

Guaranteeing Maximin Shares: Some Agents Left Behind

Hadi Hosseini
Pennsylvania State University
University Park, PA, USA
hadi@psu.edu

Andrew Searns
Rochester Institute of Technology
Rochester, NY, USA
abs2157@rit.edu

ABSTRACT

The maximin share (MMS) guarantee is a desirable fairness notion for allocating indivisible goods. While MMS allocations do not always exist, several approximation techniques have been developed to ensure that all agents receive a fraction of their maximin share. We focus on an alternative approximation notion, based on the population of agents, that seeks to guarantee MMS for a fraction of agents. We show that no optimal approximation algorithm can satisfy more than a constant number of agents, and discuss the existence and computation of MMS for all but one agent and its relation to approximate MMS guarantees. We then prove the existence of allocations that guarantee MMS for $\frac{2}{3}$ of agents, and devise a polynomial time algorithm that achieves this bound for up to nine agents. A key implication of our result is the existence of allocations that guarantee MMS^[3n/2], *i.e.*, the value that agents receive by partitioning the goods into $\lceil \frac{3}{2}n \rceil$ bundles, improving the best known guarantee of MMS²ⁿ⁻². Finally, we provide empirical experiments using synthetic data.

KEYWORDS

Fair Division, Maximin Share

ACM Reference Format:

Hadi Hosseini and Andrew Searns. 2021. Guaranteeing Maximin Shares: Some Agents Left Behind. In *Appears at the 3rd Games, Agents, and Incentives Workshop (GAIW 2021). Held as part of the Workshops at the 20th International Conference on Autonomous Agents and Multiagent Systems., London, UK, May 2021, IFAAMAS*, 22 pages.

1 INTRODUCTION

Fair division deals with the allocation of a set of resources to a set of agents in a fair manner [6, 13, 21]. One of its most notable application domains deals with the allocation of *indivisible* (and non-shareable) goods. These applications arise, for example, in dividing inheritance among heirs and dispute resolution [6], task assignment or course allocation [7], and have been popularized in recent years due to publicly accessible platforms such as *Spliddit* [18].

The most desirable fairness notion, envy-freeness, requires that each agent weakly prefers his own allocation to that of all other agents. A weaker notion, called proportionality, requires that each agent receives $\frac{1}{n}$ of her valuation of all goods, where n is the total

number of agents. Unfortunately, in the indivisible domain neither of these notions are guaranteed to exist: consider a single good and two interested agents. Neither envy-freeness nor proportionality can be satisfied as one agent is destined to remain empty handed.

A third fairness notion, proposed by [7], is the *maximin share* (MMS) guarantee, which can be seen as a generalization of the *cut-and-choose* protocol [5, 6]. In a nutshell, the maximin share is the value that an agent can guarantee by dividing the goods into n bundles, assuming that all other agents choose a bundle before she does. There is no reason to believe that such a bound can always be satisfied for *all* agents. It turns out that an MMS allocation does not always exist, and even when it exists, computing an MMS partition is intractable [4]. Thus, several approximation techniques were developed to guarantee that each agent receives a fraction of her MMS. While these techniques are compelling, they may leave a large fraction of agents without their MMS guarantee.

We propose to circumvent this obstacle by taking an orthogonal direction based on the degree of fairness in the society of agents. Our goal is to find allocations that give a large fraction of agents their maximin share guarantee, possibly leaving a small fraction of agents behind (since MMS allocations do not always exist). This approach is related to ordinal approximation of MMS and is of interest in several application domains: resources in a hospital must be distributed among tasks/procedures in a manner that a subset of critical tasks can be fully accomplished; senior college students (as a fraction of all students) must be guaranteed seats in courses to avoid graduation delays.

Unfortunately, envy-freeness and proportionality cannot be approximated in this dimension. For instance, one may be interested in minimizing the fraction of envious agents. Envy-freeness (similarly proportionality) may leave almost all but one agent unsatisfied: consider n goods and n agents with identical valuations, having $1 - (n - 1)\epsilon$ value for one good and ϵ for the rest. Any distribution of goods will leave $n - 1$ agents envious. Therefore, focusing on the maximin share guarantee we ask the following questions:

What fraction of agents can be guaranteed their MMS value? And can we compute allocations that guarantee MMS for the majority of the agents?

Our Contributions. We propose a novel fairness framework, called (α, β) -MMS, wherein α fraction of agents receive β approximations of their MMS value and investigate the interplay between the two approximation parameters. We establish the connection between (α, β) -MMS and ordinal approximations of MMS (Proposition 1), and investigate the computational boundaries of α and β as follows:

- **Optimal MMS:** We prove the existence of a family of instances where almost all agents do not receive their MMS by any optimal-MMS allocation. We give an intricate counterexample that uses a *quadratic* number of goods, where $n - 3$ agents ($n - 4$ if n is odd)

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

Appears at the 3rd Games, Agents, and Incentives Workshop (GAIW 2021). Held as part of the Workshops at the 20th International Conference on Autonomous Agents and Multiagent Systems., Aziz, Ceppi, Dickerson, Hosseini, Lev, Mattei, McElfresh, Zick (chairs), May 2021, London, UK. © 2021 Copyright held by the owner/author(s).

do not receive their MMS value in any optimal-MMS allocation (Theorem 1).

- **Computing** ($\frac{n-1}{n}, \beta$)-MMS: We further devise a polynomial-time algorithm achieving (α, β) -MMS when $\alpha = \frac{n-1}{n}$ (Theorem 2). A consequence of this result is a tight approximation for $n \leq 4$ that immediately implies an algorithm for computing MMS^{n+1} for $n < 4$ (Corollary 1).

- **Existence of** ($\frac{2}{3}, 1$)-MMS: We prove the existence of ($\frac{2}{3}, 1$)-MMS for any number of agents (Theorem 3), and provide an algorithm that achieves this bound in polynomial time for $n < 9$ (Theorem 4). A key implication of our result is the existence of allocations that guarantee $\text{MMS}^{\lceil 3n/2 \rceil}$, i.e., the value that an agent receives by partitioning the goods into $\lceil \frac{3}{2}n \rceil$ bundles (Corollary 2). Our result significantly improves the best known guarantee of MMS^{2n-2} [1].

On the experimental front, we show that a simplified, and thus polynomial-time, variant of our algorithm satisfies a large fraction of agents on the most ‘difficult’ instances and this fraction grows as the ratio of goods to agents increases.

Related Work. On a high level, our approach is related to notions defined to measure the degree of fairness in a society of agents whether it pertains to minimizing the maximum or sum of envy [9, 10, 27], minimizing envy ratio [24], balancing the amount of envy experienced in a society [33], or promoting fairness through equitable allocations where agents receive the same level of utility [14, 31]. Another closely related line of work suggests the notion of counting instances of envy (pairs of agents)—among several other plausible measures—as opposed to those that measure the intensity of envy [12]. Similar ideas were briefly mentioned by [11] and [25]; all regarding the degree of envy in a society.

On a technical level, the non-existence of MMS allocations [23] and its intractability [4, 34] has given rise to a number of approximation techniques. These algorithms guarantee that each agent receives an approximation of their maximin share. Recently, [26] gave a Polynomial Time Approximation Scheme (PTAS) for a notion defined as *optimal-MMS*, that is, the largest value, β , for which each agent i receives the value of βMMS_i . Since the number of possible partitions is finite, an optimal-MMS allocation always exists, and it is an MMS allocation if $\beta \geq 1$. The current best results guarantee $\beta \geq 2/3$ [15, 23] and $\beta \geq 3/4$ [16, 17] in general, and $\beta \geq 7/8$ [2] and $\beta \geq 8/9$ [19] in the case of three agents.

2 PRELIMINARIES

Let $N = \{1, \dots, n\}$ be a set of agents and M denote a set of m indivisible goods, where $m > n$. We denote the value of agent $i \in N$ for good $g \in M$ by $v_i(g) \geq 0$. We assume that the valuation functions are *additive*; that is, for each subset $G \subseteq M$, $v_i(G) = \sum_{g \in G} v_i(g)$. An **instance** of the problem is $I = \langle N, M, V \rangle$ where V is the valuation profile of agents. An allocation $A = (A_1, \dots, A_n)$ is an n -partition of M that allocates the bundle of goods in A_i to each agent $i \in N$.

Definition 1 (Envy-freeness). An allocation A is *envy-free* (EF) if for every pair of agents $i, j \in N$, $v_i(A_i) \geq v_i(A_j)$. An allocation A is *envy-free up to one good* (EF1) if for every pair of agents $i, j \in N$ such that $A_j \neq \emptyset$, there exists some good $g \in A_j$ such that $v_i(A_i) \geq v_i(A_j \setminus \{g\})$. An allocation A is *envy-free up to any good*

(EFX) if for every pair $i, j \in N$ such that $A_j \neq \emptyset$, for any good $\forall g \in A_j$, $v_i(A_i) \geq v_i(A_j \setminus \{g\})$. These definitions are due to Foley [13], Budish [7], and Caragiannis et al. [8] respectively.

Definition 2 (Maximin Share Guarantee). Let $\Pi_k(M)$ denote the set of k -partitions of M . The **k -maximin share guarantee** of agent $i \in N$ on $\Pi_k(M)$ is

$$\text{MMS}_i^k(M) = \max_{(A_1, A_2, \dots, A_k) \in \Pi_k(M)} \min_{j \in [k]} v_i(A_j),$$

where $[k] = \{1, \dots, k\}$. Intuitively, this is the minimum value that can be guaranteed if agent i partitions the goods into k bundles and chooses the least valued bundle.

An allocation $A = (A_1, \dots, A_k) \in \Pi_k(M)$ is an *MMS^k allocation* if and only if $\forall i \in N$, $v_i(A_i) \geq \text{MMS}_i^k(M)$. Note that $\text{MMS}_i^n(M) \leq \frac{v_i(M)}{n}$ since proportionality implies MMS. When it is clear from the context, we write MMS_i instead of $\text{MMS}_i^n(M)$ and simply refer to it as agent i 's *MMS value*.

Definition 3 (Optimal-MMS). While an MMS allocation does not always exist, an optimal relaxation of MMS guarantees that agents receive a fraction of their MMS value [26]. Given an instance $I = \langle N, M, V \rangle$, the optimal-MMS value is defined by

$$\lambda^*(I) = \max_{(A_1, \dots, A_n) \in \Pi_n(M)} \min_i \frac{v_i(A_i)}{\text{MMS}_i^n}.$$

By definition, an allocation that gives each agent $\lambda^*(I)$ fraction of its $\text{MMS}_i^n(M)$ value is always guaranteed to exist.

We now provide a set of relevant lemmas and observations.

Ordered instance. An instance is *ordered* when all agents agree on the linear ordering of the goods, irrespective of their valuations. Formally, I is an *ordered instance* if there exists an ordering of goods, (g_1, g_2, \dots, g_m) such that for all agents $i \in N$ we have $v_i(g_1) \geq v_i(g_2) \geq \dots \geq v_i(g_m)$. Bouveret and Lemaître [4] showed that ordered instances are the ‘hardest’ for achieving MMS. In fact, these instances are the only known structures for which MMS does not exist [23]. The next lemma states that given an *unordered* instance, it is always possible to generate a corresponding ordered instance in polynomial time. Furthermore, if the ordered instance admits an MMS allocation, the original instance also admits an MMS allocation which can be computed in polynomial-time.

Lemma 1 (Barman and Krishna Murthy [3]). *Let $I' = \langle N, M, V' \rangle$ be an ordered instance constructed from the original instance $I = \langle N, M, V \rangle$. Given allocation A' on I' , a corresponding allocation A on I can be computed in polynomial time such that for all $i \in N$, $v_i(A_i) \geq v'_i(A'_i)$.*

Scale invariance. The scale invariance property of MMS states that if an agent's valuations are scaled by a factor, then its MMS value scales by the same factor.

Lemma 2 (Ghodsi et al. [17]). *Let $I = \langle N, M, V \rangle$ be an instance and $c > 0$ be a real scalar. Let $I' = \langle N, M, V' \rangle$ be constructed so that $v'_i(g) = cv_i(g)$ for all $g \in M$. Then $\text{MMS}_i^k(M) = c \text{MMS}_i^k(M)$.*

Thus, an instance $I = \langle N, M, V \rangle$ and a real value k can be scaled to form a new instance $I' = \langle N, M, V' \rangle$, such that for each agent $i \in N$, $v'_i(g) = \frac{k}{v_i(M)} v_i(g)$, and $v'_i(M) = k$.

Valid reduction. We call the act of removing a set $A_i \subseteq M$ of goods and an agent i a valid reduction if the following two conditions hold: i) $v_i(A_i) \geq \text{MMS}_i^n(M)$ and ii) $\forall j \in N \setminus \{i\}, \text{MMS}_j^{n-1}(M \setminus A_i) \geq \text{MMS}_j^n(M)$.

Lemma 3 (Garg et al. [15]). *Given an ordered instance $I = \langle N, M, V \rangle$ with $|N| = n$ such that $\text{MMS}_i^n \leq 1$, if $v_i(\{g_n, g_{n+1}\}) \geq 1$, then removing $A_i = \{g_n, g_{n+1}\}$ and agent i forms a valid reduction. Similarly, the removal of $\{g_1\}$ and agent i forms a valid reduction if $v_i(\{g_1\}) \geq 1$.*

Normalized instance. An instance is normalized if all of the following properties hold: i) the instance is ordered, ii) it is scaled so that $v_i(M) = n$, and iii) it is reduced so that $v_i(g_1) < 1$ and $v_i(\{g_n, g_{n+1}\}) < 1$.

By combining Lemma 1, 2, and 3, we may assume that instances are normalized. We prove this claim in Lemma 4.

3 APPROXIMATING MAXIMIN SHARE

We introduce a fairness concept that allows for interpolation between two dimensions in approximating MMS pertaining to the fraction of agents α that receive β approximation of their maximin share.

Definition 4 ((α, β) -MMS). An allocation A guarantees (α, β) -MMS if $\alpha \in (0, 1]$ fraction of agents receive at least their $\beta \in (0, 1]$ approximation of their MMS_i^n . Formally, given an instance $I = \langle N, M, V \rangle$, an allocation A guarantees (α, β) -MMS if there exists a subset $N' \subseteq N$ with $|N'| \geq \lfloor \alpha |N| \rfloor$ such that for all $i \in N', v_i(A_i) \geq \beta \text{MMS}_i^n$. We say that (α, β) -MMS exists if for any instance $I = \langle N, M, V \rangle$, for every subset $N' \subseteq N$ of agents such that $|N'| = \lfloor \alpha |N| \rfloor$, there exists an allocation A such that for all $i \in N', v_i(A_i) \geq \beta \text{MMS}_i^n$.

Previous MMS approximation results can be seen in this context as efforts to tighten the approximation bound for all agents ($\alpha = 1$). Notably, greedy algorithms exist to compute $(1, \frac{2}{3})$ -MMS [3, 15] and $(1, \frac{3}{4})$ -MMS [17] allocations.

