Simplification and Improvement of MMS Approximation

Hannaneh Akrami[1,2] , **Jugal Garg**[3] , **Eklavya Sharma**[3] and **Setareh Taki**[4]
[1]Max Planck Institute for Informatics, Germany
[2]Graduiertenschule Informatik, Universität des Saarlandes, Germany
[3]University of Illinois at Urbana-Champaign, USA
[4]Grubhub, USA
hakrami@mpi-inf.mpg.de, {jugal,eklavya2}@illinois.edu, Staki@grubhub.com

Abstract

We consider the problem of fairly allocating a set of indivisible goods among n agents with additive valuations, using the popular fairness notion of maximin share (MMS). Since MMS allocations do not always exist, a series of works provided existence and algorithms for approximate MMS allocations. The Garg-Taki algorithm gives the current best approximation factor of $(\frac{3}{4} + \frac{1}{12n})$. Most of these results are based on complicated analyses, especially those providing better than $2/3$ factor. Moreover, since no tight example is known of the Garg-Taki algorithm, it is unclear if this is the best factor of this approach. In this paper, we significantly simplify the analysis of this algorithm and also improve the existence guarantee to a factor of $(\frac{3}{4} + \min(\frac{1}{36}, \frac{3}{16n-4}))$. For small n, this provides a noticeable improvement. Furthermore, we present a tight example of this algorithm, showing that this may be the best factor one can hope for with the current techniques.

1 Introduction

Fair division of a set of indivisible goods among n agents with diverse preferences is a fundamental problem in many areas, including game theory, social choice theory, and multi-agent systems. We assume that agents have additive valuations. Maximin share (MMS) is one of the most popular fairness notion in this setting, introduced by Budish [2011], which has attracted a lot of attention in recent years. It is preferred by participating agents over other notions, as shown in real-life experiments by [Gates *et al.*, 2020]. Every agent i has an associated threshold, called her maximin share (MMS_i), defined as the maximum value i can get by partitioning the set of goods into n bundles (one for each agent) and picking a lowest-value bundle. An agent considers an allocation to be fair if she receives goods of total value at least her MMS.

A natural question is whether we can always find an allocation that gives each agent her MMS. Surprisingly, such an allocation need not always exist. Procaccia and Wang [2014] showed examples for any $n \geq 3$ in which MMS allocations do not exist. This motivated them to initiate the study of approximate MMS. Agent i considers an allocation to be α-MMS fair to her for $\alpha \in (0, 1)$ if she receives goods of total value at least $\alpha \cdot \mathrm{MMS}_i$. They showed that a $2/3$-MMS allocation always exists. Ghodsi *et al.* [2018] improved this result by showing the existence of a $3/4$-MMS using a sophisticated algorithm with a very involved analysis. More recently, Garg and Taki [2021] improved this result to $(\frac{3}{4} + \frac{1}{12n})$-MMS using a simple combinatorial algorithm, though their analysis remains quite involved. Furthermore, there is no tight example known for this algorithm, so it is unclear if this is the best factor of the approach.

A complementary problem is to construct examples with the smallest upper bound on α, say α^*, such that α-MMS allocations do not always exist for $\alpha > \alpha^*$. Feige, Sapir, and Tauber [2021] recently obtained the best-known $\alpha^* = 1 - 1/n^4$ for $n \geq 4$. They also gave an improved value of $\alpha^* = 39/40$ for the special case of $n = 3$ agents. However, there is still a substantial gap between the lower and upper bounds.

In this paper, we investigate the Garg-Taki algorithm and obtain the following results.

- A significantly simple analysis of the algorithm.
- An improved bound of $(\frac{3}{4} + \min(\frac{1}{36}, \frac{3}{4(4n-1)}))$-MMS by slightly modifying the algorithm. Since $\min(\frac{1}{36}, \frac{3}{4(4n-1)}) \geq \frac{1}{12n}$ for all $n \geq 3$, this provides noticeable improvement for small n. We note that $\frac{3}{4} + \frac{1}{12n}$ was the best-known bound for $n > 4$.
- A tight example of the Garg-Taki's and our algorithms, which shows the limits of this approach in obtaining a better bound of $\frac{3}{4} + O(1)$. Interestingly, our example only utilizes identical valuations, for which MMS allocations are known to exist.

Our simplified analysis not only helped us to improve the MMS bound but also, together with the tight example, shed more light on why and for which instances the algorithm cannot do better. We believe that these results would help reduce the gap further between the lower and upper bounds.

1.1 Related Work

Computing the maximin share of any agent is NP-hard (even for 2 agents)[1], but a PTAS exists [Woeginger, 1997]. Procac-

[1]by a straightforward reduction from the partition problem.

cia and Wang [2014] showed the existence of a $2/3$-MMS allocation, which can also be computed in polynomial time for a constant n. Later, the algorithm was modified [Amanatidis *et al.*, 2017b; Kurokawa *et al.*, 2018] to compute a $(2/3-\varepsilon)$-MMS allocation in polynomial time (here $\varepsilon > 0$ is a parameter of the algorithm, whose running time increases with $1/\varepsilon$). Barman and Krishnamurthy [2020] gave a simple greedy algorithm with an involved analysis to find a $\frac{2}{3}(1+\frac{1}{3n-1})$-MMS allocation. Garg *et al.* [2018] gave a simple algorithm with a simple analysis to output a $2/3$-MMS allocation.

Ghodsi *et al.* [2018] showed the existence of a $3/4$-MMS allocation using a complicated algorithm and analysis. Garg and Taki [2021] showed how to find a $3/4$-MMS allocation in strongly polynomial time, and showed that $(\frac{3}{4}+\frac{1}{12n})$-MMS allocations exist. Their results use simple algorithms, but their analysis is still quite involved.

Special cases. Amanatidis *et al.* [2017b] showed that when $m \le n+3$, an MMS allocation always exists. Feige *et al.* [2021] improved this to $m \le n+5$. For $n=2$, MMS allocations always exist [Bouveret and Lemaître, 2016]. For $n=3$, the MMS approximation was improved from $3/4$ [Procaccia and Wang, 2014] to $7/8$ [Amanatidis *et al.*, 2017b] to $8/9$ [Gourvès and Monnot, 2019], and then to $11/12$ [Feige and Norkin, 2022]. For $n=4$, Ghodsi *et al.* [2018] showed the existence of $4/5$-MMS.

Experiments. Bouveret and Lemaître [2016] showed that MMS allocations usually exist (for data generated randomly using uniform or Gaussian valuations). Amanatidis *et al.* [2017b] gave a simple and efficient algorithm and showed that when the valuation of each good is drawn independently and randomly from the uniform distribution on $[0,1]$, the algorithm's output is an MMS allocation with high probability when the number of goods or agents is large. Kurokawa *et al.* [2016] gave a similar result for arbitrary distributions of sufficiently large variance.

