamma) - yz + z(\frac{1}{4} - \gamma), z(\frac{1}{4} - \gamma)\}$, where $z = \max\{2h_a - |\bigcup_{t=t_1+1}^{t_2} P_t \cap J|, 0\}$.

The following lemma is an extension of Theorem 3.14, and its proof is given in Appendix A.

Theorem 5.11. For agent $a \in N^2$, if $t_1 = 0$ then

$$v_a(M \cup D \setminus J) \geq x_a + \frac{l_a}{8}. \quad (17)$$

Next, we handle the general case when $t_1 > 0$. Let $L_a^1 \subseteq L_a$ be the set of bags in L_a , which have one item with value more than $\frac{5}{8} + \gamma$, and $L_a^2 = L_a \setminus L_a^1$. Clearly, $|L_a^1| = t_1$ and $|L_a^2| = l_a - t_1$. Note that $|L_a^2| \geq 1$ because $B_n \in L_a^2$.

Dummy items (D_3): For each bag in L_a^1 we add a dummy item d_j where $v_a(d_j) = \gamma$. Therefore,

$$v_a(D_3) = |D_3| \cdot \gamma = |L_a^1| \cdot \gamma < l_a \cdot \gamma. \quad (18)$$

The following theorem is an extension of Theorem 3.15, and its proof is given in Appendix A.

Theorem 5.12. For agent $a \in N^2$, we have $v_a((M \cup D) \setminus J) \geq x_a + \frac{l_a}{8}$.

Next, we obtain the bound without the dummy items in the next theorem.

Theorem 5.13. For any agent $a \in N^2$, $v_a(M \setminus J) \geq x_a + \frac{l_a}{8} - \frac{1}{8}$.

Proof. From Lemma 5.3, (16), and (18), the total value of the items in D is at most

$$v_a(D) \leq \frac{4\gamma r}{3} + \frac{3\gamma|D_2|}{2} + \gamma l_a \leq \frac{3\gamma n}{2} \leq \frac{1}{8},$$

because $|D_2| + r \leq (n - h_a - l_a)$ and $\gamma = \frac{1}{12n}$, where n is the number of agents in the original instance. This, together with Theorem 5.12, proves the theorem. $\square$

5.3. Better approximation factor?

The algorithm is specifically designed for the approximation factor of $\frac{3}{4}$. The first issue that the algorithm faces by increasing the approximation factor, is that removing S_4 will no longer be a valid reduction (see Lemma 3.1 and Section 5.1). However, as we showed in Section 5, with the use of the extra $\frac{1}{8}$ value in $M \setminus J$ (Lemma 3.5 and Theorem 3.15), we could use the same algorithm for the factor of $\frac{3}{4} + \gamma$ with $\gamma = \frac{1}{12n}$. This raises a natural question of whether this is the best algorithm can do, i.e., are there tight examples for which the same algorithm cannot guarantee better than $\frac{3}{4} + \frac{1}{12n}$? We leave this as an interesting open question. However, we can show that our algorithm does not guarantee a factor greater than $\frac{3}{4} + \frac{1}{12}$.

Table 5
Valuation of agents for item $j, j \in M$ in Example 5.14.

Item	1	2 – 4	5 – m
v_j	$\frac{5}{6}$	$\frac{1}{3}$	$\frac{\epsilon}{2}$

Example 5.14. Let $\mathcal{I} = \langle N, M, V \rangle$ be an instance for which we want $\alpha (= \frac{5}{6} + \epsilon)$ -MMS allocation for some $\epsilon > 0$. Let $N = \{1, 2\}$ be the set of agents with identical valuations, $M = \{1, 2, \dots, m\}$ be the set of items where $m = \lceil \frac{1}{3\epsilon} \rceil + 4$, and $V = v$. Agents' valuation for each item is shown in Table 5.

Clearly, the MMS partition is $P_1 = \{2, 3, 4\}$ and $P_2 = \{1\} \cup \{5, \dots, m\}$ of M , which implies $\mu_i^n(M) = 1, \forall i$. When we run Algorithm 1 on $\mathcal{I} = \langle N, M, V \rangle$ then there will not be any initial assignment, and it will run the bag filling procedure with bags $B_1 = \{1, 4\}$ and $B_2 = \{2, 3\}$, which will not output an α -MMS allocation.

