



# The budgeted maximin share allocation problem

Bin Deng<sup>1</sup> · Weidong Li<sup>1</sup>

Received: 7 September 2023 / Accepted: 11 August 2024

© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2024

## Abstract

We are given a set of indivisible goods and a set of  $m$  agents where each good has a size and each agent has an additive valuation function and a budget. The budgeted maximin share allocation problem is to find a feasible allocation such that the size of the bundle allocated to each agent does not exceed its budget, and the minimum ratio of the valuation and the maximin share (MMS) value of any agent is as large as possible, where the MMS value of each agent is that he can achieve by dividing the goods into  $n$  bundles, and receiving his least desirable bundle. In this paper, we prove the existence of  $\frac{n}{3n-2}$ -approximate MMS allocation and give an instance which does not have a  $(\frac{3}{4} + \epsilon)$ -approximate MMS allocation, for any  $\epsilon \in (0, 1)$ . Moreover, we provide a polynomial time algorithm to find an  $\frac{1}{3}$ -MMS allocation, and prove that there is no  $(\frac{2}{3} + \epsilon)$ -approximate algorithm in polynomial time unless  $\mathcal{P} = \mathcal{NP}$ .

**Keywords** Fair allocation · Maximin-share · Budget constraints

## 1 Introduction

The fair allocation of indivisible goods has become an increasingly hot topic in both mathematics and economics [4, 23]. One of the most representative fair allocation problems is the max-min allocation problem [10], which assigns  $m$  indivisible goods to  $n$  heterogeneous agents such that the minimum utility of the agents is maximized, where the utility of an agent is the sum of its utilities for the items it receives. After nearly two decades of in-depth research, the best possible approximation algorithm is still far from resolved [26].

---

✉ Weidong Li  
weidongmath@126.com

Bin Deng  
dengbin96@126.com

<sup>1</sup> School of Mathematics and Statistics, Yunnan University, Kunming 650504, Yunnan, People's Republic of China

Budish [12] introduced a novel notion of fair allocation, called maximin share (MMS), where the MMS value of an agent is her best guarantee if she is to partition the goods into  $n$  bundles but receives the worst one. Kurokawa et al. [27] conducted in-depth research on maximin share, and since then, maximin share has attracted much attention from theoretical computer science and mathematics. They also proved that no algorithm can find a maximin share allocation for indivisible goods. Therefore, they turned their attention to the approximation guarantee of maximin share. They proved that a  $\frac{2}{3}$ -MMS allocation always exists. Therefore, improving the approximate guarantee of maximin share is a direct research direction in later research [1–3, 5, 7, 18, 22]. So far, for the additive valuations, the best approximation guarantee is  $\frac{3}{4} + \frac{3}{3836}$  [3] and the best inapproximation result is  $\frac{39}{40}$  [16]. Finding the best possible  $\alpha$ -approximate MMS allocation for any instance is still open [4].

In a real environment, each agent may have additional constraints, such as connectivity, cardinality, matroid, geometric, separation, budget, and conflict constraints [29]. In a company, when assigning goods (tasks) to agents (employees), each good has a size (working time) and each agent has a budget (total processing time) constraint. As the maximin share allocation [12], it is desirable to find the best possible  $\alpha$ -approximate MMS allocation to ensure agent satisfaction. This motivates us to study the budgeted maximin share problem, where we are given a set of  $m$  indivisible goods and a set of  $n$  agents. Each good  $g$  has a size  $s(g)$ , and each agent  $i$  has an additive valuation function  $v$  and a budget  $b_i$ . The budgeted maximin share allocation problem is to find a feasible allocation such that the bundle allocated to each agent  $i$  does not exceed its budget  $b_i$ , and the minimum ratio of the valuation and the maximin share (MMS) value of any agent is as large as possible, where MMS value of each agent is that he can achieve by dividing the goods into  $n$  bundles, and receiving his least desirable bundle.

Somewhat surprisingly, both the budgeted maximin share allocation problem and the budgeted max-min allocation problem (which can be defined similarly) has not been studied before. Obviously, the budgeted maximin share allocation problem generalizes the maximin share allocation [27] and the maximin share allocation under cardinality constraints [11]. However, classical methods, including bag-filling and round-robin methods [4], cannot be directly applied to budget constraints. In this paper, we propose some existence results and present a polynomial time approximation algorithm based on the modified bag-filling algorithm. The main results can be seen in Table 1.

The structure of the paper is as follows. In Sect. 2, we introduce the related works in detail. In Sect. 3, we introduce the symbols and related concepts. In Sect. 4, we give the existence results of approximate MMS allocation. In Sect. 5,

**Table 1** Our results

	Lower bound	Upper bound
Existence	$\frac{n}{3n-2}$ (Corollary 2)	$\frac{3}{4}$ (Theorem 1)
Polynomial	$\frac{1}{3}$ (Theorem 5)	$\frac{2}{3} + \epsilon$ (Theorem 4)

we present a polynomial time algorithm to find  $\frac{1}{3}$ -MMS allocation. Finally, we draw conclusions and discuss future research directions.

## 2 Related works

If  $b_i = +\infty$  for each agent  $i$ , the budgeted maximin share allocation problem is exactly the maximin share allocation [12]. Kurokawa et al. [27] demonstrated that exactly MMS may not exist, but  $\frac{2}{3}$ -approximate MMS is guaranteed to exist. Amanatidis et al. [5] proposed a  $(\frac{2}{3} - \epsilon)$ -approximation guaranteed polynomial time algorithm. Barman et al. [7] proposed a  $\frac{2}{3}$ -approximation guaranteed polynomial time algorithm. Ghodsi et al. [22] proposed a  $(\frac{3}{4} - \epsilon)$ -approximation guaranteed polynomial time algorithm. Garg et al. [18] proposed a  $(\frac{3}{4} + o(1))$ -approximation guaranteed polynomial time algorithm. Recently, Akrami et al. [1, 3] improved the existence guarantee to  $(\frac{3}{4} + \min\{\frac{1}{36}, \frac{3}{16n-4}\})$  and  $(\frac{3}{4} + \frac{3}{3836})$ . Furthermore, Akrami et al. [2] proved the existence of a randomized allocation such that each agent obtains  $\frac{3}{4}$  her MMS value (ex-post) and  $(17\sqrt{3} - 24)/4\sqrt{3}$  her MMS value (ex-ante).