Chores. MMS can be analogously defined for fair division of chores. MMS allocations do not always exist for chores [Aziz *et al.*, 2017], which motivated the study of approximate MMS [Aziz *et al.*, 2017; Barman and Krishnamurthy, 2020; Huang and Lu, 2021], with the current best approximation ratio being $11/9$. For 3 agents, $19/18$-MMS allocations exist [Feige and Norkin, 2022].

Other settings. MMS has also been studied for non-additive valuations [Barman and Krishnamurthy, 2020; Ghodsi *et al.*, 2018; Li and Vetta, 2021]. Generalizations have been studied where restrictions are imposed on the set of allowed allocations, like matroid constraints [Gourvès and Monnot, 2019], cardinality constraints [Biswas and Barman, 2018], and graph connectivity constraints [Bei *et al.*, 2022; Truszczynski and Lonc, 2020]. Stretegyproof versions of fair division have also been studied [Barman *et al.*, 2019; Amanatidis *et al.*, 2016; Amanatidis *et al.*, 2017a; Aziz *et al.*, 2019]. MMS has also inspired other notions of fairness, like weighted MMS [Farhadi *et al.*, 2019], AnyPrice Share (APS) [Babaioff *et al.*, 2021], Groupwise MMS [Barman *et al.*, 2018; Chaudhury *et al.*, 2021], 1-out-of-d share [Hosseini and Searns, 2021], and self-maximizing shares [Babaioff and Feige, 2022].

1.2 Outline of This Paper

In Section 2, we give formal definitions, notations, and preliminaries. In Section 3, we give a very simple proof that (a minor modification of) the Garg-Taki algorithm [2021] outputs a $3/4$-MMS allocation. In Section 4, we improve the analysis to show that the output is a $(\frac{3}{4}+\min(\frac{1}{36},\frac{3}{4(4n-1)}))$-MMS allocation. In Section 5, we give a tight example for our algorithm.

2 Preliminaries

For any non-negative integer n, let $[n] := \{1,2,\ldots,n\}$.

A *fair division instance* $\mathcal{I}$ is specified by a triple (N,M,v), where N is the set of agents, M is the set of goods, and $v_{i,g}$ is the value of good $g \in M$ for agent $i \in N$. For a set S of goods, define $v_i(S) := \sum_{g\in S} v_{i,g}$. Then v_i is called agent i's *valuation function*. Intuitively, $v_i(S)$ is a measure of how valuable S is to i. For ease of notation, we write $v_i(g)$ instead of $v_i(\{g\})$. We can assume without loss of generality that $N=[n]$ and $M=[m]$, where $n=|N|$ and $m=|M|$ (though when dealing with multiple related fair division instances, not making this assumption can sometimes simplify notation).

For a set S of goods, let $\Pi_n(S)$ denote the set of partitions of S into n bundles. For any valuation function u, define

$$\mathrm{MMS}_u^n(S) := \max_{X\in\Pi_n(S)} \min_{j=1}^{n} u(X_j).$$

When the fair division instance (N,M,v) is clear from context, we write MMS_i instead of $\mathrm{MMS}_{v_i}^{|N|}(M)$ for conciseness.

2.1 Ordered Instance

Definition 1. *A fair division instance (N,M,v) is called* ordered *if there is an ordering $[g_1,g_2,\ldots,g_{|M|}]$ of goods M such that for each agent i, $v_{i,g_1} \ge v_{i,g_2} \ge \ldots \ge v_{i,g_{|M|}}$.*

We will now see how to reduce the problem of finding an α-MMS allocation to the special case of ordered instances.

Definition 2. *For the fair division instance $\mathcal{I} := (N,M,v)$, $\texttt{toOrd}(\mathcal{I})$ is defined as the instance $(N,[|M|],\widehat{v})$, where for each $i\in N$ and $j\in[|M|]$, $\widehat{v}_{i,j}$ is the j^{th} largest number in the multiset $\{v_{i,g} \mid g\in M\}$.*

In Theorem 3.2 of [Barman and Krishnamurthy, 2020], it was shown that the transformation toOrd is *α-MMS-preserving*, i.e., for a fair division instance $\mathcal{I}$, given an α-MMS allocation of $\texttt{toOrd}(\mathcal{I})$, we can compute an α-MMS allocation of $\mathcal{I}$ in polynomial time. (The proof is based on ideas by Bouveret and Lemaître [2016]).

2.2 Valid Reductions

We use a technique called *valid reduction*, that helps us reduce a fair division instance to a smaller instance. This technique has been implicitly used in [Bouveret and Lemaître, 2016; Kurokawa *et al.*, 2016; Kurokawa *et al.*, 2018; Amanatidis *et al.*, 2017b; Ghodsi *et al.*, 2018; Garg *et al.*, 2018] and explicitly used in [Garg and Taki, 2021].

Definition 3 (Valid reduction)**.** *In a fair division instance (N, M, v), suppose we give the goods S to agent i. Then we are left with a new instance $(N \setminus \{i\}, M \setminus S, v)$. Such a transformation is called a* valid α-reduction *if both of these conditions hold:*

1. $v_i(S) \geq \alpha \text{MMS}_{v_i}^{|N|}(M)$.
2. $\text{MMS}_{v_j}^{|N|-1}(M \setminus S) \geq \text{MMS}_{v_j}^{|N|}(M)$ *for all* $j \in N \setminus \{i\}$.

Note that valid reductions are α-MMS-preserving, i.e., if A is an α-MMS allocation of an instance obtained by performing a valid reduction, then we can get an α-MMS allocation of the original instance by giving goods S to agent i and allocating the remaining goods as per A. A valid reduction, therefore, helps us reduce the problem of computing an α-MMS allocation to a smaller instance.

We now describe four standard transformations, called *reduction rules*, and show that they are valid reductions.

Definition 4 (Reduction rules)**.** *Consider an ordered fair division instance (N, M, v), where $M := \{g_1, \ldots, g_{|M|}\}$ and $v_{i,g_1} \geq \ldots \geq v_{i,g_{|M|}}$ for every agent i. Define*

1. $S_1 := \{g_1\}$.
2. $S_2 := \{g_{|N|}, g_{|N|+1}\}$ *if* $|M| \geq |N| + 1$, *else* $S_2 := \emptyset$.
3. $S_3 := \{g_{2|N|-1}, g_{2|N|}, g_{2|N|+1}\}$ *if* $|M| \geq 2|N| + 1$, *else* $S_3 := \emptyset$.
4. $S_4 := \{g_1, g_{2|N|+1}\}$ *if* $|M| \geq 2|N| + 1$, *else* $S_4 := \emptyset$.

Reduction rule $R_k(\alpha)$: If $v_i(S_k) \geq \alpha\text{MMS}_i$ for some agent i, then give S_k to i.

