Maximin Shares Under Cardinality Constraints*

Halvard Hummel[1 [0000−0001−5691−8177]] and Magnus Lie Hetland[1 [0000−0003−4204−2017]]

Norwegian University of Science and Technology, Trondheim, Norway,
halvard.hummel@ntnu.no, mlh@ntnu.no

Abstract. We study the problem of fair allocation of a set of indivisible items among agents with additive valuations, under cardinality constraints. In this setting, the items are partitioned into categories, each with its own limit on the number of items it may contribute to any bundle. We consider the fairness measure known as the *maximin share* (MMS) *guarantee*, and propose a novel polynomial-time algorithm for finding 1/2-approximate MMS allocations for goods—an improvement from the previously best available guarantee of 11/30. For single-category instances, we show that a modified variant of our algorithm is guaranteed to produce 2/3-approximate MMS allocations. Among various other existence and non-existence results, we show that a $(\sqrt{n}/(2\sqrt{n}-1))$-approximate MMS allocation always exists for goods. For chores, we show similar results as for goods, with a 2-approximate algorithm in the general case and a 3/2-approximate algorithm for single-category instances. We extend the notions and algorithms related to *ordered* and *reduced instances* to work with cardinality constraints, and combine these with *bag filling* style procedures to construct our algorithms.

Keywords: Constrained Fair Allocation, Indivisible Goods, Indivisible Chores, Maximin Share, Matroid Constraints, Cardinality Constraints

1 Introduction

The problem of fair allocation is one that naturally occurs in many real-world settings, for instance when an inheritance is to be divided or limited resources are to be distributed. For a long time, the research in this area primarily focused on the allocation of divisible items, but lately the interest in the more computationally challenging case of indivisible items has seen a surge. (Bouveret et al. provide a somewhat recent overview [8]). For this variant of the problem, many of the central fairness measures in the literature on divisible items, such as *envy-freeness* and *proportionality*, are less useful. Instead, relaxed fairness measures, such as the *maximin share* (MMS) *guarantee* [10], have been introduced, where all agents receive at least as much as if they partitioned the items but were the last to select a bundle. It is not always possible to find an MMS allocation [12,22,26], but good approximations exist [15,16].

* A preliminary version of this paper appeared at AAMAS 2022 as an extended abstract [21].

Fairly allocating items in the real world often involves placing constraints on the bundles allowed in an allocation. For example, consider the problem where a popular physical conference or convention offers a variety of talks and panels organized across several synchronized parallel tracks. Due to space constraints, each talk is limited to some maximum number of participants, fewer than the total number of participants at the conference. Consequently, there may be more people interested in attending some talks than there are available seats. To mitigate this, the conference wants to fairly allocate the available seats, based on participants' preferences, so that no participant receives seats they cannot use, i.e., multiple seats at the same talk or seats at multiple talks in the same time slot. In order to solve this problem, we need to be able to express that some items belong to the same category (seats at talks in the same time slot) and that there is a limit on the number of items each category can contribute to any bundle (in this case 1). This kind of constraints is called *cardinality constraints* and was introduced by Biswas and Barman [6].

The conference example highlights a general type of problems for which cardinality constraints are useful, where each agent should not receive more items of a certain type than she could possibly have use for. Another such problem is the motivating example of Biswas and Barman [6]: A museum is to fairly allocate exhibits of different types to newly opened branches. To make sure that each branch can handle its allocated exhibits, so that no exhibits go to waste, an upper limit is placed on the number of exhibits each branch can be allocated of each exhibit type. The constraints may also provide each agent with some diversity in the type of items she receives. For example, with sufficiently small limits in the museum example, each branch must receive a somewhat diverse collection of exhibits.

Another application is making sure that items of certain types are guaranteed to be roughly evenly distributed among the agents. This can be achieved by setting the number of items each agent can receive from a given category close to the number of items in this category divided by the number of agents. For example, consider a situation where a set of donated items, including a limited number of internet-capable devices, are to be fairly allocated to low-income families. A single family can make use of many internet-capable devices. However, the organization behind the allocation process may want to make sure that as many families as possible have access to the internet. By placing all the internet-capable devices in the same category and giving each family at most one item from this category, the internet-capable devices will be distributed to as many families as possible.