If  $s(g) = 1$  for each good  $g$ , the budgeted maximin share allocation problem is exactly the maximin share allocation under cardinality constraints. Biswas et al. [11] demonstrated that  $\frac{1}{3}$ -approximate MMS can guarantee its existence. Hummel et al. [24] improved the existence guarantee to  $\frac{2}{3}$  and proposed a  $\frac{2}{3}$ -approximation guaranteed polynomial time algorithm.

The budgeted maximin share allocation problem is closely related to the max-min allocation problem [10], which assigns  $m$  indivisible goods to  $n$  heterogeneous agents such that the minimum utility of the agents is maximized. Asadpour and Saberi [6] presented an  $\Omega(\frac{1}{\sqrt{n \log^3 n}})$ -approximation algorithm based an iterative method for rounding a fractional matching on a tree. Saha and Srinivasan [28] designed an  $\Omega(\frac{1}{\sqrt{n \log n}})$ -approximation algorithm by using a rounding method based on random walks in polytopes. Chakrabarty et al. [14] designed an  $\Omega(\frac{1}{m^c \log m})$ -approximation algorithm in  $n^{O(\frac{1}{\epsilon})}$  time for any  $\epsilon \geq \frac{9 \log \log m}{\log m}$ . Recently, Ko et al. [26] designed an  $\frac{1}{c}$ -approximation algorithm where  $c$  is the maximum ratio of the largest utility to the smallest utility of any goods. In addition, Caprara et al. [13] designed an  $\frac{2}{3}$ -approximation algorithm under positive integer capacity for the max-min allocation problem.

The budgeted maximin share allocation problem is also closely related to relaxed envy free allocation with budget constraints. Barman et al. [9] studied EFX, which is a stronger concept than EF1, and demonstrated that EFX allocation always exists for generalized assignment constraints. They also proposed a pseudo polynomial time algorithm to calculate EFX and a fully polynomial-time approximation scheme to calculate EFX allocation. Wu et al. [31] demonstrated

that the maximum Nash social welfare distribution is  $\frac{1}{4}$ -EF1 and PO. Gan et al. [21] proposed a polynomial time algorithm to find the a  $\frac{1}{2}$ -approximate EF1 allocation with identical additive valuations. Barman et al. [8] proposed a polynomial time algorithm to find the EF2 allocation with identical additive valuations. Garbea et al. [19] studied the EFX allocation of high Nash social welfare for two agents and three agents.

### 3 Preliminaries

For convenience, let  $[k] = \{1, 2, \dots, k\}$  for any positive integer  $k$ . We are given an instance  $I = (N, G, v_i, s, b)$ , where  $N = [n]$  is the set of agents,  $G = \{g_1, g_2, \dots, g_m\}$  is the set of indivisible goods,  $v_i : G \mapsto \mathbb{R}^+$  is the value function of agent  $i \in N$ ,  $s : G \mapsto \mathbb{R}^+$  is the size function of goods, and  $b : N \mapsto \mathbb{R}^+$  is the budget function of agents.

For any subset  $K \subseteq G$ , let  $s(K) = \sum_{g: g \in K} s(g)$  be the total size of the goods in  $K$ , and  $v_i(K) = \sum_{g: g \in K} v_i(g)$  be the total value of the goods in  $K$  for agent  $i$ . An  $n$ -partial-partition  $\mathbf{P}^{(i)} = (P_1^{(i)}, \dots, P_n^{(i)})$  of  $G$  is called feasible for agent  $i \in N$ , if  $\cup_{j=1}^n P_j^{(i)} \subseteq G$ ,  $s(P_j^{(i)}) \leq b_i$  for any  $j \in [n]$ , and

$$P_j^{(i)} \cap P_k^{(i)} = \emptyset, \text{ for any } j, k \in [n] \text{ such that } j \neq k,$$

where  $b_i$  is the budget of agent  $i$ . Let  $\mathcal{P}^{(i)}(G, n)$  be the set of all feasible  $n$ -partial-partitions of goods  $G$  for agent  $i$ . An important fairness criterion, called maximin share (MMS), is first proposed by Budish [12], which is defined as follows.

**Definition 1** [12] The maximin share (MMS) value of agent  $i \in N$  for allocating the goods in  $G$  is defined as

$$MMS_i(G, N) = \max_{\mathbf{P}^{(i)} \in \mathcal{P}^{(i)}(G, n)} \min_{j \in [n]} v_i(P_j^{(i)}).$$

An allocation  $\mathbf{A} = (A_1, A_2, \dots, A_n)$  is called feasible, if  $\cup_{i=1}^n A_i \subseteq G$ ,  $s(A_i) \leq b_i$  for any  $i \in N$ , and  $A_i \cap A_j = \emptyset$  for any  $i, j \in N$  with  $i \neq j$ , where  $A_i$  is the bundle of goods assigned to agent  $i \in N$ .

**Definition 2** ( $\alpha$ -MMS). For  $0 < \alpha \leq 1$ , a feasible allocation  $\mathbf{A}$  is called  $\alpha$ -approximate MMS if  $v_i(A_i) \geq \alpha \cdot MMS_i(G, N)$  for any agent  $i \in N$ .

Without causing confusion later, we will abbreviate  $MMS_i(G, N)$  as  $MMS_i$ . The budgeted maximin share allocation problem is to find an  $\alpha$ -approximate MMS allocation, such that  $\alpha$  is as large as possible.