A fair division instance is called $R_k(\alpha)$-irreducible *if $R_k(\alpha)$ cannot be applied, i.e., $v_i(S_k) < \alpha\text{MMS}_i$ for every agent i (otherwise it is called* $R_k(\alpha)$-reducible*). An instance is called* totally-α-irreducible *if it is $R_k(\alpha)$-irreducible for all $k \in [4]$.*

Lemma 1 (Lemma 3.1 in [Garg and Taki, 2021])**.** *For an ordered instance and for $\alpha \leq 1$, $R_1(\alpha)$, $R_2(\alpha)$, and $R_3(\alpha)$ are valid α-reductions. For an ordered instance and for $\alpha \leq 3/4$, if the instance is $R_1(\alpha)$-irreducible and $R_3(\alpha)$-irreducible, then $R_4(\alpha)$ is a valid α-reduction.*

Lemma 2. *Let $\mathcal{I} := ([n], [m], v)$ be an ordered instance where $v_{i,1} \geq \ldots \geq v_{i,m}$ for each agent i. For any $k \in [3]$, if $\mathcal{I}$ is $R_k(\alpha)$-irreducible, then for each agent i and every good $j > (k-1)n$, we have $v_{i,j} < \alpha\text{MMS}_i/k$.*

Proof. Since $\mathcal{I}$ is $R_k(\alpha)$-irreducible, we get $v_i(S_k) < \alpha\text{MMS}_i$ for each agent i. Let $t := (k-1)n + 1$. Then

$$\alpha\text{MMS}_i > v_i(S_k) = \sum_{g \in S_k} v_{i,g} \geq |S_k| \min_{g \in S_k} v_{i,g} = k v_{i,t}.$$

Hence, $\forall j \geq t$, we have $v_{i,j} \leq v_{i,t} < \alpha\text{MMS}_i/k$. ☐

Lemma 3. *If an ordered instance (N, M, v) is $R_1(\alpha)$-irreducible for any $\alpha \leq 1$, then $|M| \geq 2|N|$.*

Proof. Assume $|M| < 2|N|$. Pick any agent $i \in N$. Let P be an MMS partition of agent i. Then some bundle P_j contains a single good $\{g\}$. Then $v_{i,g} = v_i(P_j) \geq \text{MMS}_i$. Hence, the instance is not $R_1(\alpha)$-irreducible for any $\alpha \leq 1$. This is a contradiction. Hence, $|M| \geq 2|N|$. ☐

Algorithm 1 `normalize`$((N, M, v))$

1: **for** $i \in N$ **do**
2: Compute agent i's MMS partition $P^{(i)}$.
3: $\forall j \in N, \forall g \in P_j^{(i)}$, let $\widehat{v}_{i,g} := v_{i,g}/v_i(P_j^{(i)})$.
4: **end for**
5: **return** $(N, M, \widehat{v})$.

We would like to convert fair division instances into totally-α-irreducible instances. This can be done using a very simple algorithm, which we call `reduce`$_\alpha$. This algorithm works for $\alpha \leq 3/4$. It takes an ordered fair division instance as input and repeatedly applies the reduction rules $R_1(\alpha)$, $R_2(\alpha)$, $R_3(\alpha)$, and $R_4(\alpha)$ until the instance becomes totally-α-irreducible. The reduction rules can be applied in arbitrary order, except that $R_4(\alpha)$ is only applied when $R_1(\alpha)$ and $R_3(\alpha)$ are inapplicable.

Note that the application of reduction rules changes the number of agents and goods, which affects subsequent reduction rules. More precisely, the sets S_1, S_2, S_3, S_4 (as defined in Definition 4) can change after applying a reduction rule. So, for example, it is possible that an instance is $R_2(\alpha)$-irreducible, but after applying $R_3(\alpha)$, the resulting instance is $R_2(\alpha)$-reducible.

2.3 Normalized Instance

Definition 5 (Normalized instance)**.** *A fair division instance (N, M, v) is called* normalized *if for every agent i, there is a partition $P^{(i)} := (P_1^{(i)}, \ldots, P_{|N|}^{(i)})$ of M such that $v_i(P_j^{(i)}) = 1\ \forall j \in N$.*

Note that for a normalized instance, every agent's MMS value is 1. Furthermore, for each agent i and for every MMS partition Q of agent i, we have $v_i(Q_j) = 1\ \forall j \in N$, since each partition has total value at least 1 and $\sum_{j \in N} v_i(Q_j) = v_i(M) = \sum_{j \in N} v_i(P_j^{(i)}) = |N|$.

The algorithm `normalize` (c.f. Algorithm 1) converts a fair division instance to a normalized instance.

Lemma 4. *Let $(N, M, \widehat{v}) =$ `normalize`$((N, M, v))$. Then for any allocation A, $v_i(A_i) \geq \widehat{v}_i(A_i)\text{MMS}_{v_i}^{|N|}(M)$ for all $i \in N$.*

Proof. Let $\beta_i := \text{MMS}_{v_i}^n(M)$. For any good $g \in P_j^{(i)}$, $\widehat{v}_{i,g} = v_{i,g}/v_i(P_j^{(i)}) \leq v_{i,g}/\beta_i$. Hence, $v_{i,g} \geq \widehat{v}_{i,g}\beta_i$. Hence, $v_i(A_i) \geq \widehat{v}_i(A_i)\beta_i$. ☐

Lemma 4 implies that `normalize` is α-MMS-preserving, since if A is an α-MMS allocation for the normalized instance $(N, M, \widehat{v})$, then A is also an α-MMS allocation for the original instance (N, M, v).

3 Simple Proof for Existence of 3/4-MMS Allocations

We give an algorithm, called `approxMMS` (c.f. Algorithm 2), that takes as inputs a fair division instance and an approximation factor α, and outputs an α-MMS allocation. It works in three major steps:

Algorithm 2 $\texttt{approxMMS}(\mathcal{I}, \alpha)$
Input: Fair division instance $\mathcal{I} = (N, M, v)$ and approximation factor α.
Output: Allocation $A = (A_1, \ldots, A_n)$.

$\widehat{\mathcal{I}} = \texttt{toOrd}(\texttt{normalize}(\texttt{reduce}_\alpha(\texttt{toOrd}(\mathcal{I}))))$
$\widehat{A} = \texttt{bagFill}(\widehat{\mathcal{I}}, \alpha)$.
Use $\widehat{A}$ to compute an allocation A for $\mathcal{I}$ with the same MMS approximation as $\widehat{A}$. (This can be done since Sections 2.1, 2.2 and 2.3 show that toOrd, $\texttt{reduce}_\alpha$, and normalize are α-MMS-preserving.)
return A

1. Reduce the problem of finding an α-MMS allocation to the special case where the instance is Ordered, Normalized, and totally-α-Irreducible (ONI).
2. Compute an α-MMS allocation for this special case using the bagFill algorithm (c.f. Algorithm 3).
3. Convert this allocation for the special case to an allocation for the original fair division instance.

We describe steps 1 and 3 in Section 3.1 and step 2 in Section 3.2. In this section, we only consider the case where $\alpha = 3/4$. In Section 4, we slightly modify approxMMS so that it works for $\alpha = \frac{3}{4} + \min(\frac{1}{36}, \frac{3}{4(4n-1)})$. Our algorithm approxMMS is almost the same as the algorithm of Garg and Taki [2021]. The only difference is that, unlike them, we ensure that the output of step 1 is normalized.