Biswas and Barman [6] showed that under cardinality constraints, with additive valuations, it is always possible to find an allocation of goods where each agent gets at least 1/3 of her MMS. This is achieved by a reduction to an unconstrained setting with submodular valuations, where the approximate allocation is found using an algorithm described by Ghodsi et al. [16]. More recently, Li and Vetta showed that 11/30-approximate MMS allocations are guaranteed to exist under hereditary set system constraints [24]. This approximation guarantee is achievable in polynomial time for certain classes of set systems, including set systems representing cardinality constraints.

1.1 Contributions

We develop a polynomial-time algorithm for finding 1/2-approximate MMS allocations for goods under cardinality constraints, improving on the 1/3 and 11/30 guarantees of Biswas and Barman [6] and Li and Vetta [24], which are, to our knowledge, the best guarantees previously available. To construct the algorithm, we extend the notions and algorithms related to *ordered* and *reduced instances* to work with cardinality constraints, and combine these with a *bag-filling* style algorithm. Combining this algorithm with a lone-divider style [1] preprocessing step, we show that $(\sqrt{n}/(2\sqrt{n}-1))$-approximate MMS allocations always exist for goods—a large improvement for few agents. The preprocessing step unfortunately relies on finding MMS-partitions, an NP-hard problem [29]. However, the 1/2-approximate MMS algorithm is able to find both $(n/(2n-1))$-approximate MMS allocations and 1-out-of-$(2n-1)$ MMS allocations by changing a constant.

For chores, we show that a similar approach finds 2-approximate (or, more precisely, $((2n-1)/n)$-approximate) MMS allocations in polynomial time. This is, to our knowledge, the first MMS result for chore allocation under cardinality constraints.

We also examine a special case of cardinality constraints, in which all the items belong to the same category. This case is equivalent to placing a restriction on the number of items in each bundle, or equivalently restricting bundles to independent sets of a *uniform matroid*. This is a setting of interest in itself, especially for chores, where it can be useful to make sure that no agent is stuck with a much larger number of chores than anyone else. By modifying our general algorithms, we show that in this special case, (2/3)-approximate MMS allocations for goods and (3/2)-approximate MMS allocations for chores can be found in polynomial time.

1.2 Related Work

Several other constraint types have been examined in the recent literature. (See Suksompong's recent survey for a detailed overview [28].) One such constraint is that all agents must receive exactly the same number of items [13], a more restrictive version of our single-category instances. Another, studied by Bouveret et al., uses an underlying graph to represent connectivity between the items and requires each bundle to form a connected component [7]. Such connectivity constraints have since been explored in many papers [e.g., 5,19,25]. A variation is the allocation of *conflicting items*, where each bundle must be an independent set in the graph [11,20]. There is some overlap between this scenario and cardinality constraints with threshold 1 [cf. 20], but neither is a generalization of the other. Cardinality constraints have recently been studied by Shoshan et al., who considered the problem of finding allocations that are both Pareto optimal and EF1 for instances with two agents [27].

Matroids have been used to constrain allocations in several different ways [18]. The cardinality constraints placed on a single bundle may in fact be represented by a *partition matroid* or for single-category instances a *uniform matroid*. The 1/2-approximate MMS algorithm of Gourvès and Monnot [17] applies to the superficially similar problem where a single matroid constraint is placed on the union of *all* bundles. As pointed out by Biswas and Barman [6], this algorithm cannot be applied to the cardinality constraint scenario.

2 Preliminaries

For a given instance, $I = \langle N, M, V \rangle$, of the fair allocation problem, let $N = \{1, 2, \dots, n\}$ denote a set of *agents*, $M = \{1, 2, \dots, m\}$, a set of *items*, and $V = \langle v_1, v_2, \dots, v_n \rangle$, the *valuation profile*, i.e., the collection of the agents' *valuation functions* $v_i : 2^m \to \mathbb{R}$ over the subsets $S \subseteq M$. For simplicity, the valuation of a single item $v_i(\{j\})$ will be denoted by both $v_i(j)$ and v_{ij}. We assume that the valuations are additive, i.e., $v_i(S) = \sum_{j \in S} v_{ij}$. We wish to find an allocation $A = \langle A_1, A_2, \dots, A_n \rangle$ that forms a partition of M into n possibly empty subsets, or *bundles*, one for each agent. We say that an instance I consists of *goods* if $v_{ij} \geq 0$ for all $i \in N, j \in M$, and *chores* if $v_{ij} \leq 0$ for all $i \in N, j \in M$. We consider both instances consisting of goods and ones consisting of chores. However, we do not consider instances consisting of a mix of goods and chores. For simplicity, we will throughout the paper assume that all instances consist of goods, except for in Section 7, which covers our results on chores.