6. Conclusions

We developed a new approach that gives a simple algorithm for showing the existence of a $\frac{3}{4}$ -MMS allocation. Furthermore, we showed that our approach is powerful enough to be easily extended to obtain (i) a strongly polynomial time algorithm to find a $\frac{3}{4}$ -MMS allocation, and (ii) the existence of a $(\frac{3}{4} + \frac{1}{12n})$ -MMS allocation, improving the best previous factor. Consequently, this gives a PTAS for finding a $(\frac{3}{4} + \frac{1}{12n} - \epsilon)$ -MMS allocation for any $\epsilon > 0$. An interesting question is to find the maximum γ for which the proposed algorithm obtains $(\frac{3}{4} + \gamma)$ -MMS allocation. We could show that our algorithm fails for $\gamma > \frac{1}{12}$, but the question remains open for $\gamma \in (\frac{1}{12n}, \frac{1}{12}]$.

Further, it could be worth exploring whether extending the approach yields a $\frac{4}{5}$ -MMS allocation. Such an extension would be challenging because after the initial greedy assignments, there will be $3n$ high-value items, and this would make the process of initializing the bag filling procedure harder due to too many items to handle and also the value of some bags might exceed significantly more than 1.

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.

Acknowledgements

We would like to thank anonymous referees for their comments and suggestions that have helped to improve the presentation of the paper. Work on this paper supported by NSF Grant CCF-1942321 (CAREER).

Appendix A. Missing proofs

Lemma 2.3 (AVERAGE UPPER BOUNDS MMS). $\mu_i^n(M) \leq \frac{v_i(M)}{n}, \forall i \in N$.

Proof. Suppose for some agent i , $\mu_i^n(M) > v_i(M)/n$, i.e., there exists a partition of M into n bundles where all bundles have value strictly more than $\frac{v_i(M)}{n}$. Therefore, $v_i(M) \geq n \cdot \mu_i^n(M) > n \cdot v_i(M)/n = v_i(M)$, which is a contradiction. $\square$

Lemma 2.4 (SCALE INVARIANCE). Let $A = (A_1, \dots, A_n)$ be an α -MMS allocation for instance $\mathcal{I} = \langle N, M, V \rangle$. If we create an alternate instance $\mathcal{I}' = \langle N, M, V' \rangle$ where valuations of each agent i are scaled by $c_i > 0$, i.e., $v'_{ij} := c_i \cdot v_{ij}, \forall j \in M$, then $\mu'_i = c_i \cdot \mu_i$ and A is an α -MMS allocation for $\mathcal{I}'$.

Proof. For any bundle $S \subseteq M$, we have $v'_i(S) = c_i \cdot v_i(S)$. Therefore, $\mu'_i = c_i \cdot \mu_i$. Further, $v'_i(A_k) = c_i \cdot v_i(A_k) \geq c_i \cdot \alpha \cdot \mu_i = \alpha \cdot \mu'_i, \forall k$. $\square$

Lemma 2.5 (ORDERED INSTANCE [30,5]). Without loss of generality, we can assume that agents have the same order of preferences over the items, i.e., $v_{i1} \geq v_{i2} \geq \dots \geq v_{im}, \forall i \in N$.

Proof. Consider Algorithms 7 and 8. It is enough to show that given any instance $\mathcal{I} = \langle N, M, V \rangle$, we can find an ordered instance $\mathcal{I}' = \langle N, M, V' \rangle$ in polynomial time using Algorithm 7. Furthermore, given an α -MMS allocation A' for the ordered version $\mathcal{I}'$, we can find an α -MMS allocation A for the original instance $\mathcal{I}$ in polynomial time using Algorithm 8.

Clearly, items in $\mathcal{I}'$ are sorted by their values, and they have the same order for all agents. Also, it takes mn iterations to obtain $\mathcal{I}'$. Note that each agent's MMS value will remain the same in $\mathcal{I}'$ because it neither depends on the order of items

Algorithm 7: [5] Conversion to an ordered instance.