Remark 1. It is crucial to highlight two key distinctions: First, in contrast to previous works [28, 29, 32], the definition of (α, β) -MMS existence does not only hold for a fixed subset of agents. Rather, it is a stronger concept that holds for every subset of $\lfloor \alpha n \rfloor$ agents. Second, (α, β) -MMS enables a social planner to pre-select the $N' \subseteq N$ of $\lfloor \alpha n \rfloor$ agents— independent of their preferences— according to some priority ordering over the agents or by selecting the agents uniformly at random. For instance, a higher priority may be given to senior college students; a practice that is already common in most course allocation procedures.

3.1 $(\alpha, 1)$ -MMS Implies MMS^k for $k \geq \lceil \frac{n}{\alpha} \rceil$

Budish [7] showed that approximating the competitive equilibrium from equal incomes (A-CEEI) guarantees $\text{MMS}^{n+1}(M)$, i.e. adding a dummy agent and asking all agents to partition the goods into $n + 1$ bundles. However, this result does not imply the existence of MMS^{n+1} in allocating indivisible goods because allocations achieved by A-CEEI may have excess supply or excess demands. This approach can result in infeasible allocations in fair division settings that do not allow the addition of excess goods. Therefore, the existence of MMS^k for $n + 1 \leq k \leq 2n - 2$ remains an open

problem. In Proposition 1 we show the relation between MMS^k with (α, β) -MMS.

Proposition 1. *The existence of $(\alpha, 1)$ -MMS implies the existence of MMS^k for $k \geq \lceil \frac{n}{\alpha} \rceil$.*

PROOF. Suppose that $(\alpha, 1)$ -MMS exists. Given an instance $I = \langle N, M, V \rangle$ with n agents, we construct an instance I' from I by adding $\lceil \frac{n}{\alpha} \rceil - n$ dummy agents. Therefore, $|N'| = \lceil \frac{n}{\alpha} \rceil$. Since $(\alpha, 1)$ -MMS exists, for every subset $N'' \subseteq N'$ of size $\lfloor \alpha |N'| \rfloor$, there exists an allocation which satisfies $(\alpha, 1)$ -MMS on that set of agents. Thus, we may choose the $\lfloor \alpha |N'| \rfloor$ agents to contain exactly the well-defined original set of agents in N , that is, $N'' := N$. Hence, each agent $i \in N$ receives at least $v_i(A_i) \geq \text{MMS}_i^{|N'|}$, which implies $\text{MMS}_i^{\lceil \frac{n}{\alpha} \rceil}$ for agents in N . \square

3.2 Failure of Optimal MMS Algorithms

There are two primary motivations behind approximating parameter α (the fraction of agents). First, Proposition 1 states that fixing $\beta = 1$ immediately implies an ordinal approximation of MMS by partitioning goods into $k > n$ bundles. Second, our next theorem shows that maximizing the value of β for all agents may result in only a small fraction of agents (α) receiving their MMS value. Theorem 1 shows that there exists a family of instances where any optimal-MMS allocation can only give a small constant number of agents their MMS value. Therefore, an optimal-MMS algorithm (e.g. Nguyen et al. [26]) will result in an (α, λ^*) -MMS allocation such that α goes to zero as the number of agents, n , increases.

THEOREM 1. *For every $n \geq 4$, there exists an instance with $O(n^2)$ goods where every optimal-MMS allocation guarantees at most 3 (4 if n is odd) agents their MMS value.*

PROOF SKETCH. The proof is by a construction inspired by those proposed by Kurokawa et al. [22, 23], but it includes a few intricate modifications.¹ We illustrate our construction by separating agents into two groups who value items according to perturbations of a common valuation matrix. Any optimal-MMS allocation must allocate items according to the rows or columns the $n \times n$ valuation matrix. Half of the agents have $\text{MMS}_i = 1$ by selecting the rows of the matrix; these agents have value $1 - \epsilon$ for each column of the matrix except the last. Similarly, the other half of the agents form an MMS partition by selecting the columns of the matrix. Any optimal-MMS allocation must give each agent at least $1 - \epsilon$ value but this can only be achieved if the allocation corresponds to either the rows or the columns of the matrix. Thus half of the agents (except one lucky agent who receives the last row/column) are destined to receive only $1 - \epsilon < \text{MMS}_i$ value.

We extend this counterexample to a constant number of satisfied agents by dividing agents into $\lceil \frac{n}{2} \rceil$ groups. We guarantee that only one group can be fully satisfied at a time using an intricate generalization of the valuation matrix to higher dimensions. The tensors in our construction are partitioned according to $\lceil \frac{n}{2} \rceil$ sets of sheets (See Fig. 1b). Each group of two agents selects a different set of slices to compute $\text{MMS}_i = 1$. For all other slices (except the last slice

¹The details of the construction, the necessary lemmas, and proofs in this section are relegated to the appendix.

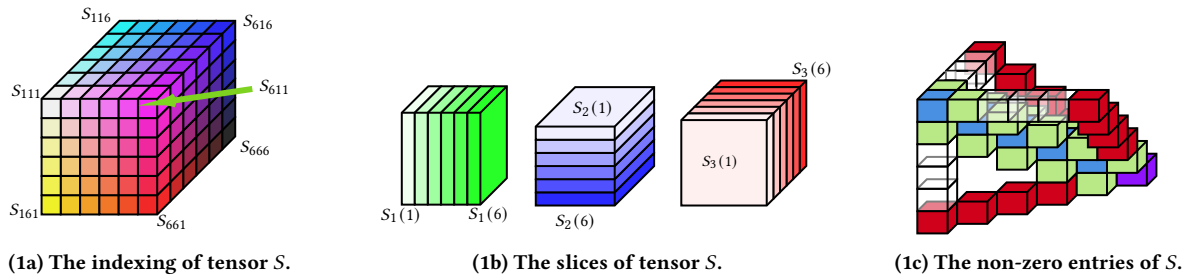


Figure 1: The shape and slices of order 3 tensors with dimensions $(6 \times 6 \times 6)$.

in each set: $S_j(n)$, these agents will have $1 - \epsilon$ value. Fig. 1c depicts the non-zero entries of the valuation tensor. The total number of non-zero goods is proportional to the number of agents multiplied by the number of groups, and thus, is quadratic in the number of agents. \square

Theorem 1 illustrates that if the goal is to reach an optimal-MMS threshold, the fraction of agents, α , who receive their MMS guarantee, $\beta = 1$, goes to zero as the number of agents increases. Thus, we ask for what values of α , $(\alpha, 1)$ -MMS is guaranteed to exist?

4 COMPUTING (α, β) -MMS FOR $n - 1$ AGENTS

Our main goal in this section is devising solutions for computing (α, β) -MMS by removing a single agent.

4.1 A $(\frac{2}{3}, 1)$ -MMS Algorithm for Three Agents

It is worth noting that although for $n = 2$, MMS always exists and can be achieved through the cut-and-choose protocol [4], but computing such an allocation remains hard [2]. Nonetheless, for two agents an allocation that guarantees MMS³ to each agent can be computed in polynomial time through an EF1 allocation [1]. We use this result to obtain the following theorem.

Proposition 2. *For three agents, $(\frac{2}{3}, 1)$ -MMS always exists and can be computed in polynomial time.*

PROOF. By the definition of (α, β) -MMS, we can select any arbitrary subset of agents $N' \subset N$ such that $|N'| = 2$. Then, run an EF1 algorithm on N' , which outputs an allocation A . Both agents in N' are guaranteed to receive their MMS³, i.e., for each $i \in N'$ we have $v_i(A_i) \geq \text{MMS}_i^3$, implying that $\frac{2}{3}$ of agents receive their MMS. By the construction proposed by Kurokawa et al. [23], an MMS allocation does not always exist for three agents; thus, $(\frac{2}{3}, 1)$ -MMS is a tight bound. \square

4.2 A General Algorithm for $n \geq 4$

To extend the analysis of approximate MMS for $n - 1$ agents, we first provide an important lemma that enables us to focus on normalized instances in the remainder of this paper. This lemma states that we can employ valid reductions repeatedly (see Section 2) for computing (α, β) -MMS allocations.

n	(α, β) -MMS
4	$(\frac{3}{4}, 1)$ -MMS
5	$(\frac{4}{5}, \frac{7}{8})$ -MMS
6	$(\frac{5}{6}, \frac{4}{5})$ -MMS
7	$(\frac{6}{7}, \frac{3}{4})$ -MMS

Table 1: Approximation bounds of (α, β) for various $n < 8$.

Lemma 4. *Given an instance $I = \langle N, M, V \rangle$, we can compute a normalized instance $I' = \langle N', M', V' \rangle$ in polynomial time such that any (α, β) -MMS allocation on I' implies (α, β) -MMS on I .*

We now focus attention on designing a polynomial-time algorithm that guarantees β MMS for $n - 1$ agents. The algorithm relies on removing an arbitrary agent and computing an EFX allocation for the remaining $n - 1$ agents. By Lemma 1 an ordered instance can be generated (and easily converted back) from an unordered instance. A simple variant to the envy-graph algorithm satisfies EFX on ordered instances [3]. We show that applying this procedure to normalized and ordered instances satisfies $(\frac{n-1}{n}, \frac{1}{2}(\frac{n+2}{n-1}))$ -MMS. Since β depends on n , we cannot immediately extend this result to any (not normalized) instances. Nonetheless, we prove that together with Lemma 4 and Lemma 1 we can compute $(\frac{n-1}{n}, \frac{1}{2}(\frac{n+2}{n-1}))$ -MMS for any instance in polynomial-time. The full proof of the theorem, along with necessary discussions, is deferred to the appendix.

THEOREM 2. *Given any instance of n agents, $(\frac{n-1}{n}, \frac{1}{2}(\frac{n+2}{n-1}))$ -MMS can be computed in polynomial time.*

By Proposition 1, we can add a dummy agent when $n = 3$ and select the original set of agents to obtain the following.

Corollary 1. *For $n = 3$ agents, computing an allocation satisfying $\text{MMS}_i^4, \forall i \in N$ can be done in polynomial time.²*

Remark 2. Theorem 2 immediately illustrates an intriguing interpolation between the approximation ratios of α and β : for $n < 7$, a better approximation of MMS values (β) can be achieved in polynomial time by sacrificing only one agent. Recall that the best approximation algorithms to date guarantee only $(1, \frac{3}{4})$ -MMS [17] for general additive valuations. Table 1 shows the interpolation between α and β for $n = 4$ to $n = 7$.

²Independently, Aigner-Horev and Segal-Halevi [1] provided an algorithm based on envy-free matchings for computing MMS⁴ for 3 agents.

5 THE EXISTENCE OF $(\frac{2}{3}, 1)$ -MMS ALLOCATIONS

Theorem 2 optimizes the fraction of agents, α , and provides an efficient approach in achieving approximate MMS for $n - 1$ agents. Another plausible, and often practical, approach aims at maximizing the fraction of agents α who receive their MMS guarantee ($\beta = 1$).

It turns out that simple modifications to existing approximation algorithms can guarantee $(\frac{1}{2}, 1)$ -MMS allocations (see the appendix for detailed algorithms and proofs). A key question is whether we can improve this bound for α beyond $\frac{1}{2}$ and show the existence of such allocations. In what follows, we show that $(\frac{2}{3}, 1)$ -MMS exists and discuss an algorithm achieving this bound in polynomial-time when $n < 9$.

Our existence proof of $(\frac{2}{3}, 1)$ -MMS relies on combining techniques of bag-filling, strong normalization, and a variant of the lone divider procedure. Before discussing our main result, we briefly describe these techniques.

Bag-filling. Given a normalized instance, a good is *high-value* for agent i if $v_i(g) \geq \frac{1}{2}$; otherwise it is *low-value*. The bag-filling algorithm is a greedy approach for forming bundles in a normalized instance. An agent initializes a bag with a high-value good, and then adds low-value goods until the bag is worth at least 1. She then picks another high-value good and repeats this process.³ For each bag, the total value never exceeds $1 + \frac{1}{2} = \frac{3}{2}$ since the last added good is of low-value.

Remark 3. Notice that bag-filling alone cannot guarantee $(\frac{2}{3}, 1)$ -MMS because it may ‘run out’ of low-value goods before filling $\lfloor \frac{2n}{3} \rfloor$ bundles if the remaining value consists entirely of high-value goods. Consider an instance with $n = 9$ agents with identical valuations over goods as follows: 5 goods of value 0.99, 5 goods of value 0.01, 1 good of value 0.95, 1 good of value 0.05, 3 goods of value 0.55, and 3 goods of value 0.45. Here, $\text{MMS}_i^n = 1$ for all agents. The high-value goods have value more than 0.5. During bag-filling, there will be three bundles of $\{0.99, 0.45\}$, one bundle of $\{0.99, 0.05\}$, and one bundle of $\{0.99, 0.01\}$. Since only 5 bundles were filled and $\lfloor \frac{2n}{3} \rfloor = 6$, we need to fill one more bundle. Adding all remaining low-value goods to the next high-value good 0.95 yields a total value 0.99. The remaining value is tied up with the high-value goods with value 0.55.

Lemma 4 enabled us to focus on normalized instances with the assumption that either $v_i(M) = n$ or $\text{MMS}_i^n = 1$ for all agents. Our technique in this section requires a slightly stronger assumption.

Strong Normalization. We say an instance is strongly normalized if it is ordered, $v_i(M) = n$, and $\text{MMS}_i^n = 1$. Observe that the definition of strong normalization implies that each bundle of an MMS partition is valued *exactly* 1. Furthermore, this implies that $v_i(g_1) \leq 1$ and that $v_i(g_{n+1}) \leq \frac{1}{2}$.

Lemma 5. *For any additive instance $I = \langle N, M, V \rangle$, there exists another strongly normalized instance $I' = \langle N, M, V' \rangle$ such that I' is ordered and for all $i \in N$, $\text{MMS}_i^n = 1$ and $v_i(M) = n$. Furthermore, an (α, β) -MMS allocation on I' is also an (α, β) -MMS allocation on I .*

³Similar approaches have been used by Garg et al. [15], Garg and Taki [16], Ghodsi et al. [17] to compute $(1, \frac{2}{3})$ -MMS and $(1, \frac{3}{4})$ -MMS allocations. In their algorithms, the bag is filled until *any* agent values it at least 1.

Lemma 5 shows that any instance I can be modified via a non-polynomial-time transformation into a new instance I' that is strongly normalized such that an (α, β) -MMS allocation on I' implies an (α, β) -MMS allocation on I . Hence, we can focus only on strongly normalized instances.