For the fair allocation problem *under cardinality constraints*, an instance is given by $I = \langle N, M, V, C \rangle$, where C is a set of ℓ pairs $\langle C_h, k_h \rangle$ of *categories* C_h and corresponding *thresholds* k_h. The categories constitute a partition of the items, M. An allocation A is *feasible* for the instance if no agent receives more than k_h items from any category C_h, i.e., if $|A_i \cap C_h| \leq k_h$ for all $i \in N, h \in \{1, \dots, \ell\}$. We let $\mathcal{F}_I$ denote the set of all feasible allocations for I, with the subscript omitted if it is clear from context. To guarantee that there is at least one feasible allocation, i.e., $\mathcal{F} \neq \emptyset$, no category may contain more items than we can possibly distribute, i.e., we require that $|C_h| \leq nk_h$ for all $h \in \{1, \dots, \ell\}$[1].

We are concerned with the fairness criterion known as the *maximin share guarantee* [10]. The *maximin share* (MMS) of an agent is the value of the most preferred bundle the agent can guarantee herself if she were to divide the items into feasible bundles and then choose her own bundle last. More formally:[2]

Definition 1. *Let $I = \langle N, M, V, C \rangle$ be an instance of the fair allocation problem under cardinality constraint. The* maximin share *of an agent i for the instance I is given by*

$$\mu_i^I = \max_{A \in \mathcal{F}_I} \min_{A_j \in A} v_i(A_j) \,,$$

[1] Instances with more than nk_h items in category C_h can be handled by ordering the instance (see Section 3) and ignoring the worst items in the category.

[2] The definition is equivalent for chores.

where $\mathcal{F}_I$ is the set of feasible allocations for I. If I is obvious from context, we write simply μ_i.

An allocation is said to *satisfy the MMS guarantee*, or *to be an MMS allocation*, if each agent gets a bundle valued at least as much as the agent's MMS, i.e., $v_i(A_i) \geq \mu_i$ for all agents i. We concern ourselves with allocations that satisfy this guarantee *approximately*, where an allocation is said to be an *α-approximate MMS allocation* for some $\alpha > 0$ if $v_i(A_i) \geq \alpha\mu_i$ for all agents i. An allocation A is said to be an *MMS partition* of an agent i, if $v_i(A_j) \geq \mu_i$ for all $A_j \in A$. By definition, at least one MMS partition exists for any agent in any instance. As MMS allocations are not guaranteed to exist [12,22,26], there exists a generalized and relaxed version of MMS, called the *l-out-of-d MMS* [3].[3] This fairness criterion works like MMS, except that the agent is to partition the goods into d feasible bundles maximizing the combined value of the l least valuable bundles in the partition. Our algorithms require some knowledge about the value of μ_i in order to determine when a bundle is worth at least $\alpha\mu_i$ to an agent i. Finding the MMS of an agent is known to be NP-hard for the unconstrained fair allocation problem [29]. Since unconstrained fair allocation is simply the special case of $\ell = 1$ and $k_1 = m$, finding an agent's MMS is at least as hard under cardinality constraints.[4] In order to provide polynomial-time algorithms, we exploit the fact that μ_i cannot be larger than the average bundle value, i.e., $\mu_i \leq v_i(M)/n$, and we can scale all values so that $v_i(M) = n$, so that $\mu_i \leq 1$, as shown in the following theorems. Due to space constraints, their proofs have been omitted, but can be found in the appendix along with all other omitted proofs. The proofs from ordinary fair allocation for the two succeeding theorems do in fact extend to cardinality constraints without any modification [see, e.g., 2,14]. We assume, without loss of generality, that $v_i(M) > 0$ for each agent i.[5]

Theorem 1 (Scale invariance). *If A is an MMS allocation for the instance $I = \langle N, M, V, C\rangle$, then A is also an MMS allocation for $I' = \langle N, M, V', C\rangle$, where $v_i'(S) = a_i v_i(S), a_i > 0$, for some agent i.*

Theorem 2 (Normalization). *Let $I = \langle N, M, V, C\rangle$ be an instance of the fair allocation problem of under cardinality constraints and $v_i(M) = |N|$ for some agent i. Then $\mu_i \leq 1$.*

Once valuations have been normalized, constructing an α-approximate MMS allocation reduces to providing each agent with a bundle worth at least α.