Input : Instance $\mathcal{I} = \langle N, M, V \rangle$
Output: Instance $\mathcal{I}' = \langle N, M, V' \rangle$ in which $v'_{i1} \geq \dots \geq v'_{im}$ for all $i \in N$

```

1 for  $i = 1$  to  $n$  do
2   for  $j = 1$  to  $m$  do
3      $j^* \leftarrow j^{\text{th}}$  highest value item of  $M$  for agent  $i$ 
4      $v'_{ij} \leftarrow v_{ij^*}$ 

```

Algorithm 8: [5] α -MMS allocation for unordered instance.

Input : Instance $\mathcal{I} = \langle N, M, V \rangle$, Ordered Instance $\mathcal{I}' = \langle N, M, V' \rangle$, α -MMS allocation $A' = (A'_i)_{i \in N}$ for $\mathcal{I}'$
Output: α -MMS allocation $A = (A_i)_{i \in N}$ for $\mathcal{I}$

```

1  $A_i \leftarrow \emptyset, \forall i$ 
2 for  $j = 1$  to  $m$  do
3    $i^* \leftarrow i \in N : j \in A'_i$  //  $i^*$  is the agent who has  $j^{\text{th}}$  item of  $\mathcal{I}'$  in her bundle
4    $j^* \leftarrow \arg \max_{j \in M \setminus (\bigcup_{i \in N} A_i)} v_{i^*j}$  //  $j^*$  is  $i^*$ 's favorite unassigned item
5    $A_{i^*} \leftarrow A_{i^*} \cup \{j^*\}$ 

```

nor on the valuations of other agents. We prove that $v_i(A_i) \geq v'_i(A'_i)$ for all $i \in N$. Consider the round r of the Algorithm 8 and consider item r from $\mathcal{I}'$ is in A'_i . It means that agent i is getting her r^{th} favorite item in A'_i but now after $r - 1$ round, exactly $r - 1$ items are allocated and she will get her favorite item among unallocated items. Therefore, the item she gets in this round is at least as valuable as the one she was getting in the ordered instance. Therefore, $v_i(A_i) \geq v'_i(A'_i) \geq \alpha \cdot \mu_i$. $\square$

Lemma 5.5. If $v_i(M \setminus J) \geq x_i + \frac{l_i}{8} - \frac{1}{8}, \forall i \in N^2$, then Algorithm 3 with $\alpha = \frac{3}{4} + \gamma$ gives every agent a bundle that they value at least $\frac{3}{4} + \gamma$.

Proof. This is proof by contradiction. Suppose the algorithm stops at round t because there are not enough items in L ($= M \setminus J$) to satisfy any remaining agent i , i.e., $v_i(B_t \cup L) < \frac{3}{4} + \gamma$.

If $i \in N^1$, each removed bundle in rounds $k \in [t - 1]$, has value of at most $1 + \frac{3\gamma}{2}$ for agent i . Because, if $v_i(B_k) \geq \frac{3}{4} + \gamma$ for $k \in [t - 1]$, no more item has been added to $T = B_k$. Also, if $v_i(B_k) < \frac{3}{4} + \gamma$ for $k \in [t - 1]$ before adding the last item (if any) to T the value of T is less than $\frac{3}{4} + \gamma$ and from Corollary 5.1, $v_{ij} < \frac{1}{4} + \frac{\gamma}{3}$ for $j \in L$. Therefore, at the end of the round $v_i(T) < 1 + \frac{3\gamma}{2}$.

This implies that the total value of assigned bundles in round 1 to $t - 1$ is at most $(t - 1)(1 + \frac{3\gamma}{2})$. Let n' be the number of agents remaining after Algorithm 2, and n be the number of agents in the original instance. We have $v_i(M) \geq n' - \frac{4\gamma r}{3}$ due to Lemma 5.3. Therefore, the value of items in L before the round t starts is at least

$$v_i(L) \geq n' - \frac{4\gamma r}{3} - \left((t - 1)(1 + \frac{3\gamma}{2}) + v_i(B_t) + (n' - t)(1 + \frac{3\gamma}{2}) \right).$$

This implies that

$$v_i(L) + v_i(B_t) \geq 1 - \frac{3(r + (n' - 1))\gamma}{2} \geq 1 - \frac{3(n - 1)\gamma}{2} \geq \frac{7}{8} + \frac{1}{8n} \geq \frac{3}{4} + \gamma,$$

where we use $\gamma = \frac{1}{12n}$, which is a contradiction.