3.1 Obtaining an Ordered Normalized Irreducible (ONI) Instance

Lemma 5. *Let $\mathcal{I}$ be a fair division instance. Let $\widehat{\mathcal{I}} := \texttt{toOrd}(\texttt{normalize}(\texttt{reduce}_{3/4}(\texttt{toOrd}(\mathcal{I}))))$. Then $\widehat{\mathcal{I}}$ is ordered, normalized, and totally-$3/4$-irreducible. Furthermore, the transformation of $\mathcal{I}$ to $\widehat{\mathcal{I}}$ is $3/4$-MMS-preserving, i.e., a $3/4$-MMS allocation of $\widehat{\mathcal{I}}$ can be used to obtain a $3/4$-MMS allocation of $\mathcal{I}$.*

Proof. Let $\mathcal{I}^{(1)} := \texttt{toOrd}(\mathcal{I})$. Then $\mathcal{I}^{(2)} := \texttt{reduce}_{3/4}(\mathcal{I}^{(1)})$ is totally-$3/4$-irreducible and ordered, since the application of reduction rules preserves orderedness.

Let $\mathcal{I}^{(3)} := \texttt{normalize}(\mathcal{I}^{(2)})$. By Lemma 4, normalize does not increase the ratio of a good's value to the MMS value. Hence, $\widehat{\mathcal{I}}$ is totally-$3/4$-irreducible. $\widehat{\mathcal{I}}$ is also normalized, since for each agent, toOrd only changes the identities of the goods, but the (multi-)set of values of the goods remains the same. Hence, $\widehat{\mathcal{I}}$ is ordered, normalized, and totally-$3/4$-irreducible.

Since toOrd, $\texttt{reduce}_{3/4}$, and normalize are $3/4$-MMS-preserving operations, their composition is also $3/4$-MMS-preserving. □

The order of operations is important here, as well as the need to call toOrd twice, since reduce requires the input to be ordered, reduce may not preserve normalizedness, and normalize may not preserve orderedness.

Algorithm 3 $\texttt{bagFill}(\mathcal{I}, \alpha)$
Input: Ordered instance $\mathcal{I} = ([n], [m], v)$ with $m \geq 2n$ and approximation factor α.
Output: (Partial) allocation $A = (A_1, \ldots, A_n)$.

1: **for** $k \in [n]$ **do**
2: $\quad B_k = \{k, 2n + 1 - k\}$.
3: **end for**
4: $U_G = [m] \setminus [2n]$ *// unassigned goods*
5: $U_A = [n]$ *// unsatisfied agents*
6: $U_B = [n]$ *// unassigned bags*
7: **while** $U_A \neq \emptyset$ **do** *// loop invariant: $|U_A| = |U_B|$*
8: $\quad$ **if** $\exists i \in U_A$, $\exists k \in U_B$, such that $v_i(B_k) \geq \alpha$ **then**
9: $\quad\quad$ *// assign the k^{th} bag to agent i:*
10: $\quad\quad A_i = B_k$
11: $\quad\quad U_A = U_A \setminus \{i\}$
12: $\quad\quad U_B = U_B \setminus \{k\}$
13: $\quad$ **else if** $U_G \neq \emptyset$ **then**
14: $\quad\quad g$ = arbitrary good in U_G
15: $\quad\quad k$ = arbitrary bag in U_B
16: $\quad\quad$ *// assign g to the k^{th} bag:*
17: $\quad\quad B_k = B_k \cup \{g\}$.
18: $\quad\quad U_G = U_G \setminus \{g\}$
19: $\quad$ **else**
20: $\quad\quad$ **error**: we ran out of goods. **return** null.
21: $\quad$ **end if**
22: **end while**
23: **return** $(A_1, \ldots, A_n)$

Garg and Taki [2021] transform the instance as $\texttt{reduce}_{3/4}(\texttt{toOrd}(\mathcal{I}))$, since they do not need the input to be normalized.

3.2 3/4-MMS Allocation of ONI Instance

Let $([n], [m], v)$ be a fair division instance that is ordered, normalized, and totally-$3/4$-irreducible (ONI). Without loss of generality, assume that $v_{i,1} \geq v_{i,2} \geq \ldots \geq v_{i,m}$ for each agent i.

Our algorithm, called $\texttt{bagFill}(\mathcal{I}, \alpha)$, creates n bags, where the j^{th} bag contains goods $\{j, 2n + 1 - j\}$. (To create bags in this way, there must be at least $2n$ goods. This is ensured by Lemma 3.) It then repeatedly adds a good to an arbitrary bag, and as soon as some agent i values a bag more than α, that bag is allocated to i. The algorithm terminates when all agents have been allocated a bag. See Algorithm 3 for a more precise description. (In this section, we set $\alpha = 3/4$. In Section 4, we set $\alpha = \frac{3}{4} + \min(\frac{1}{36}, \frac{3}{4(4n-1)})$.) bagFill computes a partial allocation, i.e., some goods may remain unallocated. But that can be easily fixed by arbitrarily allocating those goods among the agents.

$\texttt{bagFill}(\mathcal{I}, \alpha)$ allocates a bag B_k to agent i only if $v_i(B_k) \geq \alpha$. Hence, to prove that $\texttt{bagFill}(\mathcal{I}, 3/4)$ returns a $3/4$-MMS allocation, it suffices to show that bagFill terminates successfully, i.e., line 20 is never executed.

For $k \in [n]$, let $B_k := \{k, 2n + 1 - k\}$ be the initial contents of the k^{th} bag and B'_k be the k^{th} bag's contents after bagFill terminates. We consider two groups of agents. Let N^1 be the set of agents who value all the initial bags at most

1. Formally, $N^1 := \{i \in [n] \mid \forall k \in [n], v_i(B_k) \le 1\}$. Let $N^2 := [n] \setminus N^1 = \{i \in [n] \mid \exists k \in [n] : v_i(B_k) > 1\}$ be the rest of the agents.

Let U_A be the set of agents that did not receive a bag when `bagFill` terminated. Note that U_A is non-empty iff we execute line 20. We first show that all agents in N^1 receive a bag, i.e., $U_A \cap N^1 = \emptyset$. Then we show that $U_A \cap N^2 = \emptyset$. Together, these facts establish that `bagFill` terminates successfully, and hence its output is 3/4-MMS.

Lemma 6. *Let $([n], [m], v)$ be an ordered and normalized fair division instance. For all $k \in [n]$ and agent $i \in [n]$, if $v_{i,k} + v_{i,2n-k+1} > 1$, then $v_{i,2n-k+1} \le 1/3$ and $v_{i,k} > 2/3$.*

Proof. It suffices to prove $v_{i,2n-k+1} \le 1/3$ and then $v_{i,k} > 2/3$ follows. Let $P = (P_1, \dots, P_n)$ be an MMS partition of agent i. For $j \in [k]$ and $j' \in [2n+1-k]$, $v_{i,j} + v_{i,j'} \ge v_{i,k} + v_{i,2n+1-k} > 1$, since the instance is ordered. Furthermore, j and j' cannot be in the same bundle in P, since the instance is normalized. In particular, no two goods from $[k]$ are in the same bundle in P. Hence, assume without loss of generality that $j \in P_j$ for all $j \in [k]$.