[3] We use l instead of the usual ℓ to avoid conflicting use of symbols.

[4] In the unconstrained setting, a PTAS exists for finding the MMS of each individual agent [29], but this PTAS does not extend to fair allocation under cardinality constraints and there does not, to our knowledge, exist a PTAS for this problem.

[5] If $v_i(M) = 0$, normalization does not work. However, since this implies $\mu_i = 0$, Corollary 1 can be used to eliminate agent i from the instance.

3 Ordered Instances

In the unconstrained setting, Bouveret and Lemaître showed that each instance can be reduced to an instance where all agents have the same preference order over all goods [9]. That is, in such an instance there exists an ordering of the goods such that when $j < k$, we have $v_{ij} \geq v_{ik}$ for all agents i. While Bouveret and Lemaître introduced these as instances that satisfy *same-order preferences*, we will refer to them as *ordered instances*, as is the norm for MMS-approximation algorithms [4,14].

The reduction works as follows. For each agent, sort the good values and reassign these to the goods, which are listed in some predetermined order, common to all agents. Allocations for the reduced instance are converted into allocations for the original instance, without diminishing their value, by going through the goods in the predetermined order; the agent who originally received a given good instead chooses her highest-valued remaining good.

Since only the permutation of value assignments to goods changes, the reduction does not change the MMS of each agent. Thus, any α-approximate MMS allocation in the ordered instance will also be α-approximate in the original instance. Ordered instances are therefore at least as hard as any other instances, and it suffices to show that an algorithm produces an α-approximate MMS allocation for ordered instances.

The standard definition of an ordered instance does not work under cardinality constraints, due to an inherent loss of information about which goods belong to which category. Without this information, one is not guaranteed to be able to produce a feasible α-approximate MMS allocation when converting back to the original instance. We generalize the definition to fair allocation under cardinality constraints. In the special case where $\ell = 1$, this definition and the later conversion algorithms are equivalent to those of Bouveret and Lemaître.

Definition 2. *An instance $I = \langle N, M, V, C \rangle$ of the fair allocation problem under cardinality constraints is called an* ordered instance *if each category $C_h = \{c_1, c_2, \cdots, c_{|C_h|}\}$ is ordered such that for all agents i, $v_i(c_1) \geq v_i(c_2) \geq \cdots \geq v_i(c_{|C_h|})$.*

With the generalized definition, the reduction of MMS-approximation to ordered instances can be extended to cardinality constraints by applying the algorithms of Bouveret and Lemaître to each category C_h individually, as shown in algorithms 1 and 2.

Lemma 1. *Let $I = \langle N, M, V, C \rangle$ be an instance of the fair allocation problem under cardinality constraints, and A' a feasible α-approximate MMS allocation for the ordered instance I' produced by Algorithm 1. Then the allocation produced by conversion of A' with Algorithm 2 is a feasible α-approximate MMS allocation for I.*

Repeating the ordering and deordering procedure for each category does not affect the polynomial nature of the procedures. As a result, the reduction to ordered instances holds.

Algorithm 1 Order instance

Input: Instance $I = \langle N, M, V, C \rangle$
Output: Ordered $I' = \langle N, M, V', C \rangle$

for $(C_h, k_h) \in C$
  for $j = 1$ **to** $|C_h|$
    for $i \in N$
      $v'_i(c_j) = i$'s jth highest value in C_h
return $\langle N, M, V', C \rangle$

Algorithm 2 Recover solution

Input: Instance $I = \langle N, M, V, C \rangle$ and allocation A' for corresponding I'
Output: Allocation A for I

$A = \langle \emptyset, \ldots, \emptyset \rangle$
for $(C_h, k_h) \in C$
  for $j = 1$ **to** $|C_h|$
    $i =$ agent for which $c_j \in A'_i$
    $j^* = i$'s preferred item in C_h
    $A_i = A_i \cup \{j^*\}$
    $C_h = C_h \setminus \{j^*\}$
return A

Theorem 3. *For fair allocation under cardinality constraints, MMS-approximation reduces to MMS-approximation of ordered instances in polynomial time.*

Proof. By Lemma 1 it is sufficient to find an α-approximate MMS allocation for the reduced instance produced by Algorithm 1. Since both algorithms 1 and 2 are polynomial in the number of agents and goods for each category, the reduction is polynomial in the number of agents, goods and categories.