The user's MMS value has scale invariance, that is, if the value of an agent is multiplied by a factor, then the agent's MMS value is also scaled by the same factor.

**Lemma 1** (Scale invariance). [5, 17] *Given an instance  $I = (N, G, v_i, s, b)$  and positive scale factors  $d_i$ , let  $I' = (N, G, v'_i, s, b)$  be a scaled instance such that*

$$v'_i(g) = \frac{v_i(g)}{d_i}, \text{ for any good } g \in G \text{ and any agent } i \in N,$$

*Then,  $MMS'_i = \frac{MMS_i}{d_i}$  for any agent  $i \in [n]$ .*

Similar to [4, 5], we have

**Lemma 2** (Monotonicity). *For any  $j \in N$  and  $g \in G$ ,  $MMS_i(G \setminus \{g\}, N \setminus \{j\}) \geq MMS_i(G, N)$  holds for any agent  $i \in N \setminus \{j\}$ .*

### 4 The existence of approximate MMS allocation

As a negative side, we can easily obtain the following results.

**Theorem 1** *For any  $\epsilon > 0$ , there exists an instance for the budgeted maximin share allocation problem that does not have a  $(\frac{3}{4} + \epsilon)$ -MMS.*

**Proof** Consider an instance with  $N = \{1, 2\}$  and  $G = \{g_1, g_2, g_3, g_4, g_5\}$ . The parameters of these agents and goods are shown in Table 2.

The budgets of the two agents are  $b_1 = 11$  and  $b_2 = 20$ . Obviously, the 2-partial-partition  $\mathbf{P}^{(i)} = (\{g_1, g_3\}, \{g_2, g_5\})$  is feasible for agent 1 and  $\min\{v_1(\{g_1, g_3\}), v_1(\{g_2, g_5\})\} = 1$ , implying that  $MMS_1 \geq 1$ . Since  $v_1(G) = 2$ , we have  $MMS_1 = 1$ . Similarly, we can obtain  $MMS_2 = 1$ . If  $v_1(A_1) > (\frac{3}{4} + \epsilon)MMS_1$ , we have  $A_1 \in \{\{g_1, g_3\}, \{g_2, g_5\}, \{g_1, g_5\}\}$ . It is easy to verify that  $\frac{v_2(A_2)}{MMS_2} \leq \frac{3}{4}$  in any case, which implies that the theorem holds.  $\square$

For a subset  $K \subseteq G$  of goods and an agent  $i \in N$ , construct an instance  $(K, i)$  for the classical knapsack problem [15], whose objective is to find a subset  $K_i \subseteq K$ , such that  $v_i(K_i)$  is maximized under the constraint  $s(K_i) \leq b_i$ . It is well-known that the knapsack problem is NP-hard and admits a fully polynomial time approximation scheme (FPTAS) [15]. Let  $K_i = \mathcal{A}(K, i) \subseteq K$  be subset of items produced by an  $\alpha$ -approximation algorithm  $\mathcal{A}$  for the instance  $(K, i)$  of the knapsack problem, implying that

**Table 2** The parameters of agents and goods

	$g_1$	$g_2$	$g_3$	$g_4$	$g_5$
Agent 1	$\frac{3}{4}$	$\frac{1}{2}$	$\frac{1}{4}$	0	$\frac{1}{2}$
Agent 2	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{4}$
Sizes	9	9	2	16	2

$$v_i(K_i) = v_i(\mathcal{A}(K, i)) \geq \alpha \cdot v_i(OPT(K, i)), \tag{1}$$

where  $OPT(K, i) = \arg \max_{S \subseteq K: s(S) \leq b_i} v_i(S)$  is the optimal solution for instance  $(K, i)$  of the knapsack problem and  $\alpha \in (0, 1]$ .

Assume that the guessed MMS values are  $(d_1, d_2, \dots, d_n)$ . For convenience, let

$$\beta = \frac{\alpha}{1 + 2\alpha - \frac{2\alpha}{n}} \in \left(0, \frac{1}{2}\right] \tag{2}$$

for  $n \geq 2$ . Given an instance  $I = (N, G, v_i, s, b)$ , a good  $g \in G$  obtained by scaling the users' valuation functions (i.e.,  $v'_i = \frac{v_i}{d_i}$ ) is called 'large' for agent  $i \in N$  if  $v'_i(g) \geq \beta$  and  $s(g) \leq b_i$ . Our algorithm to find a  $\beta$ -approximate MMS allocation works as follows. For each agent  $i \in N$ , if there exists a large good, we can assign  $g$  to agent  $i$  by setting  $A_i = \{g\}$ . By monotonicity, we only need to find a feasible allocation such that each agent receives a bundle with a value of at least  $\beta$  for  $v'$  for the remaining agents. Then, let  $RG_t (RN_t)$  be the set of remaining goods (agents) at the beginning of any iteration  $t$ , where  $t \in \{0, \dots, n\}$ . First, find the set  $RN'_t = \{i \in RN_t : v'_i(\mathcal{A}(RG_t, i)) \geq \beta\}$  by using the  $\alpha$ -approximation algorithm  $\mathcal{A}$  for the classical knapsack problem if there is an agent  $i \in RN_t$  such that  $v'_i(\mathcal{A}(RG_t, i)) \geq \beta$ . Let  $K_{i_t} = \mathcal{A}(RG_t, i_t)$  where  $i_t \in RN'_t$  is the agent with the smallest budget. Then, select goods from the set  $K_{i_t}$  and continuously add them to bag  $B$  until agent  $j_t$  believes that the valuation of bag  $B$  is at least  $\beta$ . Finally, assign  $B$  to any agent  $j_t$  such that  $v'_{j_t}(B) \geq \beta$ .