For all $j \in [k]$ and $j' \in [2n-k+1]$, $j' \notin P_j$. Thus, $\{k+1, \dots, 2n-k+1\} \subseteq P_{k+1} \cup \dots \cup P_n$. By pigeonhole principle, there exists a bundle $B \in \{P_{k+1}, \dots, P_n\}$ that contains at least 3 goods g_1, g_2, g_3 in $\{k+1, \dots, 2n-k+1\}$. Hence,

$$\begin{aligned} v_{i,2n-k+1} &\le \min_{g \in \{g_1,g_2,g_3\}} v_{i,g} \le \frac{1}{3} \sum_{g \in \{g_1,g_2,g_3\}} v_{i,g} \\ &\le \frac{v_i(B)}{3} = \frac{1}{3}. \end{aligned}$$

□

Lemma 7. *Let i be any agent. For all $k \in [n]$, if $v_i(B_k) \le 1$, then $v_i(B'_k) \le 1$.*

Proof. If $B'_k = B_k$, then the claim obviously holds. Now assume $B_k \subsetneq B'_k$. Let g be the last good that was added to B'_k. We have $v_i(B'_k \setminus g) < 3/4$, otherwise g would not be added to B'_k. Also note that $g > 2n$ and hence $v_{i,g} < 1/4$ by Lemma 2. Thus, we have

$$v_i(B'_k) = v_i(B'_k \setminus g) + v_{i,g} < \frac{3}{4} + \frac{1}{4} = 1.$$

□

Lemma 8. *$U_A \cap N^1 = \emptyset$, i.e., every agent in N^1 gets a bag.*

Proof. For the sake of contradiction, assume $U_A \cap N^1 \neq \emptyset$. Hence, $\exists i \in U_A \cap N^1$. Also, for some $j \in [n]$, the j^{th} bag is unallocated. Hence, $v_i(B'_j) < 3/4$ and

$$\begin{aligned} n = v_i(M) &= v_i(B'_j) + \sum_{k \in [n]\setminus\{j\}} v_i(B'_k) \\ &\qquad\qquad (\text{since } M = \textstyle\bigcup_{k\in[n]} B'_k) \\ &< (n-1) + \frac{3}{4} = n - \frac{1}{4}, \qquad (\text{by Lemma 7}) \end{aligned}$$

which is a contradiction. Hence, $U_A \cap N^1 = \emptyset$. □

Now we prove that `bagFill` allocates a bag to all agents in N^2, i.e., $U_A \cap N^2 = \emptyset$.

Lemma 9. $i \in N^2 \implies v_{i,2n+1} < 1/12$.

Proof. Since $i \in N^2$, there exists a bag B_k such that $v_i(B_k) > 1$. By Lemma 6, $v_{i,k} > 2/3$. Thus, $v_{i,1} > 2/3$. Moreover,

$$\begin{aligned} v_{i,2n+1} &< \frac{3}{4} - v_{i,1} \qquad (\text{since } R_4(3/4) \text{ is not applicable}) \\ &< \frac{3}{4} - \frac{2}{3} = \frac{1}{12}. \qquad (\text{since } v_{i,1} > 2/3) \end{aligned}$$

□

From now on assume for the sake of contradiction that $U_A \neq \emptyset$. Let a be a fixed agent in U_A. By Lemma 8, $a \in N^2$. Let $A^+ := \{k \in [n] \mid v_a(B_k) > 1\}$, $A^- := \{k \in [n] \mid v_a(B_k) < 3/4\}$, and $A^0 := \{k \in [n] \mid 3/4 \le v_a(B_k) \le 1\}$. We will try to get upper bounds on $v_a(B'_k)$ for each of the cases $k \in A^+$, $k \in A^-$, and $k \in A^0$.

Note that $n = |A^+| + |A^-| + |A^0|$. Also, $n \in A^-$ since the instance is $R_2(3/4)$-irreducible, and $|A^+| \ge 1$ since $a \in N^2$.

Lemma 10. $\forall k \in A^-$, $v_a(B'_k) < 5/6$.

Proof. If $B'_k = B_k$, then $v_a(B'_k) < 3/4 < 5/6$. Otherwise, let g be the last good that was added to B'_k. Then $v_a(B'_k \setminus \{g\}) < 3/4$, otherwise `bagFill` would assign $B'_k \setminus \{g\}$ to agent i instead of adding g to it. Hence,

$$\begin{aligned} v_a(B'_k) &= v_a(B'_k \setminus \{g\}) + v_{a,g} \\ &< \frac{3}{4} + v_{a,2n+1} \\ &\qquad (\text{since } v_a(B'_k \setminus g) < 3/4 \text{ and } v_{a,g} \le v_{a,2n+1}) \\ &< \frac{3}{4} + \frac{1}{12} = \frac{5}{6}. \\ &\qquad (v_{a,2n+1} < 1/12 \text{ by Lemma 9}) \end{aligned}$$

□

Let ℓ be the smallest such that for all $k \in [\ell+1, n]$, $v_{a,k} + v_{a,2n-k+1+\ell} \le 1$. See Fig. 1 for a better understanding of ℓ. Note that $\ell \ge 1$, since $a \in N^2$.

Lemma 11. $\sum_{k\in A^+} v_a(B'_k) < |A^+| + \min(\ell, |A^+|)/12$.

Proof. Let $S \in A^+$ be the set of $\min(\ell, |A^+|)$ smallest indices in A^+ and $L \in A^+$ be the set of $\min(\ell, |A^+|)$ largest indices in A^+. Since $|A^+| \ge 1$ and $\ell \ge 1$, we get $|S| = |L| \ge 1$. Note that

$$\begin{aligned} \sum_{k\in A^+} v_a(B'_k) &= \left(\sum_{k\in S} v_{a,k} + \sum_{k\in L} v_{a,2n-k+1} \right) \\ &\quad + \left(\sum_{k\in A^+\setminus S} v_{a,k} + \sum_{k\in A^+\setminus L} v_{a,2n-k+1} \right). \end{aligned}$$

By Lemma 6, we get $v_{a,2n-k+1} \le \frac{1}{3}$. Since $v_{a,k} < 3/4$ and $|S| \ge 1$, we get

$$\sum_{k\in S} v_{a,k} + \sum_{k\in L} v_{a,2n-k+1} < |S| \left(\frac{3}{4} + \frac{1}{3} \right) = \frac{13}{12} |S|. \qquad (1)$$

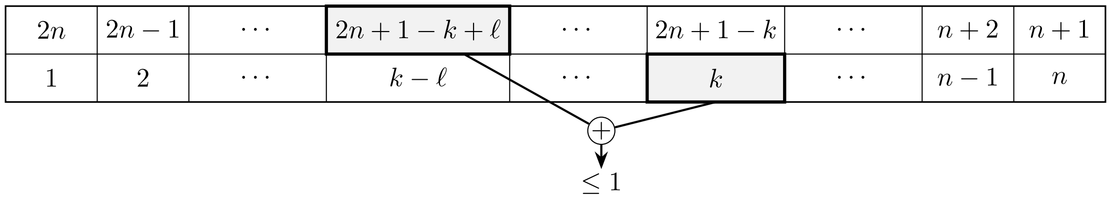

Figure 1: The items $[2n]$ are arranged in a table, where the k^{th} column is $B_k := \{k, 2n+1-k\}$. For $i \in N^1$, we have $v_i(B_k) = v_{i,k} + v_{i,2n+1-k} \leq 1$ for all k. However, $a \notin N^1$. Hence, we look for the smallest *shift* ℓ such that $v_{a,k} + v_{a,2n+1-k+\ell} \leq 1$ for all k.