If $i \in N^2$, then since at round t , $v_i(T) < \frac{3}{4} + \gamma$, we have $B_t \in L_i$. Consider a round $k \in [t - 1]$. If $B_k \notin L_i$, then $T = B_k$ has been assigned to someone with no additional items added to T from L because $i \in \Gamma(T)$. If $B_k \in L_i$, then in round k , before adding the last item (if any) to T , the value of i for T is less than $\frac{3}{4} + \gamma$ and from Lemma 5.4, all items in L have value of at most $\frac{1}{8}$. Therefore, if $B_k \in L_i$, the value of the assigned bag for i in round k is less than $\frac{7}{8} + \gamma$. Hence, in the beginning of the round t ,

$$v_i(L) \geq \left(x_i + \frac{l_i}{8} - \frac{1}{8} \right) - \left(x_i - \left(\frac{3}{4} + \gamma - v_i(B_t) \right) + (l_i - 1)/8 \right) = \frac{3}{4} + \gamma - v_i(B_t),$$

which is a contradiction. $\square$

Lemma 5.6. For an agent $i \in N^2$, there exists a bundle P_k in every partition $P = \{P_1, \dots, P_n\} \in \mathcal{P}_i^n(M \cup D)$ such that $v_i(P_k \setminus J) > \frac{1}{4} - \gamma$.

Proof. If there exists a bundle P_k with exactly one item j from J , then $v_i(P_k \setminus J) = 1 - v_{ij} > \frac{1}{4} - \gamma$ because value of every item is less than $\frac{3}{4} + \gamma$. Otherwise, each bundle has exactly two items from J , which implies that one of the bundles, say P_k , has two items j_1, j_2 from the set $\{n\} \cup J_2$. Since $v_{ij_1} + v_{ij_2} \leq v_{in} + v_{i(n+1)} < \frac{3}{4} + \gamma$, $v_i(P_k \setminus J) > \frac{1}{4} - \gamma$. $\square$

Lemma 5.7. For every agent i , there exists a partition $P = \{P_1, \dots, P_n\} \in \mathcal{P}_i^n(M \cup D)$ such that each $P_k \in P$ has at most one item j with $v_{ij} > \frac{5}{8} + \gamma$.

Proof. We show that if there are two items j_1, j_2 with each value more than $\frac{5}{8} + \gamma$ for an agent i in a bundle P_k of $P \in \mathcal{P}_i^n(M \cup D)$, then we can construct another $P' \in \mathcal{P}_i^n(M \cup D)$ where this is not true. Corollary 5.1 implies that there must exist another bundle $P_{k'}$ for which $\max_{j \in P_{k'}} v_{ij} < \frac{3}{8} + \frac{\gamma}{2}$. If there exists a partition $Q_{k'}^1$ and $Q_{k'}^2$ of items in $P_{k'}$ such that value of each $Q_{k'}^1$ and $Q_{k'}^2$ is at least $\frac{3}{8} - \gamma$, then we can rearrange items in $P_k \cup P_{k'}$ and make two new bundles $(\{j_1\} \cup Q_{k'}^1)$ and $(\{j_2\} \cup Q_{k'}^2)$. Clearly, the value of each bundle is at least 1 and each has exactly one item with value more than $\frac{5}{8} + \gamma$.

If no such $Q_{k'}^1$ and $Q_{k'}^2$ exists, then we claim that there exists a partition of $P_{k'}$ into three sets with each value less than $\frac{3}{8}$. We can find this partition as follows: Initialize three empty bundles, and repeatedly add the highest value item of $P_{k'}$ to the bundle with the lowest value. For a contradiction, suppose one of the three bundles has value more than $\frac{3}{8}$, then the sum of the values of the other two sets must be less than $\frac{3}{8}$ because otherwise, they make a partition of two with each value more than $\frac{3}{8} - \gamma$. This means that at least one of the bundles must have a value less than $\frac{3}{16}$. This implies that the value of the largest bag before adding the last item must be less than $\frac{3}{16}$ and the last item also should value less than $\frac{3}{16}$, which is a contradiction. Therefore, there exists a partition $Q_{k'}^1, Q_{k'}^2$ and $Q_{k'}^3$ of $P_{k'}$ such that each value less than $\frac{3}{8}$.