**Algorithm 1** Find a feasible allocation depending on  $(d_1, d_2, \dots, d_n)$

---

```

Input : An instance  $I = (N, G, v_i, s, b)$ ,  $(d_1, d_2, \dots, d_n)$ , and threshold  $\beta$ .
Output:  $\beta$ -MMS allocation  $\mathbf{A}$ .
1 Let  $v'_i(g) = \frac{v_i(g)}{d_i}$  for any  $i \in N$  and  $g \in G$ ;
2 Let  $A_i = \emptyset$  for each agent  $i \in N$ ;
3 while  $v'_i(g) \geq \beta$  and  $s(g) \leq b_i$  for some agent  $i \in N$  and good  $g \in G$  do
4    $\lfloor$  Set  $A_i = \{g\}$ ,  $N \leftarrow N \setminus \{i\}$  and  $G \leftarrow G \setminus A_i$ ;
5 Let  $RG_0 \leftarrow G$ ,  $RN_0 \leftarrow N$ , and  $t = 0$ ;
6 while  $RG_t \neq \emptyset$  and  $RN_t \neq \emptyset$  do
7   if  $v'_i(\mathcal{A}(RG_t, i)) < \beta$  for any agent  $i \in RN_t$  then
8      $\lfloor$  stop;
9   else
10     $\lfloor$  Set  $RN'_t = \{i \in RN_t : v'_i(\mathcal{A}(RG_t, i)) \geq \beta\}$  and  $i_t = \arg \min_{i \in RN'_t} b_i$ ;
11    Set  $B = \emptyset$  and  $K_{i_t} = \mathcal{A}(RG_t, i_t)$ ;
12    while  $v'_{j_t}(B) < \beta$  for each agent  $j \in RN_t$  and  $K_{i_t} \neq \emptyset$  do
13       $\lfloor$  Let  $\bar{B} \leftarrow B \cup \{g\}$  where  $g$  is an arbitrary good in  $K_{i_t}$ , and
14         $\lfloor$   $K_{i_t} \leftarrow K_{i_t} \setminus \{g\}$ ;
15      Set  $A_{j_t} = \bar{B}$  where  $j_t$  is an arbitrary agent in  $RN_t$  such that  $v'_{j_t}(\bar{B}) \geq \beta$ ;
16      Set  $RN_{t+1} = RN_t \setminus \{j_t\}$ ,  $RG_{t+1} = RG_t \setminus A_{j_t}$  and  $t \leftarrow t + 1$ ;
16 Output  $(A_1, A_2, \dots, A_n)$ .

```

---

**Theorem 2** *In the allocation  $(A_1, A_2, \dots, A_n)$  produced by Algorithm 1, we have  $v_i(A_i) \geq \beta \cdot d_i$ , for each agent  $i \in N$  such that  $d_i \leq MMS_i$ .*

**Proof** Let  $RN = \{i | A_i = \emptyset\}$  be the remaining agents and  $RG = RG_{|RN_0|-|RN|} = G \setminus (\cup_{i=1}^n A_i)$  be set of goods at the end of algorithm, as exactly one agent receives a bundle of goods at each iteration. Fix an agent  $i$  such that  $d_i \leq MMS_i$ . By contradiction, we assume that  $A_i = \emptyset$ , implying that  $i \in RN \neq \emptyset$  and

$$v'_j(\mathcal{A}(RG, j)) < \beta, \text{ for any agent } j \in RN, \tag{3}$$

following from the choice of Algorithm 1. By the definition of  $RG_0$ , we have

$$v'_i(g) < \beta \text{ or } s(g) > b_i, \text{ for any good } g \in RG_0.$$

By the definition of  $\mathcal{A}$  and  $MMS_i$ , we have

$$\begin{aligned} v'_i(\mathcal{A}(RG_0, i)) &\geq \alpha \cdot v'_i(OPT(RG_0, i)) \\ &\geq \alpha \cdot MMS'_i(RG_0, RN_0) \\ &= \alpha \cdot \frac{MMS_i(RG_0, RN_0)}{d_i} \\ &\geq \alpha \cdot \frac{MMS_i}{d_i} \\ &\geq \alpha \geq \beta, \end{aligned}$$

where  $MMS'_i(RG_0, RN_0)$  is the maximin share of agent  $i$  according to the valuation function  $v'_i$ , the first equality follows from Lemma 1, the third inequality follows from Lemma 2, and the fourth inequality follows from the assumption  $d_i \leq MMS_i$ .

For convenience, let  $\kappa = |RN_0|$ . Since  $RG_0 \supseteq RG_1 \supseteq \dots \supseteq RG_{|RN_0|-|RN|}$  by (3), there must exist an iteration

$$\tau < \kappa - 1$$

such that

$$v'_i(\mathcal{A}(RG_t, i)) \geq \beta \text{ for each } t \in \{0, 1, \dots, \tau\}, \text{ and } v'_i(\mathcal{A}(RG_{\tau+1}, i)) < \beta, \tag{4}$$

as  $|RN| \neq \emptyset$  otherwise it contradicts the assumption. At each iteration  $t \in \{0, 1, \dots, \tau\}$ , let  $g_t$  be the last good added to  $B = A_{j_t}$ , implying that

$$s(g_t) \leq b_i \leq b_{j_t}, v'_i(g_t) < \beta, \text{ and } v'_i(A_{j_t} \setminus \{g_t\}) < \beta.$$

Therefore, we have  $v'_i(A_{j_t}) = v'_i(A_{j_t} \setminus \{g_t\}) + v'_i(g_t) < 2\beta$ , and

$$\sum_{t=0}^{\tau} v'_i(A_{j_t}) < 2(\tau + 1)\beta. \tag{5}$$

As in the last section, let  $\mathbf{P}^{(i)} = (P_1^{(i)}, P_2^{(i)}, \dots, P_{\kappa}^{(i)})$  be a  $\kappa$ -partial-partition of  $RG_0$  achieving the maximin share of agent  $i$ , which means that

$$v_i(P_k^{(i)}) \geq MMS_i(RG_0, RN_0) \geq MMS_i \geq d_i, \text{ for } k = 1, 2, \dots, \kappa, \tag{6}$$

where the first inequality follows from the definition of  $MMS_i$ , the second inequality follows from Lemma 2, and the last inequality follows the assumption. Therefore, we have

$$\sum_{k=1}^{\kappa} v'_i(P_k^{(i)}) = \sum_{k=1}^{\kappa} \frac{v_i(P_k^{(i)})}{d_i} \geq \kappa. \tag{7}$$