Algorithm description. At its core, our algorithm implements the *lone divider procedure* for indivisible goods on $n' = \lfloor \frac{2n}{3} \rfloor$ of the agents. The lone divider procedure is an extension of the proportional cake-cutting algorithm [30] that leverages non-empty envy-free matchings in a bipartite graph. It proceeds as follows: given n' agents in N' , first, an arbitrary agent divides the goods into n' acceptable bundles $(A_1, \dots, A_{n'})$. Second, we construct a bipartite graph between agents in N' and the bundles $(A_1, \dots, A_{n'})$ wherein each edge connects an agent $i \in N'$ to a bundle A_k such that $v_i(A_k) \geq 1$. Notice that the divider will be adjacent to all bundles. Next, we compute a non-empty envy-free matching, where no unmatched agent is adjacent to a bundle that was assigned to some other agent. A non-empty envy-free matching always exists if $|\mathcal{N}(N')| \geq |N'|$, where $\mathcal{N}(N')$ denotes the set of vertices adjacent to N' [1]. All matched agents receive their bundles and leave the procedure. After the removal of matched agents and goods, we repeat the same procedure over the remaining goods and agents until there are no remaining agents. The formal algorithm, its detailed description, and technical proofs are provided in the appendix.

The primary challenge in proving the existence of $(\frac{2}{3}, 1)$ -MMS allocations lies in showing that in each iteration, given n' remaining agents, the divider agent can form n' bundles valued at least 1. We show that under the strong normalization assumption, the divider is always able to form n' bundles either by pairing high-value goods or through *restricted bag-filling*.

THEOREM 3. *A $(\frac{2}{3}, 1)$ -MMS allocation is guaranteed to exist for any subset of $\lfloor \frac{2n}{3} \rfloor$ agents.*

Our proof relies on two technical cases based on the number of high-value goods (h) for the lone divider. If $h \leq \lfloor \frac{2n}{3} \rfloor$, there are not many high-value goods, and thus, the simple bag-filling guarantees our desired result. The most challenging cases arise when $h > \lfloor \frac{2n}{3} \rfloor$, that is, when there are too many high-value goods. Let $s = h - \lfloor \frac{2n}{3} \rfloor$ be the number of excess high-value goods as shown in Figure 2. Here, we show that the divider agent is able to form the necessary bundles (*i.e.* the number of remaining agents) by adopting the following strategies: if $n' \leq s$, we can simply pair the remaining high-value goods since there are at least $2n'$ high-value goods available, which implies n' bundles are valued more than 1. On the other hand, if $n' > s$, we are able to form s bundles by pairing the remaining high-value goods. This means that we still need additional $n' - s$ bundles. The key to handling this step relies on the *restricted bag-filling* procedure with exactly one high-value good per bundle but we restrict the low-value goods used to fill bundles. In restricted bag-filling, we only use the low-value goods from bundles of the divider’s MMS^n partition whose high-value goods are no longer available (since those goods have been allocated in some previous round or assigned during pairing), as highlighted in Figure 2 by teal boxes.

The main reason behind this restriction is to ensure that not too much value is ‘wasted’ in the bag-filling procedure (as discussed

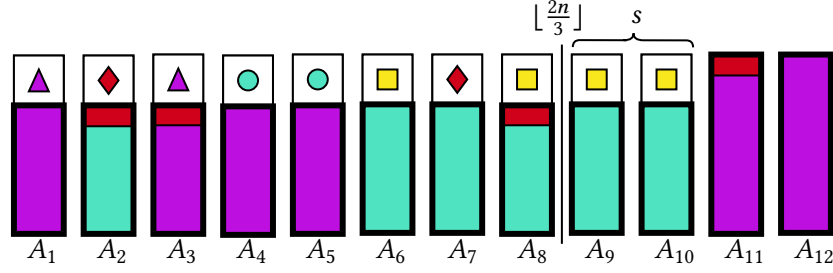


Figure 2: A sample MMS partition for the divider. The tiny shapes (triangle, circles, squares, and diamonds) represent high-value goods and are ordered from g_1 to g_{10} . Note that each bundle contains no more than one high-value good. All pieces indicated in red (including the red diamonds) indicate the goods that were allocated in previous iterations. There are two such high-value goods, so $n' = \lfloor \frac{2n}{3} \rfloor - 2 = 6$. The first step is pairing the lowest $2s$ remaining high-value goods into s bundles: $\{g_6, g_{10}\}$ and $\{g_8, g_9\}$. The second phase is restricted bag-filling that only uses low-value goods from the remainder sets corresponding to already allocated high-value goods (within teal boxes). The solid borders show *all* remainder sets. Next, the remaining goods, shown in purple are used for simple bag-filling to fill the remaining 2 bundles.

in Remark 3). The restricted bag-filling relies on the strong normalization assumption. Thus, each bundle contains at most one high-value goods, and the low-value goods used in the restricted bag-filling have sufficient value.

Our proof follows by showing that if $n' \geq 2s$, it is possible to form $n' - s$ bundles through restricted bag-filling. Otherwise if $n' < 2s$, using restricted bag-filling we can form s bundles. In this case, we still need to form additional $n' - 2s$ bundles, which is possible through simple bag filling by targeting for each bundle to have at most $\frac{3}{2}$ value. In the appendix, we provide formal proofs for each of these steps.

Remark 4. The strong normalization assumption in the proof of Theorem 3 requires modifying the values of some goods depending on an MMS partition for agent i . Furthermore, the restricted bag-filling phase of Theorem 3 selects available low-value goods from specific bundles of an MMS partition for agent i . Because of these dependencies on finding MMS partitions, Theorem 3 does not imply a polynomial time algorithm for $(\frac{2}{3}, 1)$ -MMS.

We show that by relaxing the strong normalization assumption to require only that for each agent $\text{MMS}_i^n \geq 1$, we can devise a polynomial time algorithm for $(\frac{2}{3}, 1)$ -MMS when $n < 9$. Essentially, this algorithm is a simplified variant of our previous procedure that only includes simple bag-filling and pairing to form the required bundles in the lone divider procedure. The detailed algorithm is provided in the appendix. Thus, we aim for a value of at most $\frac{3}{2}$ during bag-filling, which at most wastes $\frac{1}{2}$ value in each bundle. However, the total value of low-value goods is guaranteed to be at least $1 - v_i(g_1)$, so at least one bundle can be allocated during bag-filling. When $n < 9$, adding the pairing phase forms $\lfloor \frac{n}{2} \rfloor + 1 > \lfloor \frac{2n}{3} \rfloor$ bundles.

THEOREM 4. For $n < 9$, $(\frac{2}{3}, 1)$ -MMS can be computed in polynomial time.

Theorem 4 only guarantees a $(\frac{2}{3}, 1)$ -MMS allocation when $n < 9$. We show by construction that this is tight. We construct a family of instances with $n \geq 9$ in which a small error in computing the MMS

bound causes bag-filling to stop before $\lfloor \frac{2n}{3} \rfloor$ bundles have been allocated. The challenge presented here is not unique to our algorithm. Any algorithm that satisfies $(\frac{2}{3}, 1)$ -MMS must be able to detect that $\text{MMS}_i^n < 1$ for all agents. However, this task is intractable even when agents have identical valuations [4].

Remark 5 (Tightness for Theorem 4). Consider $n \geq 9$ agents with identical valuations as follows: $n - 1$ high-value goods of value $1 - \epsilon - \tilde{\epsilon}$, two goods of value $\frac{1}{2} - \epsilon$, one good of value $(n + 1)\epsilon$, and $n - 1$ goods of value $\tilde{\epsilon}$, where $\epsilon \gg \tilde{\epsilon}$. Then, $\text{MMS}_i^n = 1 - \epsilon$ for all agents. During the bag-filling step in Theorem 4, three bundles are allocated accounting for the three goods of value $1 - \epsilon - \tilde{\epsilon}$, both goods of value $\frac{1}{2} - \epsilon$, and the good with value $(n + 1)\epsilon$. After bag-filling, the remaining high-value goods will be paired together to create a total of $\lfloor \frac{n-4}{2} \rfloor$ bundles of paired high-value goods. Thus, this algorithm only satisfies $\lfloor \frac{n}{2} \rfloor + 1$ agents.

Combining this construction with Theorem 4 implies that no smaller counter-example exists. We note that despite this bound, the algorithm is still practical as there is no bound on the number of goods. For example, more than 99% of instances in the *Spliddit* dataset deal with less than 9 agents.

The analysis in Theorem 3 and Theorem 4 are indifferent to the set of agents N' . Consequently, we may select any subset of $\lfloor \frac{2n}{3} \rfloor$ agents to be given their full MMS. An immediate consequence of this result, together with Proposition 1, is that there always exists an allocation of goods such that each agent receives at least the value of $\text{MMS}^{\lceil \frac{3n}{2} \rceil}$. Moreover, this allocation can be computed in polynomial time when $\lceil \frac{3n}{2} \rceil < 9$.

Corollary 2. An allocation satisfying $\text{MMS}^{\lceil \frac{3n}{2} \rceil}$ always exists. Moreover, such allocations can be computed in polynomial time when $n < 6$.

6 EMPIRICAL EVALUATIONS

The polynomial algorithm of Theorem 4 guarantees $(\frac{2}{3}, 1)$ -MMS only for $n < 9$. In this section, we empirically evaluate an extension of this algorithm on instances with up to 50 agents and 200 goods.

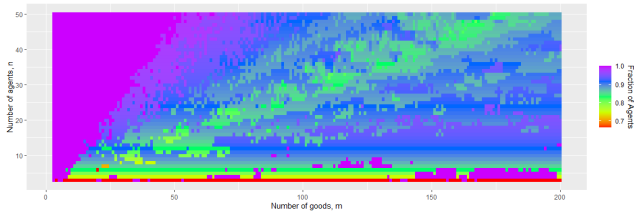


Figure 3: Fraction of agents, α , receiving their maximin share, i.e., $(\alpha, 1)$ -MMS.

This variant of the algorithm does not rely on the strong normalization assumption (see the appendix for the detailed description). It is identical that of Theorem 4 until either no more bundles can be allocated during the lone divider procedure or all $\lfloor \frac{2n}{3} \rfloor$ agents that were initially selected have received a bundle valued at least 1. In the latter case, any remaining goods are distributed among the unselected $n - \lfloor \frac{2n}{3} \rfloor$ agents using bag-filling with priority given to agents who accept the smallest bundles. Notice that this algorithm does not guarantee that all initially selected agents receive their MMS.

We focus on ordered instances—as the most difficult instances in achieving MMS [4]—and generate 1,000 instances for each combination of n and m . Instances are sampled uniformly at random, ordered, and scaled such that $v_i(M) = n$ for all agents $i \in N$.

Figure 3 illustrates the fraction of agents who receive their MMS for $n = 3$ to 50 agents and $m = 3$ to 200 goods. In almost all instances, the algorithm goes beyond the $\frac{2}{3}$ bound: on average across all instances, more than 90% of agents receive their MMS. Moreover, the fraction of agents receiving their MMS improves as either n or m increases. We also observe linear bands where a lower fraction of agents are satisfied due to the ratio of goods to agents. In addition, when $m < 2n$ a large fraction of the agents receive their MMS and removed during the reduction phase of normalization (see Section 2). In the appendix, we provide further discussions and results.

7 DISCUSSION

Theorem 3 proves the existence of $(\frac{2}{3}, 1)$ -MMS for any n , which implies the existence of $\text{MMS}^{\lceil \frac{3n}{2} \rceil}$. Therefore, improving the bound on $(\alpha, 1)$ -MMS for $\alpha > \frac{2}{3}$ and closing the gap between $\text{MMS}^{\lceil \frac{3n}{2} \rceil}$ and MMS^{n+1} is an intriguing future direction. Theorem 4 provides a tractable approach for computing $(\frac{2}{3}, 1)$ -MMS for $n < 9$. Yet, computing such allocations for any n , if possible, will require further techniques to circumvent computing exact MMS bounds. Another interesting avenue for future research is exploring PTAS algorithms that guarantee optimal-MMS [20] for a fraction of agents to achieve $(\alpha, 1)$ -MMS for $\alpha \in [\frac{2}{3}, \frac{n-1}{n}]$.

ACKNOWLEDGMENTS

Hadi Hosseini acknowledges the support from NSF grant #1850076. We thank Sawyer Welden for initial help with the experimentation, and Nisarg Shah for helpful discussions. We are especially grateful to Erel Segal-Halevi for his valuable input that helped us improve

the proof of Theorem 3. We thank the anonymous reviewers for their very helpful comments and suggestions.

REFERENCES

- [1] Elad Aigner-Horev and Erel Segal-Halevi. 2019. Envy-free Matchings in Bipartite Graphs and their Applications to Fair Division. *arXiv preprint arXiv:1901.09527* (2019).
- [2] Georgios Amanatidis, Evangelos Markakis, Afshin Nikzad, and Amin Saberi. 2017. Approximation algorithms for computing maximin share allocations. *ACM Transactions on Algorithms (TALG)* 13, 4 (2017), 52.
- [3] Siddharth Barman and Sanath Kumar Krishna Murthy. 2017. Approximation algorithms for maximin fair division. In *Proceedings of the 2017 ACM Conference on Economics and Computation*. ACM, 647–664.
- [4] Sylvain Bouveret and Michel Lemaître. 2016. Characterizing conflicts in fair division of indivisible goods using a scale of criteria. *Autonomous Agents and Multi-Agent Systems* 30, 2 (01 Mar 2016), 259–290. <https://doi.org/10.1007/s10458-015-9287-3>
- [5] Steven J Brams and Alan D Taylor. 1995. An envy-free cake division protocol. *The American Mathematical Monthly* 102, 1 (1995), 9–18.
- [6] Steven J Brams and Alan D Taylor. 1996. *Fair Division: From cake-cutting to dispute resolution*. Cambridge University Press.
- [7] Eric Budish. 2011. The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes. *Journal of Political Economy* 119, 6 (2011), 1061–1103.
- [8] Ioannis Caragiannis, David Kurokawa, Hervé Moulin, Ariel D Procaccia, Nisarg Shah, and Junxing Wang. 2016. The unreasonable fairness of maximum nash welfare. In *Proceedings of the 2016 ACM Conference on Economics and Computation*. ACM, 305–322.
- [9] Yiling Chen and Nisarg Shah. 2018. *Ignorance is often bliss: Envy with incomplete information*. Technical Report.
- [10] Yann Chevaleyre, Ulle Endriss, Sylvia Estivie, and Nicolas Maudet. 2007. Reaching envy-free states in distributed negotiation settings. In *Proceedings of the 20th international joint conference on Artificial intelligence*. Morgan Kaufmann Publishers Inc., 1239–1244.
- [11] Yann Chevaleyre, Ulle Endriss, and Nicolas Maudet. 2017. Distributed fair allocation of indivisible goods. *Artificial Intelligence* 242 (2017), 1–22.
- [12] Allan Feldman and Alan Kirman. 1974. Fairness and envy. *The American Economic Review* 64, 6 (1974), 995–1005.
- [13] Duncan K. Foley. 1967. Resource allocation and the public sector. *Yale Economic Essays* 7 (1967), 45–98.
- [14] Rupert Freeman, Sujoy Sikdar, Rohit Vaish, and Lirong Xia. 2019. Equitable Allocations of Indivisible Goods. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, 280–286. <https://doi.org/10.24963/ijcai.2019/40>
- [15] Jugal Garg, Peter McLaughlin, and Setareh Taki. 2018. Approximating Maximin Share Allocations. In *2nd Symposium on Simplicity in Algorithms (SOSA 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.
- [16] Jugal Garg and Setareh Taki. 2020. An Improved Approximation Algorithm for Maximin Shares. In *Proceedings of the 21st ACM Conference on Economics and Computation (Virtual Event, Hungary) (EC '20)*. Association for Computing Machinery, New York, NY, USA, 379–380. <https://doi.org/10.1145/3391403.3399526>
- [17] Mohammad Ghodsi, MohammadTaghi HajiAghayi, Masoud Seddighin, Saeed Seddighin, and Hadi Yami. 2018. Fair allocation of indivisible goods: Improvements and generalizations. In *Proceedings of the 2018 ACM Conference on Economics and Computation*. ACM, 539–556.
- [18] Jonathan R Goldman and Ariel D Procaccia. 2014. Spliddit: unleashing fair division algorithms. *SIGecom Exchanges* 13, 2 (2014), 41–46.
- [19] Laurent Gourvès and Jérôme Monnot. 2019. On maximin share allocations in matroids. *Theoretical Computer Science* 754 (2019), 50–64.
- [20] Tobias Heinen, Nhan-Tam Nguyen, Trung Thanh Nguyen, and Jörg Rothe. 2018. Approximation and complexity of the optimization and existence problems for maximin share, proportional share, and minimax share allocation of indivisible goods. *Autonomous Agents and Multi-Agent Systems* 32, 6 (2018), 741–778.
- [21] Hadi Hosseini, Sujoy Sikdar, Rohit Vaish, Jun Wang, and Lirong Xia. 2020. Fair Division through Information Withholding. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20)*. 2014–2021.
- [22] David Kurokawa, Ariel D Procaccia, and Junxing Wang. 2016. When can the maximin share guarantee be guaranteed?. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. AAAI Press, 523–529.
- [23] David Kurokawa, Ariel D Procaccia, and Junxing Wang. 2018. Fair enough: Guaranteeing approximate maximin shares. *Journal of the ACM (JACM)* 65, 2 (2018), 8.
- [24] Richard J Lipton, Evangelos Markakis, Elchanan Mossel, and Amin Saberi. 2004. On approximately fair allocations of indivisible goods. In *Proceedings of the 5th ACM conference on Electronic commerce*. ACM, 125–131.