If $\ell \geq |A^+|$, then $|S| = |L| = |A^+|$, and we are done. Now assume $\ell < |A^+|$. Then $|S| = |L| = \ell$.

Let $A^+ := \{g_1, \ldots, g_{|A^+|}\}$ and $g_1 < \ldots < g_{|A^+|}$. Then $A^+ \setminus S = \{g_{\ell+1}, \ldots, g_{|A^+|}\}$ and $A^+ \setminus L = \{g_1, \ldots, g_{|A^+|-\ell}\}$. The idea is to pair the goods $g_{k+\ell}$ and $2n - g_k + 1$ (for $k \in [|A^+| - \ell]$) and prove that their value is at most 1 for agent a.

Since $g_{k+\ell} \geq g_k + \ell$, we get $v_{a,g_{k+\ell}} + v_{a,2n-g_k+1} \leq 1$ by definition of ℓ. Hence,

$$\begin{aligned}&\sum_{k \in A^+ \setminus S} v_{a,k} + \sum_{k \in A^+ \setminus L} v_{a,2n-k+1} \\ &= \sum_{k \in [|A^+|-\ell]} (v_{a,g_{k+\ell}} + v_{a,2n-g_k+1}) \leq |A^+| - \ell.\end{aligned} \tag{2}$$

Equations (1) and (2) imply Lemma 11. □

Lemma 12. $v_a([m] \setminus [2n]) > \ell/4$.

Proof. By definition of ℓ, there exists a good $k \in \{\ell, \ldots, n\}$ such that $v_{a,k} + v_{a,2n-k+\ell} > 1$. Hence, for all $j \in [k]$ and $t \leq [2n-k+\ell]$, we have $v_{a,j} + v_{a,t} \geq v_{a,k} + v_{a,2n-k+\ell} > 1$.

Let $P := (P_1, \ldots, P_n)$ be an MMS partition of agent a. Then, for $j \in [k]$ and $t \in [2n-k+\ell]$, j and t cannot be in the same bundle in P, since the instance is normalized. In particular, no two goods from $[k]$ are in the same bundle in P. Hence, assume without loss of generality that $j \in P_j$ for all $j \in [k]$. Thus, $[2n-k+\ell] \setminus [k] \subseteq P_{k+1} \cup \ldots \cup P_n$.

Bundles in $\{P_1, \ldots, P_k\}$ can only have goods from $[k]$, $[2n] \setminus [2n-k+\ell]$, and $[m] \setminus [2n]$. There are $k - \ell$ goods in $[2n] \setminus [2n-k+\ell]$. Hence, at least ℓ bundles in $\{P_1, \ldots, P_k\}$ have just 1 good from $[2n]$. Let L be the indices of these bundles, i.e., $L := \{t \in [k] \mid |P_t \cap [2n]| = 1\}$. Then

$$\begin{aligned}v_a([m] \setminus [2n]) &\geq \sum_{j \in L} v_a(P_j \setminus \{j\}) \\ &= \sum_{j \in L} (v_a(P_j) - v_{a,j}) \\ &> \sum_{j \in L} \left(1 - \frac{3}{4}\right) = \frac{|L|}{4} \geq \frac{\ell}{4}. \\ &\qquad (v_{a,j} < 3/4 \text{ by Lemma 2})\end{aligned}$$

□

Lemma 13. *For all $i \in N^2$ and $k \in [n]$, $v_i(B_k) > 1/2$.*

Proof. Fix an $i \in N^2$. Let t be smallest such that $v_i(B_t) > 1$. By Lemma 6, $v_{i,t} > 2/3$. Hence, for all $k \leq t$,

$$v_i(B_k) \geq v_{i,k} \geq v_{i,t} > \frac{2}{3} > \frac{1}{2}.$$

Since $v_i(B_t) = v_{i,t} + v_{i,2n-t+1} > 1$ and $v_{i,t} < 3/4$ (by Lemma 2), we get $v_{i,2n-t+1} > 1/4$. For all $k > t$, we have $k < 2n - k + 1 < 2n - t + 1$. Hence,

$$v_i(B_k) = v_{i,k} + v_{i,2n-k+1} \geq 2 \cdot v_{i,2n-t+1} > \frac{1}{2}.$$

□

Lemma 14. $U_A \cap N^2 = \emptyset$*, i.e., every agent in* N^2 *gets a bag.*

Proof. Assume for the sake of contradiction that $U_A \cap N^2 \neq \emptyset$. Then, as discussed before, we fix an agent $a \in U_A \cap N^2$ and define A^+, A^-, A^0, and ℓ.

$$\begin{aligned}n = v_a([m]) &= \sum_{k \in [n]} v_a(B'_k) \\ &= \sum_{k \in A^-} v_a(B'_k) + \sum_{k \in A^+} v_a(B'_k) + \sum_{k \in A^0} v_a(B'_k) \\ &< \frac{5}{6}|A^-| + \left(|A^+| + \frac{\ell}{12}\right) + |A^0| \\ &\qquad \text{(by Lemmas 10 and 11)} \\ &= n + \frac{\ell}{12} - \frac{|A^-|}{6}\end{aligned}$$

Hence, $|A^-| < \ell/2$.

Now we show that there are enough goods in $[m] \setminus [2n]$ to fill the bags in A^-.

$$\begin{aligned}\frac{\ell}{4} &\leq v_a([m] \setminus [2n]) && \text{(by Lemma 12)} \\ &= \sum_{k \in A^-} (v_a(B'_k) - v_a(B_k)) \\ &&& \text{(since } B'_k = B_k \subseteq [2n] \text{ for } k \in A^+ \cup A^0\text{)} \\ &< |A^-| \left(\frac{5}{6} - \frac{1}{2}\right) && \text{(by Lemmas 10 and 13)} \\ &= |A^-| \cdot \frac{1}{3} < \frac{\ell}{6}, && \text{(since } |A^-| < \ell/2\text{)}\end{aligned}$$

which is a contradiction. □

By Lemmas 8 and 14, we get that $U_A = \emptyset$, i.e., every agent gets a bag, and hence, `bagFill`'s output is $3/4$-MMS.