According to Lemma 5.6, there exists a bundle $P_{\hat{k}} \in \mathcal{P}_i^n(M)$ such that $v_i(P_{\hat{k}} \setminus J) > \frac{1}{4} - \gamma$. Let $v_i(P_{\hat{k}}) = 1 + \delta$ for some $\delta \geq 0$. Observe that $P_{k'}$ cannot be same as $P_{\hat{k}}$ and P_k , otherwise we would have made two bundles earlier, each with value at least 1 and exactly one item more than $\frac{5}{8} + \gamma$. We initialize three bags:

$$\text{bag 1 : } \{j_1\} \cup Q_{k'}^1 \quad \text{bag 2 : } \{j_2\} \cup Q_{k'}^2 \quad \text{bag 3 : } (P_k \setminus \{j_1, j_2\}) \cup (P_{\hat{k}} \cap J) \cup Q_{k'}^3$$

Observe that the value of each of bag 1 and bag 2 is at most $\frac{9}{8} + \frac{3\gamma}{2}$, and the total value of all items in $P_k, P_{k'}$ and $P_{\hat{k}}$ is at least $3 + \frac{2}{8} + \delta + 2\gamma$. We sort the remaining items in decreasing order and add them one by one to a bag with the lowest value. Since the value of the last item added to each bag has a value of at most $\frac{1}{8}$ (from Lemma 5.4), each bag has a value of at least 1 and it has at most one item with value more than $\frac{5}{8} + \gamma$. We repeat this process for all bundles with two item of value more than $\frac{5}{8} + \gamma$ to find a desired partition. $\square$

Lemma 5.10. For agent $a \in N^2$, if $t_1 = 0$ then we have

1. $x_a \leq (\frac{3}{4} + \gamma - 2y)l_a$
2. $v_a((M \cup D) \setminus J) \geq \max\{x_a + l_a(\frac{1}{4} - \gamma) - yz + z(\frac{1}{4} - \gamma), z(\frac{1}{4} - \gamma)\}$, where $z = \max\{2h_a - |\bigcup_{t=t_1+1}^{t_2} P_t \cap J|, 0\}$.

Proof. For the first part, since $t_1 = 0$, we have $v_a(B_k) \geq 2y, \forall B_k \in L_a$ (Lemma 5.9). This implies that $x_a = (\frac{3}{4} + \gamma)l_a - \sum_{k: B_k \in L_a} B_k \leq (\frac{3}{4} + \gamma - 2y)l_a$.

For the second part, in $\bigcup_{k: B_k \in H_a} B_k$ there are exactly $2h_a$ items from J . If there are at least $2h_a$ items in $\bigcup_{t=t_1+1}^{t_2} P_t \cap J$ then $z = 0$. On the other hand, if there are $2h_a - z$ items in $\bigcup_{t=t_1+1}^{t_2} P_t \cap J$, then it means that there are at least z bundles in $P_{t_1+1}, \dots, P_{t_2}$ with exactly one item from J . Each of these bundles need more than $\frac{1}{4} - \gamma$ value of items from $M \setminus J$ to become 1.

Next, if $z = 0$, then all bundles in $\{P_{t_2+1}, \dots, P_{t_3}\}$ need at least $x_a + l_a(\frac{1}{4} - \gamma)$ value of items from $M \setminus J$ to become 1 because the value of each item in the bundles of L_a is at least y (follows from Definition 5.8 and the construction of B_k 's in (3)). If $z > 0$, then all bundles in $\{P_{t_2+1}, \dots, P_{t_3}\}$ need at least $x_a + l_a(\frac{1}{4} - \gamma) - yz$ value of items from $M \setminus J$ to become one because each of the z items has value at most y (Lemma 5.9). Therefore, in total, there should be at least $x_a + l_a(\frac{1}{4} - \gamma) - yz + z(\frac{1}{4} - \gamma)$ value of items from $M \setminus J$ in $\{P_{t_1+1}, \dots, P_{t_3}\}$.