- [25] Arnon Netzer, Amnon Meisels, and Roie Zivan. 2016. Distributed envy minimization for resource allocation. *Autonomous Agents and Multi-Agent Systems* 30, 2 (2016), 364–402.
- [26] Nhan-Tam Nguyen, Trung Thanh Nguyen, and Jörg Rothe. 2017. Approximate solutions to max-min fair and proportionally fair allocations of indivisible goods. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 262–271.
- [27] Trung Thanh Nguyen and Jörg Rothe. 2014. Minimizing envy and maximizing average Nash social welfare in the allocation of indivisible goods. *Discrete Applied Mathematics* 179 (2014), 54–68.
- [28] Kathryn Nyman, Francis Edward Su, and Shira Zerbib. 2020. Fair division with multiple pieces. *Discrete Applied Mathematics* 283 (2020), 115 – 122. <https://doi.org/10.1016/j.dam.2019.12.018>
- [29] Josué Ortega. 2018. Social integration in two-sided matching markets. *Journal of Mathematical Economics* 78 (2018), 119–126.
- [30] Jack Robertson and William Webb. 1998. *Cake-Cutting Algorithms: Be Fair if You Can*. (1998).
- [31] Sebastian Schneckeburger, Britta Dorn, and Ulle Endriss. 2017. The Atkinson inequality index in multiagent resource allocation. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 272–280.
- [32] Erel Segal-Halevi and Warut Suksompong. 2019. Democratic fair allocation of indivisible goods. *Artificial Intelligence* 277 (2019), 103167.
- [33] Koichi Tadenuma and William Thomson. 1995. Refinements of the no-envy solution in economies with indivisible goods. *Theory and Decision* 39, 2 (1995), 189–206.
- [34] Gerhard J Woeginger. 1997. A polynomial-time approximation scheme for maximizing the minimum machine completion time. *Operations Research Letters* 20, 4 (1997), 149–154.

Appendix

A MATERIAL OMITTED FROM SECTION 3.2

This section includes all the detailed discussions on constructing tensors S and T , the relevant lemmas, and the missing proofs.

We begin with the generalized version of Theorem 1. We observe that when $d = \lfloor \frac{n}{2} \rfloor$ the following theorem becomes Theorem 1.

THEOREM 5. *For any $n \geq 4$ and any $d \leq \lfloor \frac{n}{2} \rfloor$, there exists an instance with $O(dn)$ goods where any optimal-MMS allocation guarantees at most $\lceil \frac{n}{d} \rceil + 1$ agents their MMS value.*

Our construction is broken into three components: a tensor S , a tensor T , and d groups of agents. The entries of tensor S are chosen such that equi-partitions of the entries correspond to a set of slices of S . An equi-partition is a partition of the entries of a tensor where each set of the partition has the same sum. Here, all equi-partitions refer to n -partitions. Tensor T provides perturbation so that the only equi-partitions of $S + T$ are aligned slices.

All tensors in our construction are order d with dimensions $(n \times n \times \dots \times n)$.⁴ We say that tensor S is indexed by a d -tuple (x_1, \dots, x_d) corresponding to entry $S[x_1, \dots, x_d]$ or S_{x_1, \dots, x_d} for short. The $(d-1)$ -order slices of tensor S along dimension j are given by $S_j(1)$ to $S_j(n)$ where $S_j(i) = \{S_{x_1, \dots, x_d} \mid x_j = i\}$. We also define $S_{i; x_j = k}$ to be the entry where all indices are i except index $x_j = k$. For example, $S_{i; x_1 = n} = S_{nii \dots i}$. Tensor addition is entry-wise $((S+T)_{x_1, x_2, \dots, x_d} = S_{x_1, x_2, \dots, x_d} + T_{x_1, x_2, \dots, x_d})$. Unless otherwise specified, the entries of a tensor have value 0.

Figure 1a depicts the indexing of an order 3 tensor with dimensions $(6 \times 6 \times 6)$. Figure 1b illustrates the distinct sets of slices of such a tensor. Lastly, Figure 1c depicts the non-zero entries of tensors S and T in our construction when $n = 6$ and $d = 3$.

A.1 Construction of Tensor S

Let S be an order d tensor whose non-zero entries are as follows:

- For $1 \leq i < n$ and $j \in [d]$, $S_{ii \dots i} = \frac{d^{n-i} - 1}{d^{n-i}}$
- For $1 \leq i < n$ and $j \in [d]$, $S_{i; x_j = n} = \frac{1}{d-1} \cdot \frac{1}{d^{n-i}}$
- $S_{nn \dots n} = 1 - \sum_{i=1}^{n-1} \frac{1}{d-1} \cdot \frac{1}{d^{n-i}}$

Lemma 6. *Each $(d-1)$ -order slice $S_j(i)$ has sum 1.*

PROOF. Consider the non-zero entries within slice $S_j(i)$. First when $i \neq n$, the only non-zero entries of $S_j(i)$ are $S_{ii \dots i}$ and $S_{i; x_k = n}$ for each $k \neq j$. Thus:

$$\begin{aligned} \sum_{x \in S_j(i)} x &= S_{ii \dots i} + \sum_{k \neq j} S_{i; x_k = n} \\ &= \frac{d^{n-i} - 1}{d^{n-i}} + \sum_{k \neq j} \frac{1}{d-1} \cdot \frac{1}{d^{n-i}} \\ &= \frac{d^{n-i} - 1}{d^{n-i}} + (d-1) \frac{1}{d-1} \cdot \frac{1}{d^{n-i}} \\ &= \frac{d^{n-i} - 1}{d^{n-i}} + \frac{1}{d^{n-i}} \\ &= 1 \end{aligned}$$

When $i = n$, the non-zero entries of slice $S_j(i)$ are $S_{nn \dots n}$ and $S_{k; x_j = n}$ for each $k \neq n$. Thus:

$$\begin{aligned} \sum_{x \in S_j(n)} x &= S_{nn \dots n} + \sum_{k \neq n} S_{k; x_j = n} \\ &= 1 - \sum_{i=1}^{n-1} \frac{1}{d-1} \cdot \frac{1}{d^{n-i}} + \sum_{k=1}^{n-1} \frac{1}{d-1} \cdot \frac{1}{d^{n-k}} \\ &= 1 \end{aligned}$$

□

The combinatorial structure of S has a few other desirable properties. First observe that the entries on the main diagonal ($S_{ii \dots i}$) all have value greater than $\frac{1}{2}$. When considering equi-partitions of the entries of S , these entries must all be included in separate sets. Furthermore, the entries off the main diagonal increase when approaching $S_{nn \dots n}$ in such a way that all d of the entries with value $\frac{1}{d-1} \cdot \frac{1}{d^{n-i}}$ are required to account for a single entry with value $\frac{1}{d-1} \cdot \frac{1}{d^{n-(i+1)}}$.

⁴When referring to a tensor, the “order” is the number of dimensions that the tensor is embedded in. The “dimensions” of a tensor, however, refers to the number of entries along each axis, similar to the dimensions of a matrix.

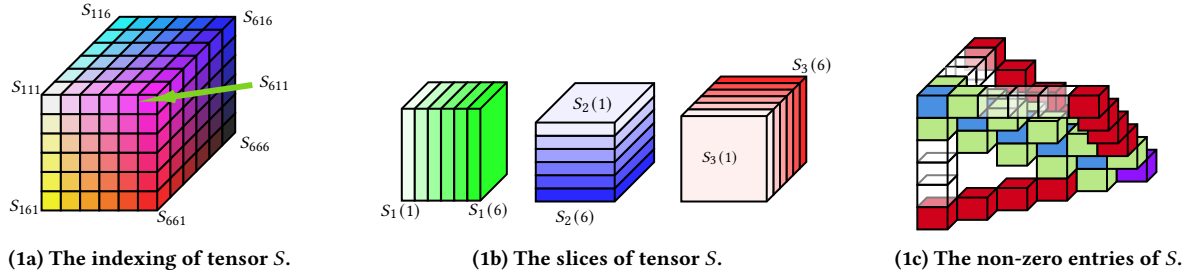


Figure 1: The shape and slices of order 3 tensors with dimensions $(6 \times 6 \times 6)$.

Lemma 7. Any equi-partition of the non-zero entries of S corresponds to a set of slices $\{S_{j_1}(1), S_{j_2}(2), \dots, S_{j_n}(n)\}$.

PROOF. Lemma 6 implies that a set of slices of the form $\{S_{j_1}(1), S_{j_2}(2), \dots, S_{j_n}(n)\}$ forms an equi-partition. In order to show that these equi-partitions are unique, consider the set which contains $S_{11\dots 1}$. Since $1 - S_{11\dots 1} = \frac{1}{d^{n-1}} < \frac{1}{d-1} \cdot \frac{1}{d^{n-2}}$. Thus the only entries which can be included with $S_{11\dots 1}$ are of the form $\frac{1}{d-1} \cdot \frac{1}{d^{n-1}}$. Importantly, $d-1$ of these entries uniquely identify a $(d-1)$ -order slice $S_{j_1}(1)$ where $S_{1;x_{j_1}=n}$ is the missing entry with value $\frac{1}{d-1} \cdot \frac{1}{d^{n-1}}$.

By induction, with a similar argument at each step, each successive entry on the main diagonal must be included in a set which corresponds exactly to a slice $S_{j_i}(i)$. \square

A.2 Construction of Tensor T

In Lemma 7, the slices of S are not necessarily aligned slices. For example, in the case of three dimensions, the first slice might be frontal while the second one might be vertical. Since all slices have the same sum, there is no reason an agent wouldn't compute MMS by taking slices at random. In order to rectify this, we construct tensor T so that equi-partitions of $S + T$ correspond only to aligned slices.

Let tensor T be an order d tensor with non-zero values as follows:

- For $1 \leq i < n-1, j \in [d], T_{i;x_j=n} = -r_{ij}$
- For $1 \leq i < n-1, j \in [d], T_{i;x_j=i+1} = u_{ij}$
- For $j \in [d], T_{n-1;x_j=n} = x_j$
- $T_{nn\dots n} = z_1$

Where

- $r_{ij} = \epsilon^{d(n-1)-di-j+1}$
- $u_{1j} = r_{1j}$
- $u_{ij} = \frac{1}{d-1} \left(\sum_{k \neq j}^d -u_{i-1,k} \right) + \frac{d-2}{d-1} u_{i-1,j} + r_{ij} \approx r_{ij}$
- $x_j = u_{n-1,j} - r_{n-1,j}$
- $z_j = \left(\sum_{i=1}^{n-2} r_{ij} \right) - x_j$

It is important to observe that the entries of T are all proportional to powers of a very small ϵ term. While some of these values are negative, the negative entries only occur in indices where S has positive value. Thus if ϵ is small enough, $S + T$ has only non-negative values. Another important observation is that the slices of T all have sum 0.

Lemma 8. Each $(d-1)$ -order slice $T_j(i)$ has sum 0.

PROOF. We first address the case of $i = 1$. Slice $T_j(1)$ contains the non-zero entries $T_{1;x_k=n}$ for each $k \neq j$ and $T_{1;x_k=2}$ for each $k \neq j$. For each $k \neq j$, the entry $T_{1;x_k=n} = -r_{1j}$ and $T_{1;x_k=2} = u_{1j}$. Since $u_{1j} = -r_{1j}$ the sum of these two entries is 0. Adding over all $k \neq j$ yields that the first slice (independent of j) has sum 0.

Now consider slice $T_j(i)$ for $1 < i < n - 1$. The non-zero entries of T in this slice are $T_{i;x_k=n}$ and $T_{i;x_k=i+1}$ for $k \neq j$ and $T_{i-1;x_j=i}$. Adding these entries together yields:

$$\begin{aligned}
& \sum_{x \in T_j(i)} x \\
&= \sum_{k \neq j}^d T_{i;x_k=i+1} + \sum_{k \neq j}^d T_{i;x_k=n} + T_{i-1;x_j=i} \\
&= \sum_{k \neq j}^d (u_{ik}) + \sum_{k \neq j}^d (-r_{ik}) + u_{i-1,j} \\
&= \sum_{k \neq j}^d \left(\frac{1}{d-1} \left(\sum_{l \neq k}^d -u_{i-1,l} \right) + \frac{d-2}{d-1} u_{i-1,k} + r_{ik} \right) \\
&\quad + \sum_{k \neq j}^d (-r_{ik}) + u_{i-1,j} \\
&= \sum_{k \neq j}^d \left(\frac{1}{d-1} \left(\sum_{l \neq k}^d -u_{i-1,l} \right) + \frac{d-2}{d-1} u_{i-1,k} \right) + u_{i-1,j} \\
&= \frac{1}{d-1} \left(\sum_{k \neq j}^d \sum_{l \neq k}^d (-u_{i-1,l}) + (d-2) \sum_{k \neq j}^d (u_{i-1,k}) \right) + u_{i-1,j} \\
&= \frac{1}{d-1} ((d-1)(-u_{i-1,j})) + u_{i-1,j} = 0
\end{aligned}$$

The last step follows because each of the $(d-1)$ summations (for each $k \neq j$) adds a single $-u_{i-1,j}$ term. For each $k \neq j$, $(d-2)$ of the summations (all but $l = k$ of the $(d-1)$ inner summations) add a $-u_{i-1,k}$ term.