Since  $RG_0 = \cup_{t=0}^{\tau} A_{j_t} \cup RG_{\tau+1}$  and  $A_{j_t}, RG_{\tau+1}$  are disjoint, we have

$$\begin{aligned} \sum_{k=1}^{\kappa} v'_i(P_k^{(i)} \cap RG_{\tau+1}) &= \sum_{k=1}^{\kappa} v'_i(P_k^{(i)} \cap RG_0) - \sum_{t=0}^{\tau} v'_i(P_k^{(i)} \cap A_{j_t}) \\ &\geq \sum_{k=1}^{\kappa} v'_i(P_k^{(i)}) - \sum_{t=0}^{\tau} v'_i(A_{j_t}) \\ &\geq \kappa - 2(\tau + 1)\beta \\ &\geq \kappa - 2(\kappa - 1)\beta, \end{aligned}$$

where the second inequality follows from (5) and (7), and the last inequality is due to  $\tau < \kappa - 1$ .

Therefore, there is a bundle  $P_k^{(i)}$  such that

$$\begin{aligned} v'_i(P_k^{(i)} \cap RG_{\tau+1}) &\geq 1 - 2\left(1 - \frac{1}{\kappa}\right)\beta \\ &\geq 1 - 2\beta + \frac{2\beta}{n}, \end{aligned}$$

where the last inequality is due to  $\kappa \leq n$ , and  $s(P_k^{(i)} \cap RG_{\tau+1}) \leq s(P_k^{(i)}) \leq b_i$ , which implies that  $OPT(P_k^{(i)} \cap RG_{\tau+1}, i) = P_k^{(i)} \cap RG_{\tau+1}$  and

$$\begin{aligned} v'_i(OPT(P_k^{(i)} \cap RG_{\tau+1}), i) &= v'_i(P_k^{(i)} \cap RG_{\tau+1}) \\ &\geq 1 - 2\beta + \frac{2\beta}{n}. \end{aligned}$$

Clearly, we have



$$\begin{aligned}
 v'_i(\mathcal{A}(RG_{\tau+1}, i)) &\geq v'_i(\mathcal{A}(P_k^{(i)} \cap RG_{\tau+1}), i) \\
 &\geq \alpha \cdot v'_i(OPT(P_k^{(i)} \cap RG_{\tau+1}), i) \\
 &\geq \alpha \cdot \left(1 - 2\beta + \frac{2\beta}{n}\right) \\
 &= \alpha \cdot \left(1 - \frac{2\alpha}{1 + 2\alpha - \frac{2\alpha}{n}} + \frac{\frac{2\alpha}{n}}{1 + 2\alpha - \frac{2\alpha}{n}}\right) \\
 &= \beta,
 \end{aligned}$$

where the second inequality follows from (1) and the first equality follows from (2), contradicting (4).

Therefore, we have  $A_i \neq \emptyset$ . From the choice of algorithm, we have  $v'_i(A_i) \geq \beta$ , implying that

$$v_i(A_i) = v'_i(A_i) \cdot d_i \geq \beta \cdot d_i,$$

implies that the theorem holds. □

If  $d_i$  is an accurate MMS value for any agent  $i \in N$ , i.e.,  $d_i = MMS_i$ , we have

**Corollary 1** *In the allocation  $(A_1, A_2, \dots, A_n)$  produced by Algorithm 1, we have  $v_i(A_i) \geq \beta \cdot MMS_i$  for any agent  $i \in N$ , where  $\beta = \frac{\alpha}{1 + 2\alpha - \frac{2\alpha}{n}}$ .*

**Theorem 3** *Algorithm 1 returns a  $\beta$ -MMS allocation and the ratio is tight, where  $\beta = \frac{\alpha}{1 + 2\alpha - \frac{2\alpha}{n}}$ .*

**Proof** According to Corollary 1, Algorithm 1 returns  $\beta$ -MMS allocation. Consider  $n$  agents,  $n - 1 + \lceil \frac{1}{\alpha} + 2 - \frac{2}{n} \rceil$  indivisible goods and a small enough number  $\epsilon > 0$ . The parameters of these agents and goods are as in Table 3.

The budget for each agent  $i \in N$  is  $b_i = 1$ . Obviously, partition  $(\{g_1\}, \{g_2\}, \dots, \{g_{n-1}\}, \{g_n, \dots, g_{n-1+\lceil \frac{1}{\alpha} + 2 - \frac{2}{n} \rceil}\})$  is feasible  $n$ -partial-partition for all agent. Since  $\min\{v_i(\{g_1\}), v_i(\{g_2\}), \dots, v_i(\{g_{n-1}\}), v_i(\{g_n, \dots, g_{n-1+\lceil \frac{1}{\alpha} + 2 - \frac{2}{n} \rceil}\})\} = 1$ , we have  $MMS_i \geq 1$ . In fact, no other feasible  $n$ -partial-partition provides a larger MMS

**Table 3** The parameters of agents and goods

	$g_1$	...	$g_2$	$g_{n-1}$	$g_n$	...	$g_{n-1+\lceil \frac{1}{\alpha} + 2 - \frac{2}{n} \rceil}$
Agent 1	1	1	1	1	$\beta$	$\beta$	$\beta$
Agent 2	1	1	1	1	$\beta$	$\beta$	$\beta$
⋮							
Agent $n$	1	1	1	1	$\beta$	$\beta$	$\beta$
Sizes	1	1	1	1	$\epsilon$	$\epsilon$	$\epsilon$

value, i.e.,  $MMS_1 = 1$ . We can verify that when  $\epsilon$  is small enough, Algorithm 1 produces a feasible allocation  $(A_1, A_2, \dots, A_n)$ , where at least one user  $i$  receives one good from  $\{g_n, \dots, g_{n-1+\lceil \frac{1}{\alpha} + 2 - \frac{2}{n} \rceil}\}$ . Obviously, we have  $v_i(A_i) = \beta = \beta \cdot MMS_i$ , implying that the ratio is tight.  $\square$