Further, in case z is large such that the value of $x_a + l_a(\frac{1}{4} - \gamma) - yz + z(\frac{1}{4} - \gamma)$ is negative, we still need $z(\frac{1}{4} - \gamma)$ value of items from $M \setminus J$ to make all bundles in $\{P_{t_1+1}, \dots, P_{t_2}\}$ value 1. $\square$

Theorem 5.11. For agent $a \in N^2$, if $t_1 = 0$ then

$$v_a(M \cup D \setminus J) \geq x_a + \frac{l_a}{8}. \quad (17)$$

Proof. Using Lemma 5.10(2) there are two cases for the maximum of $x_a + l_a(\frac{1}{4} - \gamma) - yz + z(\frac{1}{4} - \gamma)$ and $z(\frac{1}{4} - \gamma)$. We prove the claim for each case separately.

Case 1: Suppose $x_a + l_a(\frac{1}{4} - \gamma) - yz + z(\frac{1}{4} - \gamma) \geq z(\frac{1}{4} - \gamma)$, then $z \leq \frac{x_a + l_a(\frac{1}{4} - \gamma)}{y}$. Further, Lemma 5.10(1) implies $y \leq \frac{l_a(\frac{3}{4} + \gamma) - x_a}{2l_a}$. Using these, we get

$$\begin{aligned} v_a(M \cup D \setminus J) &\geq x_a + l_a(\frac{1}{4} - \gamma) - yz + z(\frac{1}{4} - \gamma) \\ &\geq \frac{x_a(\frac{1}{4} - \gamma) + l_a(\frac{1}{4} - \gamma)^2}{y} \\ &\geq \frac{2x_al_a(\frac{1}{4} - \gamma) + 2l_a^2(\frac{1}{4} - \gamma)^2}{l_a(\frac{3}{4} + \gamma) - x_a}. \end{aligned}$$

We need to show that

$$\frac{2x_al_a(\frac{1}{4} - \gamma) + 2l_a^2(\frac{1}{4} - \gamma)^2}{l_a(\frac{3}{4} + \gamma) - x_a} - \left(x_a + \frac{l_a}{8}\right) \geq 0. \quad (\text{A.1})$$

For $n \leq 4$ (i.e., number of agents in the original instance is at most 4), there is a simpler way to prove this claim. However, there are better approximation factors available for them in any case [7], so we assume that $n > 4$. Since $\gamma = \frac{1}{12n}$, it decreases as n increases. Therefore, it is enough to show (A.1) for $n = 5$. For this, we put $\gamma = \frac{1}{60}$ in (A.1) and get,

$$\begin{aligned} \frac{2x_al_a(\frac{1}{4} - \frac{1}{60}) + 2l_a^2(\frac{1}{4} - \frac{1}{60})^2}{l_a(\frac{3}{4} + \frac{1}{60}) - x_a} - \left(x_a + \frac{l_a}{8}\right) &= \frac{7(30x_al_a + 7l_a^2)}{15(23l_a - 30x_a)} - \frac{8x_a + l_a}{8} \\ &= \frac{3600x_a^2 - 630x_al_a + 47l_a^2}{120(23l_a - 30x_a)} \\ &> \frac{36(10x_a - l_a)^2 + l_a(11l_a + 90x_a)}{120(23l_a - 30x_a)} \\ &\geq 0. \end{aligned}$$

The last inequality follows because the denominator is positive using the definition of x_a .

Case 2: Suppose $x_a + l_a(\frac{1}{4} - \gamma) - yz + z(\frac{1}{4} - \gamma) \leq z(\frac{1}{4} - \gamma)$. Then, using Lemma 5.10(1), we have

$$v_a((M \cup D) \setminus J) \geq z(\frac{1}{4} - \gamma) \geq \frac{x_a(\frac{1}{4} - \gamma) + l_a(\frac{1}{4} - \gamma)^2}{y} \geq x_a + \frac{l_a}{8},$$

where the last inequality follows from the Case 1. $\square$

Theorem 5.12. For agent $a \in N^2$, we have $v_a((M \cup D) \setminus J) \geq x_a + \frac{l_a}{8}$.