We next address the case where $i = n - 1$. In this case, we observe that $i + 1 = n$ so the only entries in this slice are $T_{n-1;x_k=n}$ for $k \neq j$ and $T_{n-2;x_j=n-1}$. Adding these entries together yields:

$$\begin{aligned}
& \sum_{x \in T_j(n-1)} x \\
&= \sum_{k \neq j}^d T_{n-1;x_k=n} + T_{n-2;x_j=n-1} \\
&= \sum_{k \neq j}^d (x_k) + u_{n-2,j} \\
&= \sum_{k \neq j}^d (u_{n-1,k} - r_{n-1,k}) + u_{n-2,j} \\
&= \sum_{k \neq j}^d \left(\left(\frac{1}{d-1} \left(\sum_{l \neq k}^d -u_{n-2,l} \right) \right. \right. \\
&\quad \left. \left. + \frac{d-2}{d-1} u_{i-2,k} + r_{n-1,k} \right) - r_{n-1,k} \right) + u_{n-2,j} \\
&= \sum_{k \neq j}^d \left(\frac{1}{d-1} \left(\sum_{l \neq k}^d -u_{n-2,l} \right) + \frac{d-2}{d-1} u_{i-2,k} \right) + u_{n-2,j} \\
&= \frac{1}{d-1} \left(\sum_{k \neq j}^d \sum_{l \neq k}^d (-u_{n-2,l}) + (d-2) \sum_{k \neq j}^d (u_{i-2,k}) \right) + u_{n-2,j} \\
&= \frac{1}{d-1} ((d-1)(-u_{n-2,j})) + u_{n-2,j} = 0
\end{aligned}$$

We last address the case where $i = n$. In this case, the non-zero entries in this slice are $T_{i;x_j=n}$ for each $i \in [n-2]$, $T_{n-1;x_j=n}$, and $T_{nn\dots n}$. Adding these entries together yields:

$$\begin{aligned} \sum_{x \in T_j(n)} x &= \sum_{k=1}^{n-2} T_{k;x_j=n} + T_{n-1;x_j=n} + T_{nn\dots n} \\ &= \sum_{k=1}^{n-2} (-r_{kj}) + x_j + z_1 \end{aligned}$$

If $j = 1$ then the term $\sum_{k=1}^{n-2} (-r_{kj}) + x_j = -z_1$ so this trivially sums to 0. If $j \neq 1$, then we first observe that the sum of all entries of T is 0 by adding the sums of the slices $\{T_1(1), T_1(2), \dots, T_1(n)\}$. Since $\sum_{i=1}^n \sum_{x \in T_j(i)} x = 0$ and $\sum_{k=1}^{n-1} \sum_{x \in T_j(k)} x = 0$, subtracting the two equations gives $\sum_{x \in T_j(n)} x = 0$. \square

Combining Lemmas 7 and 8 together immediately shows that any set of aligned slices of $(S+T)$ forms an equi-partition of the non-zero entries where each set has sum 1. We now show that these partitions are unique. For intuition, the key point in our construction is that $(d-1)$ of the $T_{i;x_j=i+1}$ terms all lie in the same slice $T_j(i)$. Once j has been chosen for $T_j(1)$ the extra term $T_{1;x_j=2}$ lies in slice $T_j(2)$.

Lemma 9. *The only equi-partitions of the non-zero entries of $(S+T)$ correspond to aligned slices.*

PROOF. First observe that if ϵ is sufficiently small then the structure of Lemma 7 still holds. Consider first the set of an equi-partition which contains $(S+T)_{11\dots 1}$. In order for this set to have value 1, it must have $(d-1)$ of the entries which receive $\frac{1}{d-1} \cdot \frac{1}{d^{n-1}}$ value from S . Again, selecting $(d-1)$ of these uniquely characterizes a slice $(S+T)_j(1)$. Observe that these entries have negative perturbations from T corresponding to $-r_{1k}$ for each $k \neq j$. In order for these perturbations to be offset so that the set containing $(S+T)_{11\dots 1}$ has sum 1, the corresponding entries with values u_{1k} for $k \neq j$ must be included in this set. These entries correspond exactly to the slice $(S+T)_j(1)$.

As in the proof of Lemma 7, we observe that the entries on the main diagonal $(S+T)_{ii\dots i}$ must be in separate sets. Likewise, because ϵ is small, a similar argument enforces that $(d-1)$ of the entries with value $\frac{1}{d-1} \cdot \frac{1}{d^{n-i}}$ must be included in this set. We address the negative perturbations (from T) of these entries by induction. Observe that the smallest perturbations from T occur in the earliest slices $((1,1) = \arg \min_{i,j} r_{ij})$. Note also that $(u_{1j}, -r_{1j})$ is the only pair of perturbations from T which are excluded from slice $(S+T)_j(1)$. A similar argument for slice $(S+T)_j(1)$ holds for slice $(S+T)_j(2)$: in order for the set containing $(S+T)_{22\dots 2}$ to have value 1, it must also have $(d-1)$ entries with the $-r_{2k}$ perturbations from T . Since $\epsilon^2 \ll \epsilon$, each $-r_{2k}$ entry can only be offset with entries with perturbation u_{2k} . (All but one of the smaller perturbation entries have already been assigned to slice 1.) However, by Lemma 8, we see that adding together all $k \neq l$ entries of the form $-r_{2k}$ and u_{2k} leaves value u_{1l} (since $u_{1l} + \sum_{k \neq l}^d u_{2k} + \sum_{k \neq l}^d -r_{2k} = 0$). Observe that only a single entry of the form u_{1l} is available: notably u_{1j} . Thus the set which contains $(S+T)_{22\dots 2}$ corresponds to the slice $(S+T)_j(2)$.

By induction on this same argument we see that each set containing the entry $(S+T)_{ii\dots i}$ for $1 \leq i < n$ must correspond to a slice $(S+T)_j(i)$ for some fixed j (after j is chosen for the first slice). We conclude this proof by observing that the remaining entries all lie in the last slice $(S+T)_j(n)$ and have total sum 1 by Lemmas 7 and 8. \square

With the details of S and T , we are now ready to prove the main theorem. We restate it here for convenience.

THEOREM 5. *For any $n \geq 4$ and any $d \leq \lfloor \frac{n}{2} \rfloor$, there exists an instance with $O(dn)$ goods where any optimal-MMS allocation guarantees at most $\lceil \frac{n}{d} \rceil + 1$ agents their MMS value.*

PROOF. Lemma 9 implies that there are d unique equi-partitions of the entries of $(S+T)$. We group agents arbitrarily into d groups of size at least 2 and at most $\lceil \frac{n}{d} \rceil$ each. We next show how to perturb $(S+T)$ for each group so that each group has a unique equi-partition. Furthermore, we enforce that the only entry with positive perturbation for all agents is the last entry on the main diagonal. Let P^j be an order d tensor with non-zero values as follows:

- For $1 \leq i < n$, $P^j_{i;x_j=n} = -\tilde{\epsilon}$
- $P^j_{nn\dots n} = (n-1)\tilde{\epsilon}$

The entries where P^j is negative correspond to entries where $(S+T)$ has positive value. Thus $(S+T+P^j)$ has only non-negative entries if $\tilde{\epsilon}$ is sufficiently small. Observe that the slices $\{(S+T+P^j)_j(1), (S+T+P^j)_j(2), \dots, (S+T+P^j)_j(n)\}$ form an equi-partition of the entries of $(S+T+P^j)$ as P^j has non-zero values only in the last slice $(S+T+P^j)_j(n)$ and the total sum of its non-zero entries is 0. If $\tilde{\epsilon}$ is sufficiently small (so that the structure of $S+T$ dominates when creating an equi-partition), then none of the other sets of slices of $(S+T+P^j)$ forms an exact equi-partition as every slice except for the last contains exactly one of the $-\tilde{\epsilon}$ perturbations.

We now create an instance $I = (N, M, V)$ where M contains one good for each non-zero entry of $(S+T)$. Agents in group j value the goods according to the values of $(S+T+P^j)$. Since every agent is able to form an equi-partition of the entries of their valuation tensor, all agents have $MMS_j = 1$. By allocating goods according to slices, each agent receives at least $(1 - \tilde{\epsilon})$ of their MMS value. Since $\tilde{\epsilon}$ is very small, the only way all agents can receive at least $1 - \tilde{\epsilon}$ value is if the goods are allocated according to aligned slices (Lemma 9). Under this

Algorithm 1: Computing MMS_i^3 for 2 agents

Input : Instance $I = \langle N, M, V \rangle$ such that $n = |N| = 2$
Output : Allocation $A = (A_1, A_2)$

- 1 $I' = \langle N, M, V' \rangle \leftarrow$ scaled instance such that for each $i \in N$, $v'_i(M) = n + 1$; // Lemma 2
- 2 **if** $\exists g \in M$ such that $v'_i(g) \geq 1$ for some $i \in N$ **then**
- 3 $A_i \leftarrow \{g\}$;
- 4 $A_j \leftarrow M \setminus \{g\}$ for $j \neq i \in N$;
- 5 **else**
- 6 $A \leftarrow$ Run an EF1 algorithm on I' ;

allocation, at most one group and a single additional agent who receives the last slice (since $(S + T + P^j)_j(n)$ has positive perturbation $(n - 1)\epsilon$) receive their MMS value. Lastly we observe that there only $O(dn)$ goods in M . \square

Again we note that if $d = \lfloor \frac{n}{2} \rfloor$ in the above theorem, then we see that only 3 agents (4 if n is odd) receive their full MMS in any optimal-MMS allocation.

B MATERIAL OMITTED FROM SECTION 4

B.1 An alternative algorithm for achieving MMS^3 for two agents

Lemma 10 (Bouveret and Lemaître [4]). *Given an instance $I = \langle N, M, V \rangle$, for any agents $i \in N$ and any good $g \in M$, we have $\text{MMS}_i^{n-1}(M \setminus g) \geq \text{MMS}_i^n(M)$.*

THEOREM 6. *For 2 agents, there exists a polynomial time algorithm that computes an allocation satisfying MMS_i^3 for all $i \in N$.*

PROOF. Algorithm 1 starts by scaling the valuations in instance I such that for all agents $i \in N$ and all goods in $g \in M$ we have $v'_i(g) = \frac{3}{v_i(M)} v_i(g)$. By scale invariance (Lemma 2) and since $\text{MMS}_i \leq \frac{v_i(M)}{n}$, we have $\forall i \in N$, $\text{MMS}_i^3 \leq 1$.

Case I. Consider two agents $i, j \in N$. Suppose agent i is allocated a good g with value $v'_i(g) \geq 1$ (Line 2), then by definition $v'_i(A_i) \geq \text{MMS}_i^3$. The remaining agent j receives $A_j = M \setminus \{g\}$. By definition of MMS, $v_j(M \setminus \{g\}) \geq \text{MMS}_j^2(M \setminus \{g\})$. By Lemma 10, $\text{MMS}_j^2(M \setminus \{g\}) \geq \text{MMS}_j^3(M)$. Therefore, $v_j(M \setminus \{g\}) \geq \text{MMS}_j^3(M)$.

Case II. Suppose that after scaling the valuations, no agent values any good greater or equal to 1. Then, Algorithm 1 proceeds by running an EF1 algorithm (e.g. round-robin), which results in an EF1 allocation. For each agent $i \in N$, let g^* be agent i 's highest valued good in agent j 's bundle. By EF1 we have $v_i(A_i) \geq v_i(A_j) - v_i(g^*)$. Adding $v_i(A_i)$ to both sides of the inequality, and since there are two agents, we get $2v_i(A_i) \geq v_i(M) - v_i(g^*)$. Since $v_i(M) = 3$, $v_i(g^*) < 1$, we have that $v_i(A_i) \geq \frac{3}{2} - \frac{1}{2}$, implying that $v_i(A_i) \geq \text{MMS}_i^3$ according to Lemma 2 and the MMS bound, that is, $\text{MMS}_i \leq \frac{v_i(M)}{n}$. \square

B.2 Proof of Lemma 4

Lemma 4. *Given an instance $I = \langle N, M, V \rangle$, we can compute a normalized instance $I' = \langle N', M', V' \rangle$ in polynomial time such that any (α, β) -MMS allocation on I' implies (α, β) -MMS on I .*

PROOF. Given an instance $I = \langle N, M, V \rangle$, we can compute a normalized instance $I' = \langle N', M', V' \rangle$ as follows: first, order the instance I according to Lemma 1. Call this instance $\hat{I} = \langle \hat{N}, \hat{M}, \hat{V} \rangle$ such that $\hat{n} = \hat{N}$ and $\hat{m} = \hat{M}$. Then, in each iteration by Lemma 2, we scale the instance such that $\bar{v}_i(\hat{M}) = \bar{n}$. Then, apply one of the valid reductions among $\{g_1\}$ or $\{g_{\hat{n}}, g_{\hat{n}+1}\}$. An agent receives a bundle according to the valid reduction (Lemma 3). We then update the instance \hat{I} by removing the allocated agent and their bundle and relabel goods from g_1 to $g_{\hat{m}}$. The procedure stops once there are no valid reductions remaining.

Let $k \leq n$ be the number of agents who were allocated and removed during the previous phase. By the definition of a valid reduction, the MMS of each agent only weakly increases after any reduction. Thus each agent who receives a bundle during the reduction phase receives at least their $\text{MMS}_i^n(M)$. Any (α, β) -MMS allocation on the resulting normalized instance I' guarantees that $\lfloor \alpha n' \rfloor$ agents receive at least βMMS_i^n because $n' \leq n$. Combining this with the fact that $\lfloor \alpha(n - k) \rfloor + k \geq \lfloor \alpha n \rfloor$ implies that at least $\lfloor \alpha n \rfloor$ agents receive βMMS_i^n in the ordered instance corresponding to I . By Lemma 1 when we unorder the allocation, the resulting allocation for I satisfies (α, β) -MMS. \square

B.3 Proof of Theorem 2

We first show that a modification to the envy-graph algorithm [24] provides a bounded MMS guarantee to $n - 1$ on normalized instances.

The modified envy-graph algorithm. The modified envy-graph algorithm (due to Barman and Krishna Murthy [3]) proceeds as follows: given a normalized instance, in each round the highest valued good (breaking ties arbitrarily) among the remaining goods is assigned to an agent that is not envied by any agent. One such agent is guaranteed to exist by resolving cyclic envy relations in the envy-graph.