4 Better than $3/4$-MMS

In this section, we give an overview of how to refine the techniques of Section 3 to get an algorithm that outputs a $(\frac{3}{4} + \min(\frac{1}{36}, \frac{3}{4(4n-1)}))$-MMS allocation. The details can be found in Appendix A of the full version of our paper [Akrami *et al.*, 2023].

Theorem 1. *For any fair division instance with additive valuations, a* $(\frac{3}{4} + \min(\frac{1}{36}, \frac{3}{4(4n-1)}))$*-MMS allocation exists.*

Algorithm `approxMMS` from Section 3 does not work with $\alpha > 3/4$, since $R_4(\alpha)$ may not be a valid reduction. To fix this, we modify $R_4(\alpha)$ using the *dummy goods* technique from [Garg and Taki, 2021].

Consider the fair division instance $([n], [m], v)$. When performing $R_4(\alpha)$, in addition to giving the goods $S_4 := \{1, 2n+1\}$ to some agent i for whom $v_i(S_4) \geq \alpha\mathrm{MMS}_i$, we create a *dummy good* g where $v_j(g) := \max(0, v_j(S_4) - \mathrm{MMS}_j)$ for each agent $j \neq i$. With this change, $R_4(\alpha)$ becomes a valid reduction even for $\alpha > 3/4$. See Appendix A.1 for a proof. Note that dummy goods are fictional, i.e., they exist solely to guide the valid reductions. No agent is allocated a dummy good.

Formally, a fair division instance with dummy goods is represented as a tuple $\mathcal{I} := (N, M, v, D)$, where D is the set of dummy goods and M is the set of non-dummy goods. We can extend the concepts of Section 2.1 (ordered instance), Section 2.2 (valid reductions), and Section 2.3 (normalized instance) to instances with dummy goods. See Appendix A.2 for details. In particular, instance (N, M, v, D) is ordered iff (N, M, v) is ordered, and (N, M, v, D) is normalized iff $(N, M \cup D, v)$ is normalized.

With these modifications, we can extend `approxMMS` to the case where $\alpha > 3/4$. `approxMMS` first transforms the instance into an ordered, normalized, and totally-α-irreducible instance. Then it discards all the dummy goods and allocates the remaining goods using the algorithm `bagFill`. In Appendix A.3, we show that when $\alpha \leq \frac{3}{4} + \min(\frac{1}{36}, \frac{3}{4(4n-1)})$, `bagFill` allocates a bag of value at least α to every agent. Our proof is almost the same as that in Section 3. The main difference is that the analogue of Lemma 14 (Lemma 25 in Appendix A) involves more elaborate algebraic manipulations so that we can get tighter bounds.

5 Tight Example

We give an almost tight example for our algorithm and Garg and Taki's [2021] algorithm. We show that these algorithms' output on this example is not better than $(\frac{3}{4} + \frac{3}{8n-4})$-MMS.

Example 1. *Consider a fair division instance with n agents and $m = 3n - 1$ goods. All agents have the same valuation function u, where*

$$u(j) := \begin{cases} \dfrac{2n - 1 - \lfloor (j-1)/2 \rfloor}{4n-2} & \text{if } j \leq 2n \\ \dfrac{n}{4n-2} & \text{if } j > 2n \end{cases}.$$

Lemma 15. *Example 1 is normalized.*

Proof. Let $M_1 := \{1, 2\}$ and for $i \in [n-1]$, let $M_{i+1} := \{i+2, 2n+1-i, 2n+i\}$. Then for any $i \neq j$, $M_i \cap M_j = \emptyset$. Also, $u(M_1) = u(1) + u(2) = 1$ and for each $i \in [n-1]$,

$$\begin{aligned} &(4n-2)u(M_{i+1}) \\ &= (4n-2)(u(i+2) + u(2n+1-i) + u(2n+i)) \\ &= \left(2n - 1 - \left\lfloor \frac{i+1}{2} \right\rfloor\right) + \left(2n - 1 - \left\lfloor \frac{2n-i}{2} \right\rfloor\right) + n \\ &= (2n - 1 - \lceil i/2 \rceil) + (n - 1 + \lceil i/2 \rceil) + n \\ &= 4n - 2. \end{aligned}$$

□

Define the $\mathrm{MMSscore}$ of an allocation as the maximum α such that it is an α-MMS allocation. Formally, for an allocation $A := (A_1, \ldots, A_n)$,

$$\mathrm{MMSscore}(A) := \min_{i=1}^{n} \frac{v_i(A_i)}{\mathrm{MMS}_i}.$$

Theorem 2. *Let $\mathcal{I}$ be the fair division instance of Example 1. Let $S_1 := \{1\}$, $S_2 := \{n, n+1\}$, $S_3 := \{2n-1, 2n, 2n+1\}$, $S_4 := \{1, 2n+1\}$. Consider a fair division algorithm that either outputs* `bagFill`$(\mathcal{I}, \alpha)$ *for some α, or allocates the set S_k, for some $k \in [4]$, to an agent i, and allocates the remaining goods to the remaining agents in an unspecified way. Let A be the allocation output by this algorithm. Then*

$$\mathrm{MMSscore}(A) \leq \frac{3n}{4n-2} = \frac{3}{4} + \frac{3}{8n-4}.$$

Proof. $u(S_1) = 1/2$, $u(S_2) = u(S_4) = (3n-1)/(4n-2)$, and $u(S_3) = 3n/(4n-2)$. Hence, if the algorithm allocates S_k to an agent i, for some $k \in [4]$, then that agent will get a bundle of value at most $3n/(4n-2)$.

Now suppose that the algorithm outputs `bagFill`$(\mathcal{I}, \alpha)$. Every bag initially has value $\tau := (3n-1)/(4n-2)$. If $\alpha \leq \tau$, then no bag receives any more items, and each agent gets a bag of value τ. If $\alpha > \tau$, then we run out of goods and `bagFill` fails (i.e., returns `null`), since there are n bags but only $n-1$ goods in $[m] \setminus [2n]$. □

6 Conclusion

In fair division of indivisible goods, MMS is one of the most popular notions of fairness, and determining (tight lower and upper bounds on) the maximum α for which α-MMS allocations are guaranteed to exist is an important open problem.

To gain a better understanding of this problem, we thoroughly studied Garg and Taki's [2021] algorithm for obtaining $3/4$-MMS allocations. We considerably simplified its analysis and our techniques helped improve the best-known MMS approximation factor to $\frac{3}{4} + \min(\frac{1}{36}, \frac{3}{4(4n-1)})$. Furthermore, we presented a tight example that reveals a fundamental barrier towards improving the MMS approximation guarantee using techniques in [Garg and Taki, 2021].

Acknowledgements

Jugal Garg and Eklavya Sharma were supported by NSF Grant CCF-1942321.