Proof. Let $J_{\min} = \bigcup_{B_k \in L_a^1} B_k \cap J_2$. Then, we have $v_{aj} < \frac{1}{8}, \forall j \in J_{\min}$ because these items are bagged with an item value more than $\frac{5}{8} + \gamma$ and together they have value less than $\frac{3}{4} + \gamma$. Therefore, $v_a(J_{\min}) < \frac{t_1}{8}$. Let $x'_a := (\frac{3}{4} + \gamma)t_1 - \sum_{B_k \in L_a^1} v_a(B_k)$, and $x''_a := x_a - x'_a$. For agent a , we can treat items of $J_{\min}$ as low-value items so we will prove:

$$v_a((M \cup D) \setminus J) + v_a(J_{\min}) \geq x'_a + \frac{t_1}{4} + x''_a + (l_a - t_1)/8. \quad (\text{A.2})$$

Since $v_a(J_{\min})$ is at most $\frac{t_1}{8}$, (A.2) directly implies the theorem. Let us call items in $(M \setminus J) \cup J_{\min} \cup D$ as *filler* items (each has value $< \frac{1}{8}$) and items in $J \setminus J_{\min}$ as *base* items (each has value $> \frac{1}{4} + \frac{\gamma}{3}$).

In re-enumerated $\mathcal{P}_a^n(M \cup D)$, recall that each of $\{P_1, \dots, P_{t_1}\}$ contains a base item whose value is at least the maximum of all the items in the remaining bundles. Further, if there are two base items in any bundle P_k in $\{P_1, \dots, P_{t_1}\}$, then $v_a(P_k) > 1$. If each bundle in $\{P_1, \dots, P_{t_1}\}$ has exactly one base item (t_1 bundles in total) then items of value at least $x'_a + t_1(\frac{1}{4} - \gamma)$ are needed to make all these bundles at least 1 and there are t_1 items in D_3 with value γ . This, together with Theorem 5.11, proves the bound. On the other hand, if there are more than t_1 base items in $\{P_1, \dots, P_{t_1}\}$, then for each extra base item j , there are two cases. If j comes from $B_k \in L_a^2$ then clearly, x'_a will decrease by less than v_{aj} but x''_a will increase by v_{aj} , so the bound only improves. For the other case, if j comes from $B_k \in H_a$ then it will make $B_k \setminus \{j\}$ to need more low-value items than the bundle in $\{P_1, \dots, P_{t_1}\}$ who gets j to become one, so x'_a will only increase and x''_a stays same, and hence the bound only improves. This completes the proof. $\square$