4 Reduced Instances

High-valued goods are generally harder to handle than low-valued goods in MMS-approximation. Low-valued goods can easily be distributed across bundles in an approximately even manner and to a certain extent in a way that makes up for an uneven value distribution due to the high-valued goods. High-valued goods, on the other hand, allow only for a rough and usually uneven distribution. In order to simplify the problem instances, we wish to minimize both the number of high-valued goods and the maximum value of a good.

If we remove an agent i and a bundle $B \subseteq M$ from an instance, the result is called a *reduced* instance. If the bundle's value is sufficiently high ($v_i(B) \geq \alpha\mu_i$) and the MMS of the remaining agents are at least as high after the removal, this is called a *valid reduction* [15], a concept used in many MMS approximation algorithms for the unconstrained fair allocation problem [e.g., 14,23,16].[6] With a valid reduction we can both guarantee agent i a bundle with a value of at least $\alpha\mu_i$ and reduce the original instance to a smaller problem instance.

Given the above definition, a valid reduction could leave an instance without any feasible (complete) allocations, as there may be more goods left in a category than can be allocated to the remaining agents. We require that a valid reduction leaves the reduced instance with at least one feasible allocation.

[6] The term *reduction* here refers to data reduction, as the term is used in parameterized algorithm design, rather than to the problem transformations of complexity theory.

Definition 3. *Let $I = \langle N, M, V, C \rangle$ be an instance of the fair allocation problem under cardinality constraints, B a feasible bundle, i an agent, and $I' = \langle N \setminus \{i\}, M \setminus B, V', C' \rangle$, where V' and C' are equivalent to V and C, with agent i and the items in B removed. If $v_i(B) \geq \alpha\mu_i^I$, $\mathcal{F}_{I'} \neq \emptyset$ and $\mu_{i'}^{I'} \geq \mu_{i'}^I$ for all $i' \in N \setminus \{i\}$, then allocating B to i is called a valid reduction.*

Most of the valid reductions used in unconstrained fair allocation are based on the pigeonhole principle. If you can find a set of goods that are worth at least $\alpha\mu_i$ to some agent i and show that all agents must have an MMS partition with a bundle containing an equivalent number of equally or higher valued goods, then you have a valid reduction. The latter part is exactly what the pigeonhole principle promises if we, e.g., look at the bundle $\{n, n+1\}$ in unconstrained fair allocation. Under cardinality constraints, we can also utilize the pigeonhole principle to find valid reductions. The usefulness is, somewhat reduced, due to both a lack of a common preference ordering across categories and the restrictiveness of the category thresholds. We can, however, show a general result for valid reductions based on the pigeonhole principle.

Theorem 4. *Let $I = \langle N, M, V, C \rangle$ be an ordered instance of the fair allocation problem under cardinality constraints, and let $B = \{j_1, \ldots, j_k\}$ be a feasible bundle of $k \geq 1$ goods such that $v_i(B) \geq \alpha\mu_i$ for an agent $i \in N$ and $\alpha > 0$. Let each agent $i' \in N \setminus \{i\}$ have a bundle $B_{i'}$ in one of her MMS partitions such that there is an injective map $f : B \to B_{i'}$ where, for each $j \in B$, j and $f(j)$ belong to the same category, and $v_{i'}(f(j)) \geq v_{i'}(j)$. Let B' be the bundle consisting of the goods in B and for each $C_h \in C$ the $\max(0, |C_h \setminus B| - (|N| - 1)k_h)$ lowest-valued goods in $C_h \setminus B$. Then, B' and i form a valid reduction for I and α.*

Proof sketch (full proof in appendix). For any agent $i' \neq i$, the injective map and the construction of B' guarantees that there is a way to modify the MMS partition of i' through trades and transfers of goods, such that one bundle is turned into B' and the value of any other bundle is at least as high as in the MMS partition originally. The construction of B' also guarantees a valid instance after the reduction. Since $v_i(B') \geq v_i(B) \geq \alpha\mu_i$, B' and i form a valid reduction for I and α. □

We can easily use the general result of Theorem 4 to construct similar valid reductions to those in the unconstrained setting. Any good i valued at more than $\alpha\mu_i$ for some agent i can be used for a reduction, as the identity function $f : \{j\} \to \{j\}$ satisfies the criteria of Theorem 4. Similarly, by the pigeonhole principle, we can create valid reductions with the n-th and $(n+1)$-th most valuable goods in a single category.