If  $\mathcal{A}$  is an optimal algorithm for the classical knapsack problem, i.e.,  $\alpha = 1$ , we have

**Corollary 2** *There exists a  $\frac{n}{3n-2}$ -MMS allocation for every instance  $I$  of the budgeted maximin share problem.*

### 5 Polynomial-time algorithm

Based on the information of  $MMS_i$ , we have proved the existence of  $\frac{n}{3n-2}$ -approximate MMS allocation for the budgeted maximin share problem. However, it is NP-hard to compute  $MMS_i$ . In this section, we try to design a polynomial-time algorithm to find an approximate MMS allocation. As a negative side, we can easily obtain the following results.

**Theorem 4** *For any  $\epsilon > 0$ , there is no polynomial-time algorithm that finds a  $(\frac{2}{3} + \epsilon)$ -approximate MMS for any instance  $I$ , unless  $\mathcal{P} = \mathcal{NP}$ .*

**Proof** Assuming that such an algorithm  $\mathcal{A}$  exists, we show how to use it to solve a famous NP-complete problem, called 3-partition problem [20]. Given a positive integer  $p$  and a set of  $m = 3n$  elements where the size of each element is  $a_j$  such that  $\sum_{j=1}^m a_j = np$ , and  $\frac{p}{4} < a_j < \frac{p}{2}$  for each  $j \in [m]$ . The 3-partition problem is to determine whether there is a partition that divides  $m = 3n$  elements into  $n$  subsets such that each subset contains 3 elements and the size of each subset is exactly  $p$ . Then, we construct an instance  $I$  of the budgeted maximin share problem with  $m = 3n$  goods and  $n$  agents, where  $s(g_j) = a_j$  for each  $j \in [m]$ ,  $b_i = p$  for each agent  $i \in [n]$  and  $v_i(g) = 1$  for each agent  $i$  and good  $g \in G$ . By using  $\mathcal{A}$  to solve instance  $I$ , we can obtain a feasible allocation  $\mathbf{A} = (A_1, A_2, \dots, A_n)$  with  $|A_i| \leq 3$ , as  $\frac{p}{4} < a_i < \frac{p}{2}$ , and  $b_i = p$ .

If  $|A_i| = 3$  for each agent  $i \in N$ , all the goods are allocated, implying that  $s(A_i) = p$  for each  $i \in [n]$ . Thus, the answer to the 3-partition instance is *yes*. Moreover, if the answer to the 3-partition instance is *yes*, we have  $MMS_i = 3$  for each agent  $i \in [n]$ . Therefore,  $v_i(A_i) \geq (\frac{2}{3} + \epsilon) \cdot MMS_i = 2 + 3\epsilon$ , implying that  $|A_i| = 3$ . In other words, the answer to the 3-partition instance is *yes* if and only if  $|A_i| = 3$  for all  $i$ . Therefore, we can solve the 3-partition problem by using algorithm  $\mathcal{A}$ , contradicting the assumption  $\mathcal{P} \neq \mathcal{NP}$ .  $\square$

For any small  $\epsilon \in (0, 1)$ , let  $\mathcal{A}$  be the fully polynomial time approximation scheme for the knapsack problem [15] and  $\alpha = 1 - \epsilon$ . Let  $\epsilon = \frac{1}{5n}$ . Therefore, we have

$$\beta = \frac{\alpha}{1 + 2\alpha - \frac{2\alpha}{n}} = \frac{5n^2 - n}{15n^2 - 15n + 5} \tag{8}$$

in Algorithm 1.

**Algorithm 2** Finding a feasible  $\frac{1}{3}$ -approximate MMS allocation in polynomial time

---

**Input** : An instance  $I = (N, G, v_i, s, b)$ , threshold  $\beta$ .  
**Output**:  $\frac{1}{3}$ -MMS allocation  $\mathbf{A}$ .

- 1 Find a feasible  $n$ -partial-partition  $\mathbf{P}^{(i)} = (P_1^{(i)}, P_2^{(i)}, \dots, P_n^{(i)}) \in \mathcal{P}^{(i)}(G, n)$  for any agent  $i$  by using the method in [30];
- 2 **for every**  $i \in N$  **do**
- 3      $d_i = (n + 2) \cdot \min_{k \in N} v_i(P_k^{(i)})$ ;
- 4     Assume that the MMS values are  $(d_1, d_2, \dots, d_n)$ ;
- 5     Run **Algorithm 2** according to  $(d_1, d_2, \dots, d_n)$  and obtain  $\mathbf{A} = (A_1, A_2, \dots, A_n)$ ;
- 6     **while** there exist  $i \in N$  such that  $v'_i(A_i) < \beta$  **do**
- 7          $d_i \leftarrow (1 - \delta) \cdot d_i$ ;
- 8         Run **Algorithm 2** on  $(d_1, d_2, \dots, d_n)$  and obtain  $\mathbf{A} = (A_1, A_2, \dots, A_n)$ ;