References

- [1] A.D. Procaccia, J. Wang, Fair enough: guaranteeing approximate maximin shares, in: Proc. 15th Conf. Economics and Computation, EC, 2014, pp. 675–692.
- [2] D. Kurokawa, A.D. Procaccia, J. Wang, When can the maximin share guarantee be guaranteed?, in: Proc. 30th Conf. Artif. Intell., AAAI, 2016, pp. 523–529.
- [3] D. Kurokawa, A.D. Procaccia, J. Wang, Fair enough: guaranteeing approximate maximin shares, *J. ACM* 65 (2) (2018) 8:1–8:27.
- [4] G. Amanatidis, E. Markakis, A. Nikzad, A. Saberi, Approximation algorithms for computing maximin share allocations, *ACM Trans. Algorithms* 13 (4) (2017) 52:1–52:28.
- [5] S. Barman, S.K. Krishnamurthy, Approximation algorithms for maximin fair division, in: Proc. 18th Conf. Economics and Computation, EC, 2017, pp. 647–664.
- [6] J. Garg, P. McGlaughlin, S. Taki, Approximating maximin share allocations, in: 2nd Symposium on Simplicity in Algorithms, SOSA@SODA 2019, 2019, pp. 20:1–20:11.
- [7] M. Ghodsi, M. Hajiaghayi, M. Seddighin, S. Seddighin, H. Yami, Fair allocation of indivisible goods: improvements and generalizations, in: Proc. 19th Conf. Economics and Computation, EC, 2018, pp. 539–556.
- [8] J. Garg, S. Taki, An improved approximation algorithm for maximin shares, in: Proc. 21st Conf. Economics and Computation, EC, 2020, pp. 379–380.
- [9] H. Steinhaus, The problem of fair division, *Econometrica* 16 (1948) 101–104.
- [10] E. Budish, The combinatorial assignment problem: approximate competitive equilibrium from equal incomes, *J. Polit. Econ.* 119 (6) (2011) 1061–1103.
- [11] A. Farhadi, M. Ghodsi, M.T. Hajiaghayi, S. Lahaie, D.M. Pennock, M. Seddighin, S. Seddighin, H. Yami, Fair allocation of indivisible goods to asymmetric agents, *J. Artif. Intell. Res.* 64 (2019) 1–20.
- [12] V. Gates, T.L. Griffiths, A.D. Dragan, How to be helpful to multiple people at once, *Cogn. Sci.* 44 (2020) e12841.
- [13] G.J. Woeginger, A polynomial-time approximation scheme for maximizing the minimum machine completion time, *Oper. Res. Lett.* 20 (4) (1997) 149–154.
- [14] 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.
- [15] L. Gourvès, J. Monnot, On maximin share allocations in matroids, *Theor. Comput. Sci.* 754 (2019) 50–64.
- [16] S. Barman, A. Biswas, S.K.K. Murthy, Y. Narahari, Groupwise maximin fair allocation of indivisible goods, in: Proc. 32nd Conf. Artif. Intell., AAAI, 2018, pp. 917–924.
- [17] B.R. Chaudhury, T. Kavitha, K. Mehlhorn, A. Sgouritsa, A little charity guarantees almost envy-freeness, in: Proc. 31st Symp. Discrete Algorithms, SODA, 2020, pp. 2658–2672.
- [18] Z. Li, A. Vetta, The fair division of hereditary set systems, in: Proc. 14th Conf. Web and Internet Economics, WINE, 2018, pp. 297–311.
- [19] A. Biswas, S. Barman, Fair division under cardinality constraints, in: Proc. 27th Intl. Joint Conf. Artif. Intell., IJCAI, 2018, pp. 91–97.
- [20] S. Brânzei, T.P. Michalak, T. Rahwan, K. Larson, N.R. Jennings, Matchings with externalities and attitudes, in: Proc. 12th Conf. Auton. Agents and Multi-Agent Systems, AAMAS, 2013, pp. 295–302.
- [21] N. AhmadiPourAnari, S. Ehsani, M. Ghodsi, N. Haghpahan, N. Immorlica, H. Mahini, V.S. Mirrokni, Equilibrium pricing with positive externalities, *Theor. Comput. Sci.* 476 (2013) 1–15.
- [22] X. Bei, A. Igarashi, X. Lu, W. Suksompong, Connected fair allocation of indivisible goods, *CoRR arXiv:1908.05433*.
- [23] Z. Lonc, M. Truszczynski, Maximin share allocations on cycles, *CoRR arXiv:1905.03038*.
- [24] H. Aziz, G. Rauchecker, G. Schryen, T. Walsh, Algorithms for max-min share fair allocation of indivisible chores, in: Proc. 31st Conf. Artif. Intell., AAAI, 2017, pp. 335–341.
- [25] X. Huang, P. Lu, An algorithmic framework for approximating maximin share allocation of chores, *CoRR arXiv:1907.04505*.
- [26] S. Barman, G. Ghalme, S. Jain, P. Kulkarni, S. Narang, Fair division of indivisible goods among strategic agents, in: Proc. 18th Conf. Auton. Agents and Multi-Agent Systems, AAMAS, 2019, pp. 1811–1813.
- [27] G. Amanatidis, G. Birmas, E. Markakis, On truthful mechanisms for maximin share allocations, in: S. Kambhampati (Ed.), Proc. 25th Intl. Joint Conf. Artif. Intell., IJCAI, 2016, pp. 31–37.
- [28] G. Amanatidis, G. Birmas, G. Christodoulou, E. Markakis, Truthful allocation mechanisms without payments: characterization and implications on fairness, in: Proc. 18th Conf. Economics and Computation, EC, 2017, pp. 545–562.
- [29] H. Aziz, B. Li, X. Wu, Strategyproof and approximately maxmin fair share allocation of chores, in: Proc. 28th Intl. Joint Conf. Artif. Intell., IJCAI, 2019, pp. 60–66.
- [30] S. Bouveret, M. Lemaître, Efficiency and sequenceability in fair division of indivisible goods with additive preferences, *CoRR arXiv:1604.01734*.