Lemma 11 (Barman and Krishna Murthy [3]). *For any ordered instance, the modified envy-graph algorithm guarantees EFX.*

Lemma 12. *Given a normalized instance, the modified envy-graph algorithm satisfies $(\frac{n-1}{n}, \frac{1}{2}(\frac{n+2}{n-1}))$ -MMS.*

PROOF. We first show that no agent receives an empty bundle when $m \geq n$. Let A be the allocation produced by the modified envy graph algorithm agents in $N' \subset N$ such that $|N'| = n - 1$. Consider the first $n - 1$ goods in an ordered instance. In each of the first $n - 1$ rounds of the envy graph algorithm, there exists at least one unenvied agent. By assumption, we know that $v_i(g_1) < 1$, $v_i(M) = n$. This implies that $\forall i \in N$, $v_i(g_n) > 0$; otherwise there exists at least one good g with value $v_i(g) \geq \frac{n}{n-1} > 1$. Since for all $i \in N'$, $v_i(g_n) > 0$ and the fact that the instance is ordered, all agents also have positive value for all goods ordered before g_n , i.e., for all $i \in N$, $v_i(g_1) \geq v_i(g_2) \geq \dots \geq v_i(g_n) > 0$. Thus any agent who has not yet received any good envies each agent who has already received a good. Since the algorithm does not allocate any good to an envied agent, no agent is allocated a second good until all bundles are allocated at least one good, and at most one bundle can receive two goods from $\{g_1, \dots, g_n\}$. Therefore, all bundles are nonempty after allocating the first n good, i.e., $\forall i \in N'$, $|A_i| > 0$.⁵

Next, we must show that each agent's bundle is valued $v_i(A_i) \geq \frac{1}{2}(\frac{n+2}{n-1})$ once all goods are allocated.

Let S be the set of agents who receive a single good in the allocation A and let $s = |S|$. Notice that $s < n$, since one agent does not participate in the algorithm ($|N'| = n - 1$). By Lemma 11, A satisfies EFX, i.e., for all $j \in [n - 1]$, $v_i(A_i) \geq v_i(A_j \setminus \hat{g}_j)$ for any good $\hat{g}_j \in A_j$. By summing over the set of agents who received at least two goods we have,

$$((n - 1) - s)v_i(A_i) \geq v_i(M \setminus (\bigcup_{j \in S} A_j)) - \sum_{\substack{j \neq i \\ j \notin S}}^{n-1} v_i(\hat{g}_j).$$

Since $v_i(M) = n$ and $v_i(\bigcup_{j \in S} A_j) \leq s$, we can write:

$$((n - 1) - s)v_i(A_i) \geq (n - s) - \sum_{\substack{j \neq i \\ j \notin S}}^{n-1} v_i(\hat{g}_j)$$

The \hat{g}_j are distinct and each worth at most $v_i(g_n)$. Thus the two with highest value among them are worth at most $v_i(\{g_n, g_{n+1}\}) < 1$. Pulling these items out of the sum, and upper-bounding the remaining goods $v_i(\hat{g}_j) \leq v_i(g_{n+2}) \leq v_i(g_{n+1}) < \frac{1}{2}$, we can write:

$$\begin{aligned} ((n - 1) - s)v_i(A_i) &\geq (n - s) - 1 - \frac{1}{2}(n - s - 4) \\ ((n - 1) - s)v_i(A_i) &\geq (n - s) - \frac{1}{2}(n - s - 2) \\ v_i(A_i) &\geq \frac{n - s}{n - s - 1} - \frac{1}{2}(\frac{n - s - 2}{n - s - 1}) \\ v_i(A_i) &\geq \frac{1}{2}(\frac{n - s + 2}{n - s - 1}) \\ v_i(A_i) &\geq \frac{1}{2}(\frac{n + 2}{n - 1}). \end{aligned}$$

Notice that the last step holds because $s < n$. □

THEOREM 2. *Given any instance of n agents, $(\frac{n-1}{n}, \frac{1}{2}(\frac{n+2}{n-1}))$ -MMS can be computed in polynomial time.*

PROOF. Consider an instance of the problem I with n agents. We begin by computing a normalized instance I' from I following the procedure of Lemma 4. On the normalized instance, we compute an $(\frac{n'-1}{n'}, \frac{1}{2}(\frac{n'+2}{n'-1}))$ -MMS allocation (Lemma 12). We now prove that the resulting allocation satisfies $(\frac{n-1}{n}, \frac{1}{2}(\frac{n+2}{n-1}))$ -MMS.

The key observation is that the value of β is dependent on n and is monotonic decreasing: $\frac{n'+2}{n'-1} \geq \frac{n+2}{n-1}$ when $n' \leq n$ and that $\frac{n'-1}{n'}n' + (n - n') = n - 1$. Thus each agent who is allocated a bundle in the reduced instance I' receives at least $\text{MMS}_i^{n'}$. Furthermore, each agent allocated a bundle during the reductions of Lemma 4 receives their $\text{MMS}_i^{n'} \geq \text{MMS}_i^n$. In total $n - 1$ agents receive at least $\frac{1}{2}(\frac{n+2}{n-1})\text{MMS}_i^n$. After unordering, by Lemma 1 the resulting allocation satisfies $(\frac{n-1}{n}, \frac{1}{2}(\frac{n+2}{n-1}))$ -MMS. □

⁵Indeed, two agents may exchange bundles through cycle resolution when the n th good is allocated.

C MATERIAL OMITTED FROM SECTION 5

C.1 Computing $(\frac{1}{2}, 1)$ -MMS allocations

There are several plausible ways to achieve MMS for half of the agents, that is, $(\frac{1}{2}, 1)$ -MMS. We first provide a simple algorithm relying on EF1, and then show a generalization that enables us to exploit the already known MMS approximation guarantees.

Proposition 3. *If $\forall i \in N, \forall g \in M, v_i(g) < \frac{v_i(M)}{n}$, then any EF1 algorithm on half of the agents guarantees $(\frac{1}{2}, 1)$ -MMS.*

PROOF. Consider an EF1 allocation A that allocates goods to the half of the agents. This implies that $v_i(A_i) \geq v_i(A_j) - v_i(g_j)$ for some good $g_j \in A_j$. As the EF1 allocation is over $\lfloor \frac{n}{2} \rfloor$ agents, the union over all bundles is M . Thus

$$\begin{aligned}
\sum_{j=1}^{\lfloor \frac{n}{2} \rfloor} v_i(A_i) &\geq \sum_{j=1}^{\lfloor \frac{n}{2} \rfloor} (v_i(A_j) - v_i(g_j)) \\
\lfloor \frac{n}{2} \rfloor v_i(A_i) &\geq v_i(M) - \sum_{j=1}^{\lfloor \frac{n}{2} \rfloor} v_i(g_j) \\
\lfloor \frac{n}{2} \rfloor v_i(A_i) &\geq v_i(M) - \sum_{j=1}^{\lfloor \frac{n}{2} \rfloor} \frac{v_i(M)}{n} \\
\lfloor \frac{n}{2} \rfloor v_i(A_i) &\geq v_i(M) - \lfloor \frac{n}{2} \rfloor \frac{v_i(M)}{n} \\
v_i(A_i) &\geq \frac{1}{\lfloor \frac{n}{2} \rfloor} \left(v_i(M) - \lfloor \frac{n}{2} \rfloor \frac{v_i(M)}{n} \right) \\
v_i(A_i) &\geq \frac{v_i(M)}{\lfloor \frac{n}{2} \rfloor} - \frac{v_i(M)}{n} \\
v_i(A_i) &\geq \frac{v_i(M)}{\frac{n}{2}} - \frac{v_i(M)}{n} \\
v_i(A_i) &\geq \frac{2v_i(M)}{n} - \frac{v_i(M)}{n} \\
v_i(A_i) &\geq \frac{v_i(M)}{n} \\
v_i(A_i) &\geq \text{MMS}_i^n
\end{aligned}$$

□

The immediate consequence of Proposition 3 is an intuitive algorithm for satisfying $(\frac{1}{2}, 1)$ -MMS: apply valid reductions according to Lemma 3 followed by applying any EF1 algorithm (e.g. Round Robin or Maximum Nash welfare) on half of the remaining agents in the reduced instance.

Next we show that running any β -MMS approximation algorithm with $\beta \geq \frac{1}{2}$ on half of the agents is sufficient to guarantee $(\frac{1}{2}, 1)$ -MMS.

Proposition 4. *Any algorithm satisfying at least $(1, \frac{1}{2})$ -MMS also guarantees $(\frac{1}{2}, 1)$ -MMS given half of the agents.*

PROOF. Consider an algorithm that guarantees $(1, \beta)$ -MMS where $\beta \geq \frac{1}{2}$. In any MMS_i^n partition each bundle in the partition has value at least MMS_i^n . If these n bundles are paired together to form $\lfloor \frac{n}{2} \rfloor$ bundles, the resulting allocation forms a lower bound for $\text{MMS}_i^{\lfloor n/2 \rfloor}$. Each of these $\lfloor n/2 \rfloor$ bundles has value at least 2MMS_i^n which implies that $\text{MMS}_i^{\lfloor n/2 \rfloor}(M) \geq 2\text{MMS}_i^n(M)$. When this algorithm is run on half of the agents, those agents are guaranteed $\beta \text{MMS}_i^{\lfloor n/2 \rfloor} \geq \frac{1}{2} \text{MMS}_i^{\lfloor n/2 \rfloor}(M) \geq \frac{1}{2} 2\text{MMS}_i^n(M)$. Thus, the resulting allocation satisfies $(\frac{1}{2}, 1)$ -MMS. □

Remark 6. An EF1 algorithm alone is not sufficient for computing $(1, \frac{1}{2})$ -MMS, and thus, cannot be directly used for satisfying $(\frac{1}{2}, 1)$ -MMS. Consider 3 agents with identical valuations over 7 goods:

	g_1	g_2	g_3	g_4	g_5	g_6	g_7
$v_i(g_j)$	0.99	0.99	0.4	0.4	0.2	0.01	0.01

The allocation $A = (\{g_1, g_4, g_7\}, \{g_2, g_5\}, \{g_3, g_6\})$ is EF1 but not $(1, \frac{1}{2})$ -MMS.

Algorithm 2: A $(\frac{2}{3}, 1)$ -MMS algorithm

Input : An ordered, strongly normalized instance $I = \langle N, M, V \rangle$
Output : A $(\frac{2}{3}, 1)$ -MMS allocation on I

- 1 Let N' be a subset of $\lfloor \frac{2n}{3} \rfloor$ agents from N ;
- 2 Let $M' = M$;
- 3 **while** $|N'| > 0$ **do**
- 4 Select an agent i from N' to be the divider;
- 5 Let $h = |\{g | v_i(g) > \frac{1}{2}\}|$ and $s = \max(h - \lfloor \frac{2n}{3} \rfloor, 0)$;
- 6 **if** $s = 0$ **then**
- 7 Bag-fill n' bundles;
- 8 **else**
- 9 Pair $\min(n', s)$ bundles;
- 10 **if** $s < n'$ **then**
- 11 Restricted bag-fill $\min(n' - s, s)$ bundles;
- 12 **if** $s < n' - s$ **then**
- 13 Bag-fill $n' - 2s$ bundles;
- 14 Let $G = (N' \cup D, E)$ be the bipartite graph;
- 15 Find a non-empty envy-free matching \mathcal{M} of G ;
- 16 **foreach** $(j, D_k) \in \mathcal{M}$ **do**
- 17 Allocate D_k to agent j ;
- 18 $N' = N' \setminus \{j\}$;
- 19 $M' = M' \setminus D_k$;

C.2 Proof $(\frac{2}{3}, 1)$ -MMS Existence for $n > 4$

Lemma 13 (Aigner-Horev and Segal-Halevi [1]). *If $|\mathcal{N}(X)| \geq |X|$, then a non-empty envy-free matching exists and can be found in polynomial time.*

Lemma 5. *For any additive instance $I = \langle N, M, V \rangle$, there exists another strongly normalized instance $I' = \langle N, M, V' \rangle$ such that I' is ordered and for all $i \in N$, $\text{MMS}_i^n = 1$ and $v_i(M) = n$. Furthermore, an (α, β) -MMS allocation on I' is also an (α, β) -MMS allocation on I .*

PROOF. By Lemma 2, we can assume without loss of generality that the original instance I is scaled so that $\text{MMS}_i^n = 1$ for all agents $i \in N$. We observe that this scaling implies that $v_i(M) \geq n$ for all agents.

Let (M_1, \dots, M_n) be an MMS_i^n partition for some agent, say agent i , for which $v_i(M) > n$. For each bundle M_j which has value greater than 1, we arbitrarily reduce the value of goods in M_j until $v'_i(M_j) = 1$.⁶ After doing this for each bundle, $v'_i(M) = n$ for agent i . Observe that this procedure does not change the MMS of any agent because once the valuations are changed all bundles have value 1 so $\text{MMS}_i^n = 1$ both before and after the valuations are reduced.

Let $A = (A_1, \dots, A_n)$ be an allocation on instance I' that satisfies (α, β) -MMS. When we add the removed value back to each good to return to the original instance I , the total value of any bundle does not decrease; i.e. for each $S \subseteq M$, $v_i(S) \geq v'_i(S)$. Thus for each agent $v_i(A_j) \geq v'_i(A_j)$. This implies that A also satisfies (α, β) -MMS on I .

We lastly observe that we may order I' to make an ordered instance I'' with $\text{MMS}_i^n = 1$ and $v_i(M) = n$ for all $i \in N$. By Lemma 1, any (α, β) -MMS allocation on I'' implies an (α, β) -MMS allocation on I' which in turn implies an (α, β) -MMS allocation on I . \square

Algorithm 2 details the steps of our algorithm for guaranteeing $(\frac{2}{3}, 1)$ -MMS allocations,

THEOREM 3. *A $(\frac{2}{3}, 1)$ -MMS allocation is guaranteed to exist for any subset of $\lfloor \frac{2n}{3} \rfloor$ agents.*

PROOF. By combining Lemma 1 and Lemma 5, we assume that the given instance is ordered and strongly normalized. Thus for all agents $\text{MMS}_i^n = 1$ and $v_i(g_1) < 1$.

We begin by selecting a subset $N' \subseteq N$ with $n' = |N'| = \lfloor \frac{2n}{3} \rfloor$. Our approach will be through the lone divider technique. At each iteration, we select one remaining agent to partition the remaining goods M' into n' bundles $(D_1, \dots, D_{n'})$ each with value at least 1. We create a bipartite graph with edges connecting the remaining agents N' with the bundles that they value at least 1. There are at least n' acceptable bundles for the agent who formed the partition, so $|\mathcal{N}(N')| \geq |N'|$. Thus by Lemma 13, a non-empty envy-free matching exists. For every edge in the matching, we allocate the bundle to the corresponding agent and then iterate this procedure on the remaining agents and goods. In order to guarantee that this process continues, we require that each bundle contains exactly one good from $\{g_1, \dots, g_{\lfloor \frac{2n}{3} \rfloor}\}$.