Corollary 1. *Let $I = \langle N, M, V, C \rangle$ be an ordered instance of the fair allocation problem under cardinality constraints, where there is an agent $i \in N$ and a good $j \in M$ such that $v_{ij} \geq \alpha\mu_i$ for $\alpha > 0$. Then, a valid reduction can be constructed from the bundle $B = \{j\}$.*

Corollary 2. *Let $I = \langle N, M, V, C\rangle$ be an ordered instance of the fair allocation problem under cardinality constraints, with a category $C_h = \langle c_1, c_2, \ldots, c_{|C_h|}\rangle$, $|C_h| \geq |N| + 1$, where $v_i(\{c_{|N|}, c_{|N|+1}) \geq \alpha\mu_i$ for some $i \in N$ and $\alpha > 0$. Then, a valid reduction can be constructed from the bundle $B = \{c_{|N|}, c_{|N|+1}\}$.*

It can be tempting to think that we can employ the same valid reductions within a single category as is possible in the unconstrained setting. This is not the case, even when the instance only has a single category and three agents with identical valuations. For example, in the unconstrained setting, any bundle B consisting of two goods, with $v_i(B) \geq \alpha\mu_i$ for an agent $i \in N$ and $v_{i'}(B) \leq \mu_{i'}$ for all other agents $i' \in N \setminus \{i\}$, can be used for a valid reduction. This, is not the case under cardinality constraints, even when removing B and i produces a feasible instance without removing any other goods.[7]

5 MMS Results under Cardinality Constraints

The reductions of theorems 2 and 3 and Corollary 1, which can be performed in polynomial-time, let us restrict finding α-approximate MMS allocations to normalized ordered instances where each good is worth less than α, without loss of generality. For such instances, Algorithm 3 can be used to find $(|N|/(2|N| - 1))$-approximate MMS allocations, which for any number of agents is at least a 1/2-approximate MMS allocation.

The algorithm works in a somewhat similar manner to bag filling algorithms for unconstrained fair allocation [see, e.g., 14,16], i.e., by incrementally adding goods to (and, in our case, removing goods from) a "bag," or partial bundle, B, until $v_i(B) \geq \alpha$ for some agent i. The major difference is the initial content of the bundle. To make sure that a complete feasible allocation is found, the bundle initially contains the $\lfloor |C_h|/n\rfloor$ least-valuable remaining goods in each category C_h (denoted by C_h^L). This guarantees that the required number of goods is given away from each category. The value of the bundle is then incrementally increased, so as to not increase the value by more than α in each step, by exchanging one of the goods in B from some C_h^L, for one of the $\lfloor |C_h|/n\rfloor$ most valuable remaining goods in the same category (denoted C_h^H). To mitigate possible effects of rounding $|C_h|/n$, one additional good may be added from any category where $|C_h|/n > \lfloor |C_h|/n\rfloor$.

Before proving that the algorithm does indeed find a 1/2-approximate MMS allocation, we first need a lower bound on the value of the remaining goods at any point during the execution of the algorithm.

Lemma 2. *Let $I = \langle N, M, V, C\rangle$ be a normalized ordered instance of the fair allocation problem under cardinality constraints where all goods are worth less than α for some $\alpha \geq 1/2$. Let n denote the number of remaining agents at any given point during the execution of Algorithm 3. Then each remaining agent assigns a value of at least $|N| - 2(|N| - n)\alpha$ to the set of unallocated goods.*

[7] See Example 1 in the appendix for a simple instance where this fails.