---

For budgeted submodular partitioning, Wang et al. [30] introduced a method for finding a  $n$ -partial-partition  $\mathbf{P}^{(i)} = (P_1^{(i)}, P_2^{(i)}, \dots, P_n^{(i)}) \in \mathcal{P}^{(i)}(G, n)$  of  $G$  for agent  $i$  such that  $\min_{k \in N} v_i(P_k^{(i)}) \geq \frac{\rho}{n\rho+1} \cdot MMS_i$ , where  $\rho$  is the approximation ratio of the greedy algorithm for the classic knapsack problem. However, for the additive utility function, the approximation ratio of greedy algorithms for classical knapsack problems is  $\rho = \frac{1}{2}$ . Therefore, we can find a feasible  $n$ -partial-partition  $\mathbf{P}^{(i)} = (P_1^{(i)}, P_2^{(i)}, \dots, P_n^{(i)}) \in \mathcal{P}^{(i)}(G, n)$  for any agent  $i$  in polynomial time such that

$$\frac{1}{n + 2} \cdot MMS_i \leq \min_{k \in N} v_i(P_k^{(i)}) \leq MMS_i.$$

For each  $i \in N$ , let  $d_i = (n + 2) \cdot \min_{k \in N} v_i(P_k^{(i)})$ . Clearly,

$$MMS_i \leq d_i \leq (n + 2)MMS_i, \text{ for each } i \in N. \tag{9}$$

Since  $d_i \geq MMS_i$ , if agent  $i$  receives a bundle  $A_i \neq \emptyset$  in the allocation  $(A_1, A_2, \dots, A_n)$  produced by Algorithm 1, then  $v_i(A_i) \geq \beta \cdot d_i \geq \beta \cdot MMS_i$ . Otherwise, let

$$\delta = \frac{12n - 5}{15n^2 - 3n} \in (0, 1),$$

and reduce  $d_i$  by multiplying it by a constant factor  $(1 - \delta)$ . In this way, it can be ensured that Algorithm 1 produces a nonempty bundle for agent  $i$  when  $MMS_i \geq d_i > (1 - \delta)MMS_i$ . Formally, our algorithm is shown in Algorithm 2.

**Theorem 5** Algorithm 2 produces a feasible allocation  $\mathbf{A} = (A_1, A_2, \dots, A_n)$  in polynomial time with  $v_i(A_i) \geq \frac{1}{3} \cdot MMS_i$  for any agent  $i \in N$ .

**Proof** By Theorem 2, for any value  $d_i \in ((1 - \delta) \cdot MMS_i, MMS_i]$ , we have

$$v'_i(A_i) = \frac{v_i(A_i)}{d_i} \geq \beta,$$

which implies that the value of  $d_i$  will not decrease. In other words,  $d_i \geq (1 - \delta) \cdot MMS_i$  for each agent  $i$  at the end of Algorithm 2, which implies that

$$v_i(A_i) = v'_i(A_i) \cdot d_i \geq \beta \cdot d_i \geq \beta \cdot (1 - \delta) \cdot MMS_i = \frac{1}{3} \cdot MMS_i,$$

where the first inequality follows from the choice of Algorithm 2, and the last equality follows from the definitions of  $\beta$  and  $\delta$ .

The initial values of  $d_i$  can be computed in time  $O(m^3 \log m)$  by using the method in [30]. Since  $d_i$  is reduced by factor  $(1 - \delta)$ , as in [25], the number of adjustments per agent is at most

$$\log_{1-\delta} \left( \frac{1}{n+2} \right) = \frac{\ln\left(\frac{1}{n+2}\right)}{\ln\left(1 - \frac{12n-5}{15n^2-3n}\right)} \leq \left( 1.25n + \frac{13n}{4(12n-5)} \right) \ln(n+2),$$

where the inequality comes from the Maclaurin formula  $\ln(1 - x) = -\sum_{r=1}^{\infty} \frac{x^r}{r}$ , i.e.,

$$\ln\left(1 - \frac{12n-5}{15n^2-3n}\right) = -\sum_{r=1}^{\infty} \frac{\left(\frac{12n-5}{15n^2-3n}\right)^r}{r} \leq -\frac{12n-5}{15n^2-3n}.$$

Thus, the maximum number of iterations required is  $O(n \log_{1-\delta}(n+2)) = O(n^2 \log n) = O(m^2 \log m)$ , implying that the total running time is polynomial in  $m$ .

Therefore, the theorem holds. □

## 6 Conclusion

In this article, we propose a modified bag-filling algorithm to obtain a  $\frac{1}{3}$ -approximate MMS allocation for the budgeted maximin allocation problem in polynomial time. Designing a polynomial time algorithm to find the best-possible approximate MMS allocation is a very interesting open problem. Another interesting direction is to consider utility functions beyond additivity in the model presented in this article.

**Acknowledgements** The authors thank the anonymous reviewers for the valuable comments. The work is supported in part by the National Natural Science Foundation of China (No. 12071417) and Scientific Research and Innovation Project of Postgraduate Students in the Academic Degree of Yunnan University (No. KC-23236212).

**Data availability** We do not analyse or generate any datasets, because our work proceeds within a theoretical and mathematical approach.