⁶It does not matter which goods of M_j are reduced in value.

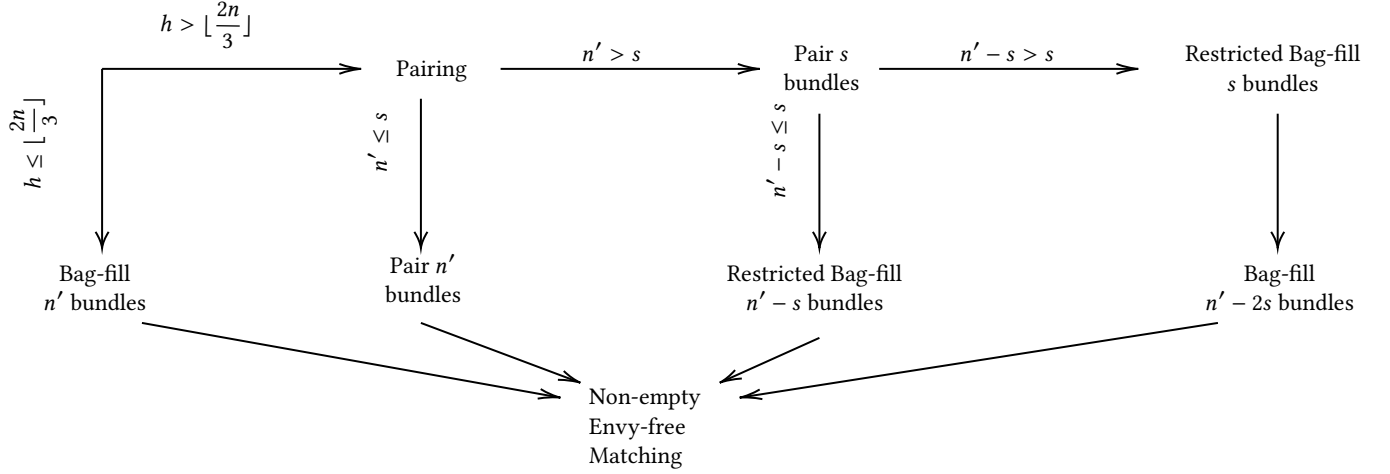


Figure 5: The steps of our algorithm for dividing the remaining goods into bundles at each iteration by the lone divider in the $(\frac{2}{3}, 1)$ -MMS existence proof.

Consider an arbitrary iteration of this procedure and an arbitrary agent, say agent i , who is selected to divide the items. We say that a good g is high-value for i if $v_i(g) > \frac{1}{2}$ and low-value otherwise. Let h be the number of goods that are high-value for agent i . Let $A = (A_1, \dots, A_n)$ be an MMS_i^n partition for agent i . By the strong normalization assumption, $\text{MMS}_i^n = 1$ and $v_i(M) = n$, meaning that the value of each bundle of A is exactly 1. Thus no bundle of A contains more than one high-value good. Suppose without loss of generality that the bundles of A are labeled so that each high-value goods g_j is in the respective bundle A_j for $j \in [h]$. We further define the remainders of these bundles to be $R_j := A_j \setminus \{g_j\}$. These remainder sets have value $r_j = v_i(R_j) = 1 - v_i(g_j)$.

Suppose that $n' = \lfloor \frac{2n}{3} \rfloor - k$ which implies that k bundles $(B_c)_{c=1}^k$ have already been allocated to other agents in previous rounds of the lone divider algorithm. If agent i valued any such bundle B_c at least 1, then agent i would have been envious in the matching where B_c was allocated. Thus $v_i(B_c) < 1$ for each $c \in [k]$. Furthermore, since we maintain the constraint that each bundle in every partition for the lone divider has exactly one good from $\{g_1, \dots, g_{\lfloor \frac{2n}{3} \rfloor}\}$, there are $\lfloor \frac{2n}{3} \rfloor - k = n'$ of these items remaining.

We now consider two cases depending on the number of high-value goods.

Case 1: $h \leq \lfloor \frac{2n}{3} \rfloor$. The total remaining value is at least $n - k$. We let agent i form bundles using bag-filling with each bag initialized with one of the remaining n' goods from $\{g_1, \dots, g_{\lfloor \frac{2n}{3} \rfloor}\}$. The bags are filled with low-value goods until they have value in the range $[1, \frac{3}{2})$. This process is continued until the value from remaining goods is less than 1. Suppose that t bundles are filled this way. Then the total value of all goods is less than $\frac{3}{2}t + 1 > n - k$. Thus $t \geq \frac{2}{3}n - \frac{2}{3}k - \frac{2}{3}$. Since t, n, k are all integers, $t \geq \lfloor \frac{2n}{3} \rfloor - k = n'$.

Case 2: $h > \lfloor \frac{2n}{3} \rfloor$. When $h > \lfloor \frac{2n}{3} \rfloor$, we must be more careful about the remaining high-value goods. We define the number of surplus high-value goods as $s = h - \lfloor \frac{2n}{3} \rfloor$. In any partition of the goods into $\lfloor \frac{2n}{3} \rfloor$ bundles (counting the k bundles $(B_c)_{c=1}^k$), there will necessarily be at least s high-value goods in a bundle which contains another high-value good. Thus our first step is to pair high-value goods.

Step 1: Pairs. Define H^+ to be the $\lfloor \frac{2n}{3} \rfloor - k$ highest-valued remaining goods (those contained in $\{g_1, \dots, g_{\lfloor \frac{2n}{3} \rfloor}\}$). Likewise, define H^- to be the other $h - \lfloor \frac{2n}{3} \rfloor$ high-value goods (those not contained in $\{g_1, \dots, g_{\lfloor \frac{2n}{3} \rfloor}\}$).

Agent i forms $\min(n', s)$ bundles by pairing the lowest-value goods from H^+ with the goods from H^- . If $n' \leq s$ then agent i will form n' bundles by pairing, so we are done.

Otherwise, $s < n'$, and agent i still needs to construct $n' - s$ bundles. There are still a total of $(h - k) - 2s = ((s + \lfloor \frac{2n}{3} \rfloor) - k) - 2s = n' - s$ high-value goods remaining. These goods all fall within $\{g_1, \dots, g_{\lfloor \frac{2n}{3} \rfloor}\}$.

In Figure 6, the high-value goods g_2 and g_7 were allocated in the previous round, so there remains $n' = 6$ agents. The high-value goods g_6, g_8, g_9 , and g_{10} are first paired into two bundles $\{g_6, g_{10}\}$ and $\{g_8, g_9\}$.

Step 2: Restricted Bag-filling. The next step is to construct $\min(n' - s, s)$ bundles from bag-filling using only a subset of the available low-value goods. If $n' - s$ is smaller then there are sufficiently many bundles formed for the envy-free matching and we are done. Otherwise, we now have $2s$ acceptable bundles leaving $n' - 2s$ high-value goods from $\{g_1, \dots, g_{\lfloor \frac{2n}{3} \rfloor}\}$.

For the restricted bag-filling we consider only the low-value goods which fall in bundles of A whose high-value goods have already been accounted for. Thus we define two specific types of *remainder sets* as follows:

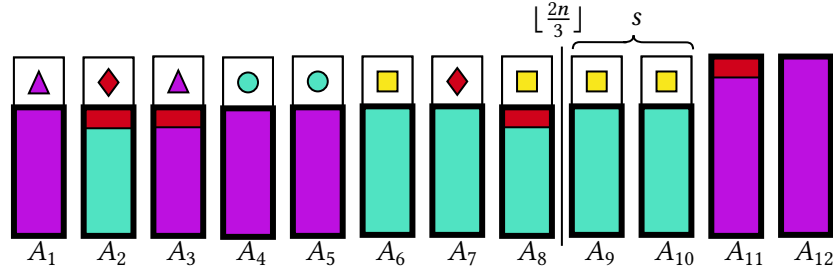


Figure 6: An example MMS partition for the current divider. Sections in red illustrate the goods that were allocated in previous iterations. There are two such goods, so $n' = \lfloor \frac{2n}{3} \rfloor - 2 = 6$. The first step is pairing the lowest $2s$ remaining high-value goods into s bundles: $\{g_6, g_{10}\}$ and $\{g_8, g_9\}$. The second phase is restricted bag-filling. Here, the only low-value goods considered are the remainder sets corresponding to high-value goods which were already used: $\{R'_2, R'_6, R'_7, R'_8, R'_9, R'_{10}\}$ shown in teal. The solid boxes illustrate the remainder sets R_j of which the teal sections are R'_j . All remaining goods, shown in purple, are then available for simple bag-filling to fill the remaining 2 bundles.

- The k remainder sets $(R_c^U)_{c=1}^k$ corresponding to the high-value goods $(g_c^U)_{c=1}^k$ previously allocated in the bundles $(B_c)_{c=1}^k$. We likewise define $r_c^U = v_i(R_c^U)$ for $c \in [k]$. In Figure 2, the previously allocated high-value goods are $g_1^U = g_2$ and $g_2^U = g_7$ which were allocated in bundles B_1 and B_2 . The corresponding remainder sets are $R_1^U = R_2$ and $R_2^U = R_7$.
- The $2s$ remainder sets $(R_d^P)_{d=1}^{2s}$ of the s pairs constructed in step 1. We likewise define $r_d^P = v_i(R_d^P)$ for $d \in [2s]$. In Figure 6, there are four such bundles $R_1^P = R_6$, $R_2^P = R_8$, $R_3^P = R_9$, and $R_4^P = R_{10}$.

By definition, the total value of these remainder sets is:

$$v_i \left(\bigcup_{c=1}^k R_c^U \cup \bigcup_{d=1}^{2s} R_d^P \right) = \sum_{c=1}^k r_c^U + \sum_{d=1}^{2s} r_d^P.$$

We further define $R'_j := R_j \setminus (\bigcup_{c=1}^k B_c)$; that is, R'_j is the subset of R_j that remains after removing any goods that were previously allocated in the k unacceptable bundles $(B_c)_{c=1}^k$. Since $v_i(B_c) < 1$ for each $c \in [k]$, the total value of low-value goods of each B_c is less than r_c^U (because $r_c^U + v_i(g_c^U) = v_i(R_c^U \cup \{g_c^U\}) = v_i(A_c^U) = 1 > v_i(B_c)$ and $g_c^U \in B_c$). Thus the total value of low-value goods allocated in the k unacceptable bundles is at most $\sum_{c=1}^k r_c^U$. Furthermore, the total remaining value of the remainder sets $(R_c^U)_{c=1}^k$ and $(R_d^P)_{d=1}^{2s}$ is more than $\sum_{d=1}^{2s} r_d^P$. That is,

$$v_i \left(\bigcup_{c=1}^k R_c^U \cup \bigcup_{d=1}^{2s} R_d^P \right) > \sum_{d=1}^{2s} r_d^P.$$

We then run a restricted bag-filling where we initialize each bundle with a single high-value good g_j with $j \in [\lfloor \frac{2n}{3} \rfloor - s]$ (when this step starts there are $n' - s$ such goods). We fill bags by adding the remainder sets R_c^U and R_d^P arbitrarily until the bag has a value of 1.⁷

The bag-filling stops when either there are no more high-value goods or when there are no more remainder sets to fill another bundle. In the former case, $n' - s$ bundles are filled yielding a total of n' bundles when we include the bundles formed in Step 1. We now consider the latter case.

Let t be the number of bundles completed during restricted bag-filling. For each $a \in [t]$, the a -th bag contains one high-value good, call it g_a^H for $g_a^H \in \{g_1, \dots, g_{\lfloor \frac{2n}{3} \rfloor - s}\}$, and some remainder sets R'_j . Since the bundle is worth at least 1, the total value of the remainder sets in the bag is at least $r_a^H = 1 - v_i(g_a^H)$. Let R'_a^L be the last remainder set added to bag a with value r_a^L . Then the total value of the remainder sets in bag a is at most $r_a^H + r_a^L \leq r_a^H + r_a^L$. Furthermore, since the $(t+1)$ -th bag could not be filled, the total value of the remainder sets in this last bag is less than r_{t+1}^H . Therefore the total value of remainder sets satisfies

$$\begin{aligned} v_i \left(\bigcup_{c=1}^k R_c^U \cup \bigcup_{d=1}^{2s} R_d^P \right) &< r_{t+1}^H + \sum_{a=1}^t (r_a^H + r_a^L) \\ &= \left(\sum_{a=1}^{t+1} r_a^H \right) + \left(\sum_{a=1}^t r_a^L \right) \end{aligned}$$

⁷Observe that r_c^U and r_d^P are both less than $\frac{1}{2}$, so by “glueing” these low-value goods together, we can treat the entire remainder set as a single low-value good.

Algorithm 3: A polynomial-time variant of Algorithm 2 without restricted bag-filling.

Input : A normalized instance $I = \langle N, M, V \rangle$
Output: A $(\frac{2}{3}, 1)$ -MMS allocation A on I

- 1 Let N' be a subset of $\lfloor \frac{2n}{3} \rfloor$ agents from N ;
- 2 Let $M' = M$;
- 3 **while** $|N'| > 0$ and $|M'| > 0$ **do**
- 4 Select an agent i from N' to be the divider;
- 5 Let $h = |\{g|v_i(g) > \frac{1}{2}\}|$ and $s = \max(h - \lfloor \frac{2n}{3} \rfloor, 0)$;
- 6 **if** $s = 0$ **then**
- 7 Bag-fill n' bundles;
- 8 **else**
- 9 Pair $\min(n', s)$ bundles;
- 10 **if** $s < n'$ **then**
- 11 Bag-fill $n' - s$ bundles;
- 12 Let $G = (N' \cup D, E)$ be the acceptability graph;
- 13 Find a non-empty envy-free matching \mathcal{M} of G ;
- 14 **foreach** $(j, D_k) \in \mathcal{M}$ **do**
- 15 Allocate D_k to agent j ;
- 16 $N' = N' \setminus \{j\}$;
- 17 $M' = M' \setminus D_k$;

Combining this with the lower bound of the remainder sets yields

$$\left(\sum_{a=1}^{t+1} r_a^H \right) + \left(\sum_{a=1}^t r_a^L \right) > \sum_{d=1}^{2s} r_d^P.$$

In the left-hand side there are $2t + 1$ terms, while on the right-hand side there are $2s$ terms. We now show that each term in the left-hand side is less than or equal to a unique term in the right-hand side. Since the left-hand side is larger than the right-hand side, this implies that the left-hand side must have more terms than the right-hand side. Thus $t \geq s$ and agent i successfully constructs s bundles during the restricted bag-filling.