Algorithm 3 Find a α-MMS solution to ordered instance

Input: A normalized ordered instance $I = \langle N, M, V, C \rangle$ with all $v_{ij} < \alpha$
Output: Allocation A consisting of each bundle B allocated

1 **while** there is more than one agent left
2     $B = \cup_{h=1}^{\ell} C_h^L$
3   **while** $v_i(B) < \alpha$ for all agents i
4   **if** $B \cap C_h^L \neq \emptyset$ for some C_h
5             $j =$ any element of $C_h^H \setminus B$
6             $j' =$ any element of $B \cap C_h^L$
7             $B = (B \setminus \{j'\}) \cup \{j\}$
8   **else** $j =$ any $c_{\lceil |C_h|/n \rceil}$ not in B
9             $B = B \cup \{j\}$
10   allocate B to some agent i with $v_i(B) \geq \alpha$
11   remove B and i from I and update n, and C_h^H and C_h^L for all h
12 allocate the remaining goods to the last agent

Proof. Because the instance is normalized, the lemma holds at the start of the algorithm. Assume that there are n remaining agents at the start of an iteration, and for each remaining agent i, $v_i(M) \geq |N| - 2(|N| - n)\alpha$. Let i' be the agent receiving B in the iteration. For any remaining agent $i \neq i'$, we wish to show that $v_i(M \setminus B) \geq |N| - 2(|N| - n + 1)\alpha$. Because the valuations are additive, the only way this cannot hold is if $v_i(B) > 2\alpha$. Since any change to B after the initial creation adds a good to B or exchanges a good in B for another, any individual change cannot increase the value of B by more than α. Thus, because the loop at line 3 terminates as soon as $v_i(B) \geq \alpha$, the only way we may have $v_i(B) > 2\alpha$ is if it holds initially, i.e., $B = \bigcup_{h=1}^{\ell} C_h^L$ and $v_i(\bigcup_{h=1}^{\ell} C_h^L) > 2\alpha$. However, by definition $v_i(C_h^L) \leq v_i(C_h)/n$ which implies $v_i(B) \leq v_i(M)/n$. Consequently, $v_i(M \setminus B) \geq (n-1)v_i(B) \geq (n-1)2\alpha \geq (n-1) \geq |N| - 2(|N| - n + 1)\alpha$. ☐

With Lemma 2 we have a sufficient lower guarantee for the remaining value. We are now ready to show the guarantees of the algorithm.

Lemma 3. *Given a normalized ordered instance $I = \langle N, M, V, C \rangle$ of the fair allocation problem under cardinality constraints where all goods are worth less than $\alpha = |N|/(2|N| - 1)$, Algorithm 3 finds a feasible $(|N|/(2|N| - 1))$-approximate MMS allocation in polynomial time in the number of agents and goods.*

Proof. When allocating the remaining goods to the last agent, Lemma 2 guarantees that the goods are worth at least α, if $|N| - 2(|N| - 1)\alpha \geq \alpha$, which holds for $\alpha \leq |N|/(2|N| - 1)$. Additionally, as long as B reaches a value of α before running out of improvement operations, any other agent is also guaranteed to receive a bundle they value at no less than α. Since B contains the $\lceil C_h/n \rceil$ most valuable goods in each category C_h when the algorithm runs out of operations, B reaches a value of at least $1/n$ of the remaining value. We thus only need to show that the remaining value is always at least $n\alpha$ for any remaining agent. Lemma 2 guarantees that the remaining value is at least $|N|-2(|N|-n)\alpha$. Since, this value is at least α for $n=|N|-1$, the value is at least $2(n-1)\alpha+\alpha \geq n\alpha$ for any other n, and we are guaranteed that the value of B reaches at least α in any iteration. Since $\mu_i \leq 1$ for $i \in N$, each agent i receives at least $\alpha\mu_i$ value.

It remains to show that any bundle allocated is feasible. As long as $|C_h| \leq nk_h$, it holds that $\lceil |C_h|/n \rceil \leq k_h$ and any bundle allocated is feasible. Obviously, $|C_h| \leq nk_h$ holds when $n = |N|$, as all instances are assumed to have at least one feasible complete allocation. Assume that $|C_h| \leq nk_h$ holds at the start of an iteration. The bundle B starts with $\lfloor |C_h|/n \rfloor \geq |C_h| - (n-1)k_h$ of the goods in C_h and no good is removed without adding another from the same C_h. Thus, $|C_h \setminus B| \leq (n-1)k_h$ and the condition holds for $n-1$ after allocating B. Consequently, each allocated bundle, including the bundle allocated to the last agent, is feasible.