## Declarations

**Conflict of interest** The authors declare that they have no known competing financial interests.

## References

1. Akrami, H., Sharma, E., Taki, S.: Simplification and improvement of MMS approximation. In: Proceedings of the 32th International Joint Conference on Artificial Intelligence, pp. 2485–2493 (2023). <https://doi.org/10.24963/ijcai.2023/276>
2. Akrami, H., Garg, J., Taki, S.: Improving approximation guarantees for maximin share. Preprint at [arXiv:2307.12916](https://arxiv.org/abs/2307.12916) (2023)
3. Akrami, H., Garg, J.: Breaking the 3/4 barrier for approximate maximin share. In: Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 74–91 (2024). <https://doi.org/10.1137/1.9781611977912.4>
4. Amanatidis, G., Aziz, H., Birmpas, G., Filos-Ratsikas, A., Li, B., Moulin, H., Voudouris, A.A., Wu, X.: Fair division of indivisible goods: recent progress and open questions. *Artif. Intell.* (2023). <https://doi.org/10.1016/j.artint.2023.103965>
5. Amanatidis, G., Markakis, E., Nikzad, A., Saberi, A.: Approximation algorithms for computing maximin share allocations. *ACM Trans. Algorithms* **13**(4), 1–28 (2017). <https://doi.org/10.1145/3147173>
6. Asadpour, A., Saberi, A.: An approximation algorithm for max-min fair allocation of indivisible goods. *SIAM J. Comput.* **39**(7), 2970–2989 (2010). <https://doi.org/10.1145/1250790.1250808>
7. Barman, S., Krishnamurthy, S.: Approximation algorithms for maximin fair division. *ACM Trans. Econ. Comput.* **8**(1), 1–28 (2020). <https://doi.org/10.1145/3381525>
8. Barman, S., Khan, A., Shyam, S., Sreenivas, K.: Finding fair allocations under budget constraints. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, no. 5, pp. 5481–5489 (2023). <https://doi.org/10.1609/aaai.v37i5.25681>
9. Barman, S., Khan, A., Shyam, S., Sreenivas, K.: Guaranteeing envy-freeness under generalized assignment constraints. In: Proceedings of the 24th ACM Conference on Economics and Computation, pp. 242–269 (2023). <https://doi.org/10.1145/3580507.3597698>
10. Bezakova, I., Dani, V.: Allocating indivisible goods. *ACM SIGecom Exch.* **5**(3), 11–18 (2005). <https://doi.org/10.1145/1120680.1120683>
11. Biswas, A., Barman, A.: Fair division under cardinality constraints. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence, pp. 91–97 (2018)
12. Budish, E.: The combinatorial assignment problem: approximate competitive equilibrium from equal incomes. *J. Polit. Econ.* **119**(6), 1061–1103 (2011)
13. Caprara, A., Kellerer, H., Pferschy, U.: The multiple subset sum problem. *SIAM J. Optim.* **11**(2), 308–319 (2000). <https://doi.org/10.1137/S1052623498348481>
14. Chakrabarty, D., Chuzhoy, J., Khanna, S.: On allocating goods to maximize fairness. In: Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science, pp. 107–116 (2009). <https://doi.org/10.1109/FOCS.2009.51>
15. Chen, L., Lian, J., Mao, Y., Zhang, G.: A nearly quadratic-time FPTAS for knapsack. In: Proceedings of the 56th Annual ACM Symposium on Theory of Computing, pp. 283–294 (2024). <https://doi.org/10.1145/3618260.3649730>
16. Feige, U., Sapir, A., Tauber, L.: A tight negative example for MMS fair allocations. In: Proceedings of the International Conference on Web and Internet Economics, pp. 355–372 (2021). [https://doi.org/10.1007/978-3-030-94676-0\\_20](https://doi.org/10.1007/978-3-030-94676-0_20)
17. Garg, J., McGlaughlin, P., Taki, S.: Approximating maximin share allocations. In: 2nd Symposium on Simplicity in Algorithms, vol. 69, pp. 20:1–20:11 (2019). <https://doi.org/10.4230/OASiCS.SOSA.2019.20>
18. Garg, J., Taki, S.: An improved approximation algorithm for maximin shares. *Artif. Intell.* (2021). <https://doi.org/10.1145/3391403.3399526>
19. Garbea, M., Gkatzelis, V., Tan, X.: EFX budget-feasible allocations with high Nash welfare. In: 26th European Conference on Artificial Intelligence, pp. 795–802 (2023). <https://doi.org/10.3233/FAIA230346>

20. Garey, M.R., Johnson, D.S.: Computers and intractability: A guide to the theory of NP-completeness. W.H. Freeman, San Francisco (1979)
21. Gan, J., Li, B., Wu, X.: Approximation algorithm for computing budget-feasible EF1 allocations. In: Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems, pp. 170–178 (2023)
22. Ghodsi, M., HajiAghayi, M., Seddighin, M., Seddighin, S., Yami, H.: Fair allocation of indivisible goods: improvements and generalizations. In: Proceedings of the 2018 ACM Conference on Economics and Computation, pp. 539–556 (2018). <https://doi.org/10.1145/3219166.3219238>
23. Guo, H., Li, W., Deng, B.: A survey on fair allocation of chores. Mathematics (2023). <https://doi.org/10.3390/math11163616>
24. Hummel, H., Hetland, M.L.: Maximin shares under cardinality constraints. In: Proceedings of the European Conference on Multi-Agent Systems, pp. 188–206 (2022). [https://doi.org/10.1007/978-3-031-20614-6\\_11](https://doi.org/10.1007/978-3-031-20614-6_11)
25. Hummel, H.: Maximin shares in hereditary set systems. Preprint at [arXiv: 2404.11582](https://arxiv.org/abs/2404.11582) (2024)
26. Ko, S.Y., Chen, H.L., Cheng, S.W., Hon, W.K., Liao, C.S.: Polynomial-time combinatorial algorithm for general max-min fair allocation. Algorithmica **86**(2), 485–504 (2024). <https://doi.org/10.1007/s00453-023-01105-3>
27. Kurokawa, D., Procaccia, A., Wang, J.: Fair enough: guaranteeing approximate maximin shares. J. ACM **65**(2), 1–27 (2018). <https://doi.org/10.1007/s00453-023-01105-3>
28. Saha, B., Srinivasan, A.: A new approximation technique for resource-allocation problems. Random Struct. Algorithms **52**(4), 680–715 (2018). <https://doi.org/10.1002/rsa.20756>
29. Suksompong, W.: Constraints in fair division. ACM SIGecom Exch. **19**(2), 46–61 (2021). <https://doi.org/10.1145/3505156.3505162>
30. Wang, S., Zhou, T., Lavania, C., Bilmes, J.: Constrained robust submodular partitioning. In: Advances in Neural Information Processing Systems, vol. 34, pp. 2721–2732 (2021)
31. Wu, X., Li, B., Gan, J.: Budget-feasible maximum Nash social welfare is almost envy-free. In: International Joint Conferences on Artificial Intelligence, pp. 465–471 (2021)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.