References

[Akrami *et al.*, 2023] Hannaneh Akrami, Jugal Garg, Eklavya Sharma, and Setareh Taki. Simplification and improvement of MMS approximation. *arXiv:2303.16788v1*, 2023.

[Amanatidis *et al.*, 2016] Georgios Amanatidis, Georgios Birmpas, and Evangelos Markakis. On truthful mechanisms for maximin share allocations. In *International Joint Conference on Artificial Intelligence*, pages 31–37, 2016.

[Amanatidis *et al.*, 2017a] Georgios Amanatidis, Georgios Birmpas, George Christodoulou, and Evangelos Markakis. Truthful allocation mechanisms without payments: Characterization and implications on fairness. In *ACM Conference on Economics and Computation*, pages 545–562, 2017.

[Amanatidis *et al.*, 2017b] Georgios Amanatidis, Evangelos Markakis, Afshin Nikzad, and Amin Saberi. Approximation algorithms for computing maximin share allocations. *ACM Transactions on Algorithms (TALG)*, 13(4):1–28, 2017.

[Aziz *et al.*, 2017] Haris Aziz, Gerhard Rauchecker, Guido Schryen, and Toby Walsh. Algorithms for max-min share fair allocation of indivisible chores. In *AAAI Conference on Artificial Intelligence*, 2017.

[Aziz *et al.*, 2019] Haris Aziz, Bo Li, and Xiaowei Wu. Strategyproof and approximately maxmin fair share allocation of chores. In *International Joint Conference on Artificial Intelligence*, pages 60–66, 2019.

[Babaioff and Feige, 2022] Moshe Babaioff and Uriel Feige. Fair shares: Feasibility, domination and incentives. In *ACM Conference on Economics and Computation*, page 435. Association for Computing Machinery, 2022.

[Babaioff *et al.*, 2021] Moshe Babaioff, Tomer Ezra, and Uriel Feige. Fair-share allocations for agents with arbitrary entitlements. In *ACM Conference on Economics and Computation*, pages 127–127, 2021.

[Barman and Krishnamurthy, 2020] Siddharth Barman and Sanath Kumar Krishnamurthy. Approximation algorithms for maximin fair division. *ACM Transactions on Economics and Computation (TEAC)*, 8(1):1–28, 2020.

[Barman *et al.*, 2018] Siddharth Barman, Arpita Biswas, Sanath Krishnamurthy, and Yadati Narahari. Groupwise maximin fair allocation of indivisible goods. In *AAAI Conference on Artificial Intelligence*, 2018.

[Barman *et al.*, 2019] Siddharth Barman, Ganesh Ghalme, Shweta Jain, Pooja Kulkarni, and Shivika Narang. Fair division of indivisible goods among strategic agents. In *International Conference on Autonomous Agents and Multi-Agent Systems*, page 1811–1813, 2019.

[Bei *et al.*, 2022] Xiaohui Bei, Ayumi Igarashi, Xinhang Lu, and Warut Suksompong. The price of connectivity in fair division. *SIAM Journal on Discrete Mathematics*, 36(2):1156–1186, 2022.

[Biswas and Barman, 2018] Arpita Biswas and Siddharth Barman. Fair division under cardinality constraints. In *IJCAI*, pages 91–97, 2018.

[Bouveret and Lemaître, 2016] Sylvain Bouveret and Michel Lemaître. Characterizing conflicts in fair division of indivisible goods using a scale of criteria. *Autonomous Agents and Multi-Agent Systems*, 30(2):259–290, 2016.

[Budish, 2011] Eric Budish. The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes. *Journal of Political Economy*, 119(6):1061–1103, 2011.

[Chaudhury *et al.*, 2021] Bhaskar Ray Chaudhury, Telikepalli Kavitha, Kurt Mehlhorn, and Alkmini Sgouritsa. A little charity guarantees almost envy-freeness. *SIAM Journal on Computing*, 50(4):1336–1358, 2021.

[Farhadi *et al.*, 2019] Alireza Farhadi, Mohammad Ghodsi, Mohammad Taghi Hajiaghayi, Sebastien Lahaie, David Pennock, Masoud Seddighin, Saeed Seddighin, and Hadi Yami. Fair allocation of indivisible goods to asymmetric agents. *Journal of Artificial Intelligence Research*, 64:1–20, 2019.

[Feige and Norkin, 2022] Uriel Feige and Alexey Norkin. Improved maximin fair allocation of indivisible items to three agents. *arXiv:2205.05363*, 2022.

[Feige *et al.*, 2021] Uriel Feige, Ariel Sapir, and Laliv Tauber. A tight negative example for MMS fair allocations. In *International Conference on Web and Internet Economics*, pages 355–372. Springer, 2021.

[Garg and Taki, 2021] Jugal Garg and Setareh Taki. An improved approximation algorithm for maximin shares. *Artificial Intelligence*, page 103547, 2021.

[Garg *et al.*, 2018] Jugal Garg, Peter McGlaughlin, and Setareh Taki. Approximating maximin share allocations. In *Symposium on Simplicity in Algorithms (SOSA 2019)*, volume 69, pages 20:1–20:11, 2018.

[Gates *et al.*, 2020] Vael Gates, Thomas L. Griffiths, and Anca D. Dragan. How to be helpful to multiple people at once. *Cognitive Science*, 44, 2020.

[Ghodsi *et al.*, 2018] Mohammad Ghodsi, MohammadTaghi HajiAghayi, Masoud Seddighin, Saeed Seddighin, and Hadi Yami. Fair allocation of indivisible goods: Improvements and generalizations. In *ACM Conference on Economics and Computation*, pages 539–556, 2018.

[Gourvès and Monnot, 2019] Laurent Gourvès and Jérôme Monnot. On maximin share allocations in matroids. *Theoretical Computer Science*, 754:50–64, 2019.

[Hosseini and Searns, 2021] Hadi Hosseini and Andrew Searns. Guaranteeing maximin shares: Some agents left behind. In *International Joint Conference on Artificial Intelligence*, pages 238–244, 2021.

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

[Kurokawa *et al.*, 2016] David Kurokawa, Ariel D Procaccia, and Junxing Wang. When can the maximin share guarantee be guaranteed? In *AAAI Conference on Artificial Intelligence*, 2016.

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

[Li and Vetta, 2021] Zhentao Li and Adrian Vetta. The fair division of hereditary set systems. *ACM Transactions on Economics and Computation (TEAC)*, 9(2):1–19, 2021.

[Procaccia and Wang, 2014] Ariel D Procaccia and Junxing Wang. Fair enough: Guaranteeing approximate maximin shares. In *ACM Conference on Economics and Computation*, pages 675–692, 2014.

[Truszczynski and Lonc, 2020] Miroslaw Truszczynski and Zbigniew Lonc. Maximin share allocations on cycles. *Journal of Artificial Intelligence Research*, 69:613–655, 2020.

[Woeginger, 1997] Gerhard J Woeginger. A polynomial-time approximation scheme for maximizing the minimum machine completion time. *Operations Research Letters*, 20(4):149–154, 1997.