- Consider first the terms r_a^H and compare them to the terms r_d^P for $d \in [s]$ - the remainders of the s high-value goods of H^+ used in pairing. Since the pairs in Step 1 were constructed from the lowest valued goods in $\{g_1, \dots, g_{\lfloor \frac{2n}{3} \rfloor}\}$, every remainder r_d^P is larger than any remainder r_a^H . Therefore, every term r_a^H is weakly smaller than the corresponding term r_d^P .
- Consider now the terms r_a^L and compare them to r_d^P for $d \in \{s+1, \dots, 2s\}$ - the remainders of the s high-value goods of H^- used in pairing. Each of the r_a^L is unique, so it is either equal to some remainder r_d^P or to some remainder r_c^U . All remainders r_c^U correspond to high-value goods g_j for $j \in [\lfloor \frac{2n}{3} \rfloor]$ so they are smaller than the remainders r_d^P for $d \in \{s+1, \dots, 2s\}$. Therefore, every r_a^L is weakly smaller than some unique term r_d^P .

Step 3: Plain Bag-filling. Up to this point, agent i has used a total value of at most $3s + k/2$:

- The $2s$ high-value goods used in Step 1 plus their remainders r_d^P .
- An additional s high-value goods used in Step 2.
- The remainders r_c^U for $c \in [k]$ used in Step 2, each of which has a value of less than $\frac{1}{2}$.

Taking account of each of the k unacceptable bundles which have value less than 1, the total remaining value is more than

$$n - (3s + k/2) - k \geq \frac{3}{2} \lfloor \frac{2n}{3} \rfloor - 3s - \frac{3}{2}k$$

$$\frac{3}{2} (\lfloor \frac{2n}{3} \rfloor - 2s - k) = \frac{3}{2} (n' - 2s).$$

Furthermore, there are exactly $n' - 2s$ high-value goods remaining, and the value of all other remaining goods is at most $\frac{1}{2}$. Therefore, by plain bag-filling as in Case 1, agent i can fill the required $n' - 2s$ bags. \square

C.3 Proof of Theorem 4

THEOREM 4. For $n < 9$, $(\frac{2}{3}, 1)$ -MMS can be computed in polynomial time.

PROOF. Our algorithm follows the intuition of the proof of Theorem 3. The main difference is that we relax strong normalization (Lemma 5) to normalization (Lemma 4), and we skip the restricted bag-filling step. We show that the resulting algorithm, seen in Algorithm 3, computes an $(\frac{2}{3}, 1)$ -MMS allocation when $n < 9$.

Using Lemma 4, we assume that the instance is normalized; that is, the instance is ordered, scaled so that $v_i(M) = n$, and reduced $v_i(g_1) < 1$ and $v_i(\{g_n, g_{n+1}\}) < 1$. As in the proof of Theorem 3, we use the lone divider technique. We begin with N' as an arbitrary subset of $\lfloor \frac{2n}{3} \rfloor$ agents. We select one agent to divide items into $n' = |N'|$ bundles. We create a bipartite graph with agents of N' on one side and the bundles $(D_1, \dots, D_{n'})$ on the other. Since there are n' bundles which are each acceptable to the divider, an envy-free matching exists by Lemma 13. We find such a matching and allocate the matched bundles to the corresponding agents. Then we remove the agents and bundles from the market and repeat this process until no agents remain. As in Theorem 3, we require that each allocated bundle contains exactly one good from $\{g_1, \dots, g_{\lfloor \frac{2n}{3} \rfloor}\}$.

As in Theorem 3, the key challenge is to show that at each iteration, the divider is able to form n' bundles. Consider an arbitrary iteration with an arbitrary divider i . Suppose that k bundles $(B_c)_{c=1}^k$ have previously been allocated. Since agent i was not envious of the agents who received these bundles, $v_i(B_c) < 1$. We define a good g to be high-value if $v_i(g) \geq \frac{1}{2}$. Since $v_i(\{g_n, g_{n+1}\}) < 1$, $v_i(g_{n+1}) < \frac{1}{2}$ implying that there are at most n high-value goods.

We consider two cases dependent upon the number of high-value goods.

Case 1 $h \leq \lfloor \frac{2n}{3} \rfloor$. The total value remaining is at least $n - k$. In this case, agent i forms bundles by bag-filling with each bag initialized with one of the remaining n' goods from $\{g_1, \dots, g_{\lfloor \frac{2n}{3} \rfloor}\}$. Each bag is filled using low-value goods in order of decreasing value until the value of the bundle falls in the range $[1, \frac{3}{2})$. This process is continued until the remaining goods have value less than 1. Suppose that t bundles are filled this way. Then the total value of all goods is less than $\frac{3}{2}t + 1 > n - k$. Thus $t \geq \frac{2}{3}n - \frac{2}{3}k - \frac{2}{3}$. Since t, n, k are all integers, $t \geq \lfloor \frac{2n}{3} \rfloor - k = n'$.

Case 2 $h > \lfloor \frac{2n}{3} \rfloor$. In this case, define $s = h - \lfloor \frac{2n}{3} \rfloor$ to be the number of surplus high-value goods. Agent i begins by pairing $\min(n', s)$ bundles each containing one high-value good from $\{g_1, \dots, g_{\lfloor \frac{2n}{3} \rfloor}\}$ and one high-value good from $\{g_{\lfloor \frac{2n}{3} \rfloor + 1}, \dots, g_h\}$. If $n' \leq s$, then there are sufficiently many bundles for the envy-free matching and we are done. If $s < n'$, then we let agent i form the remaining bundles using the simple bag-filling of Case 1.

We now show that when $n < 9$, this bag-filling will form at least $n' - s$ bundles. Suppose that t bundles were formed during this bag-filling. Consider the following cases which depend on h, s , and t .

- (1) Suppose $t \geq n' - s$. Then since $\lfloor \frac{2n}{3} \rfloor - s \geq n' - s$, we have sufficiently many bundles continue to the envy-free matching.
- (2) Suppose $t \leq n' - 2s$. Since the instance is ordered, the $(h - t)$ high-value goods not allocated during bag-filling all have value at most $v_i(g_{t+1})$. Thus the total remaining low-value goods have value at least $n - k - \frac{3}{2}t - (n' + s - t)v_i(g_{t+1})$.

$$\begin{aligned}
n - k - \frac{3}{2}t - (n' + s - t)v_i(g_{t+1}) &\geq \frac{3}{2} \lfloor \frac{2n}{3} \rfloor - k - \frac{3}{2}t - (n' + s - t)v_i(g_{t+1}) \\
&= \frac{1}{2} \lfloor \frac{2n}{3} \rfloor + (\lfloor \frac{2n}{3} \rfloor - k) - \frac{3}{2}t - (n' + s - t)v_i(g_{t+1}) \\
&= \frac{1}{2} \lfloor \frac{2n}{3} \rfloor + n' + s - s - \frac{3}{2}t - (n' + s - t)v_i(g_{t+1}) \\
&= \frac{1}{2} \lfloor \frac{2n}{3} \rfloor - \frac{1}{2}t - s + (n' + s - t) - (n' + s - t)v_i(g_{t+1}) \\
&= \frac{1}{2} (\lfloor \frac{2n}{3} \rfloor - 2s - t) + (n' + s - t)(1 - v_i(g_{t+1}))
\end{aligned}$$

Since $t \leq n' - 2s$, $\lfloor \frac{2n}{3} \rfloor - 2s - t \geq 0$ and $n' + s - t > 0$ implying that the total remaining value from low-value goods is at least $1 - v_i(g_{t+1})$ and another bundle could have been filled. This contradicts that bag filling stopped with $t \leq n' - 2s$.

- (3) The remaining cases arise when $n' - 2s < t < n' - s$. We enumerate each case to show that they do not occur when $n < 9$. We consider cases as a tuple of the form (n, n', s, t) for clarity. Since $n < 9$, and $n' - 2s < t < n' - s$, there are 12 cases, seen in Table 2

The cases $(8, 5, 3, 0)$, $(8, 4, 3, 0)$, $(8, 3, 2, 0)$, $(7, 4, 3, 0)$, $(7, 3, 2, 0)$, $(6, 3, 2, 0)$, and $(5, 3, 2, 0)$ assume that no bundles are allocated during bag-filling. However, combining the facts that there is $n - k$ value remaining, $v_i(g) \leq v_i(g_1) < 1$, there are at most $n - k$ high-value goods remaining, and $v_i(M) = n$, there must be at least $(n - k) - (n - k)v_i(g_1) \geq (n - k)(1 - v_i(g_1)) \geq 1 - v_i(g_1)$ value from low-value goods. Thus at least one bundle is allocated during bag-filling and these cases do not occur.

The cases $(6, 4, 2, 1)$ and $(8, 5, 3, 1)$ only occur if agent i believes there are n high-value goods. Notice also that $n' = \lfloor \frac{2n}{3} \rfloor$ in both cases so no bundles were previously allocated. Since $v_i(\{g_n, g_{n+1}\}) < 1$ all low-value goods are worth less than $1 - v_i(g_n)$. If $v_i(\{g_1, g_{n+1}\}) < 1$

then the first bundle from bag-filling was $\{g_1, g_{n+1}\}$. In this case, the amount of value remaining from low-value goods is at least:

$$\begin{aligned}
& n - (v_i(g_1) + v_i(g_{n+1})) - \sum_{k=2}^n v_i(g_k) \\
&= n - (v_i(g_n) + v_i(g_{n+1})) - \sum_{k=1}^{n-1} v_i(g_k) \\
&> n - 1 - \sum_{k=1}^{n-1} v_i(g_k) \\
&\geq (1 - v_i(g_2)) + (n - 2) - \sum_{k=1, k \neq 2}^{n-1} v_i(g_k) \\
&> (1 - v_i(g_2)) + (n - 2) - (n - 2) \\
&> 1 - v_i(g_2)
\end{aligned}$$

If $\{g_1, g_{n+1}\}$ is not worth 1 agent i , then because low-value goods are added in descending order, the maximum value from low-value goods allocated in the first bundle is $2(1 - v_i(g_1))$. Thus the value remaining from low-value goods is at least:

$$\begin{aligned}
& n - \sum_{k=1}^n v_i(g_k) - 2(1 - v_i(g_1)) \\
&\geq (n - 2) - \sum_{k=2}^{n-1} v_i(g_k) + (1 - v_i(g_1)) + (1 - v_i(g_n)) \\
&\quad - 2(1 - v_i(g_1)) \\
&\geq (n - 2) - \sum_{k=2}^{n-1} v_i(g_k) \\
&= (1 - v_i(g_2)) + (n - 3) - \sum_{k=3}^{n-1} v_i(g_k) \\
&\geq (1 - v_i(g_2)) + (n - 3) - (n - 3) \\
&= 1 - v_i(g_2)
\end{aligned}$$

In either case, there is at least enough value from low-value goods to fill another bundle during bag-filling contradicting that bag-filling stopped. Thus the cases (6, 4, 2, 1) and (8, 5, 3, 1) do not occur.

The next cases we address are (7, 4, 2, 1) and (8, 5, 2, 2). In these cases, $n' = \lfloor \frac{2n}{3} \rfloor$, and there are $n - 1$ high-value goods. We observe that the amount of value from low-value goods in these cases is at least:

$$\begin{aligned}
& n - \sum_{k=1}^{n-1} v_i(g_k) \\
&\geq 1 + (n - 1) - v_i(g_1) - v_i(g_2) - v_i(g_3) - \sum_{k=4}^{n-1} v_i(g_k) \\
&= 1 + (1 - v_i(g_1)) + (1 - v_i(g_2)) + (1 - v_i(g_3)) \\
&\quad + (n - 4) - \sum_{k=4}^{n-1} v_i(g_k) \\
&> 1 + (1 - v_i(g_1)) + (1 - v_i(g_2)) + (1 - v_i(g_3)) \\
&\quad + (n - 4) - (n - 4) \\
&> \left(\frac{3}{2} - v_i(g_1)\right) + \left(\frac{3}{2} - v_i(g_2)\right) + (1 - v_i(g_3))
\end{aligned}$$

Since the first two bundles have total value at most $\frac{3}{2}$, there is at least enough value to fill the first two bundles and still have enough low-value goods to fill a third bundle. This contradicts that only one or two bundles were filled during bag-filling (based on the cases). Thus neither cases (7, 4, 2, 1) and (8, 5, 2, 2) occur.

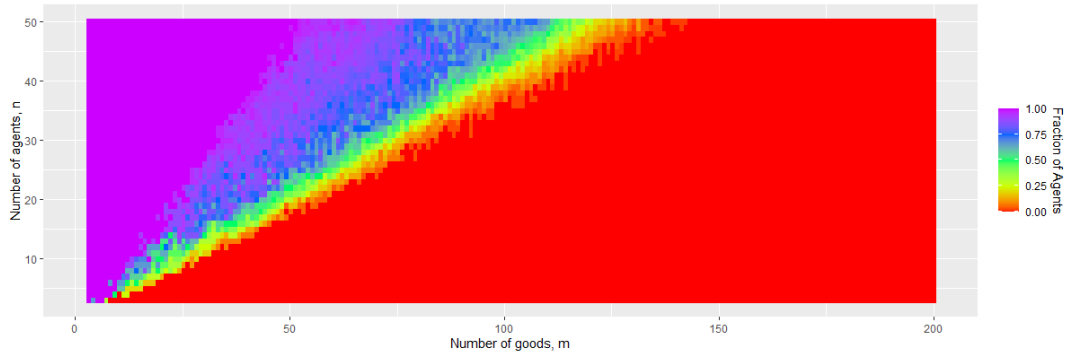


Figure 7: The fraction of agents removed during the reduction phase of normalization in our experiments.

The last case is (8, 4, 2, 1). In this case, one bundle has previously been allocated which leaves at least $n - k = 7$ value. Accounting for the 4 high-value goods used in pairing and the $\frac{3}{2}$ value allocated in the first bundle from bag-filling, there is at least $7 - 4 - \frac{3}{2} = \frac{3}{2}$ value remaining from only low-value goods. Thus a second bundle is filled during bag-filling and the case (8, 4, 2, 1) does not occur.

n	n'	s	t
8	5	2	2
8	5	3	1
8	5	3	0
8	4	2	1
8	4	3	0
8	3	2	0
7	4	2	1
7	4	3	0
7	3	2	0
6	4	2	1
6	3	2	0
5	3	2	0

Table 2: The remaining cases where $n' - 2s < t < n' - s$.

□

D ADDITIONAL MATERIAL FOR SECTION 6

Figure 7 shows the number of agents that were removed during the reduction phase. It is important to observe that with uniformly random valuations, the more goods there are, the closer each good's value is to $\frac{n}{m}$ after valuations are scaled so that $v_i(M) = n$.

This observation likely explains the strong banding apparent in Figure 7. When $m < 2n$, the pigeonhole principle guarantees that there is a bundle of any MMS partition which contains exactly one good. Using a simple swapping argument, we see that $v_i(g_1) \geq \text{MMS}_i^n$ when $m < 2n$. While we do not explicitly implement this reduction in our experiment, the band close to $m = 2n$ illustrates that random valuations will still exhibit this behavior.