In each iteration of the algorithm, goods are added to and exchanged through a set of operations. As no good is added back into B after being removed, the number of operations in each iteration is polynomial in the number of agents and goods. Since there are $|N|-1$ iterations, the running time of the algorithm is also polynomial in the number of agents and goods. □

We have now showed everything needed to show that 1/2-approximate MMS allocations exist and can be found in polynomial time.

Theorem 5. *For an instance $I = \langle N, M, V, C \rangle$ of the fair allocation problem under cardinality constraints, a $(|N|/(2|N|-1))$-approximate MMS allocation always exists and can be found in polynomial time.*

Proof. By theorems 2 and 3 and Corollary 1, any instance I can in polynomial time be converted to one, I', that Algorithm 3 accepts. Since I' has no more agents than I, Lemma 3 guarantees that for I' an at least $(|N|/(2|N|-1))$-approximate MMS allocation is found in polynomial time by Algorithm 3. The allocation for I' can then be turned back to one for I in polynomial time. □

Algorithm 3 is guaranteed to find α-approximate MMS allocations for *all* possible problem instances when $\alpha \leq |N|/(2|N|-1)$. However, there exist many types of problem instances for which the algorithm will find a feasible α-approximate MMS allocation when a larger α is used. For example, for an instance where $v_{ij} \leq \mu_i/4$ for all $i \in N$, $j \in M$, the algorithm will always find a feasible α-approximate MMS allocation when $\alpha = 3/4$, because then each bundle allocated in the bag filling step is worth no more than 1, unless the bundle is the starting bag. Generally, increasing α might in the worst case result in the remaining value decreasing to the point where $v_i(B) < \alpha$ for any remaining agent i after all improvements have been performed on B. However, for many problem instances, the average value of each allocated bundle is quite a bit smaller than 2α for any remaining agent i. Thus, even for larger values of α, the algorithm can often find a α-approximate MMS allocation. While it is hard to determine the largest α that works for a certain problem instance through calculation, it is possible to simply check if the algorithm finishes for various values of α. Preliminary experiments suggest that trying the algorithm for a limited number of different values of α often provides much better approximations.

Since Theorem 5 in fact guarantees each agent a bundle of value at least $(|N|/(2|N|-1))v_i(M)$, it directly allows us to show that a 1-out-of-$(2|N|-1)$ MMS allocation always exists and can be found in polynomial time.

Corollary 3. *For an instance $I = \langle N, M, V, C\rangle$ of the fair allocation problem under cardinality constrains, a* 1*-out-of-*$(2|N|-1)$ *MMS allocation always exists and can be found in polynomial time.*

Proof. In a similar fashion to Theorem 2, the 1-out-of-$(2|N|-1)$ MMS of any agent can at most be $v_i(M)/(2|N|-1)$. The proof of Lemma 3 shows that Algorithm 3 gives each agent a bundle valued at least $|N|/(2|N|-1)$ when $v_i(M) = |N|$, which is at least the 1-out-of-$(2|N|-1)$ MMS of any agent. □

It is possible to improve the existence guarantee for MMS approximation by using bag filling in combination with the lone-divider technique of Aigner-Horev and Segal-Halevi [1]. In the lone-divider technique, agent i, one of the remaining agents, is chosen to partition the remaining goods into bundles that all have a value of at least $\alpha\mu_i$ to i. Then, a non-empty subset of the bundles is allocated to some subset of the remaining agents, through an *envy-free matching* which is guaranteed to exist. An envy-free matching is here a matching where each agent matched to a bundle values it at no less than $\alpha\mu_i$ and all non-matched remaining agents value the matched bundles at less than $\alpha\mu_i$. Aigner-Horev and Segal-Halevi showed that an envy-free matching always exists [1]. The process is then repeated until no agent remains. In order to improve the existence guarantee, we first use the lone-divider technique with a partition scheme that only works when a large number of agents remain. When the partition scheme no longer works, the ratio of remaining value to remaining agents has increased, since $\alpha < 1$ and any bundle already allocated is worth less than $\alpha\mu_i$ to any remaining agent. The increased ratio allows Algorithm 3 to be able to provide each remaining agent with a greater value than before the allocations. Unfortunately, the existence result is only of an existential nature, as the partition scheme depends on finding arbitrary MMS-partitions, which is known to be NP-hard [29].

Theorem 6. *For an instance $I = \langle N, M, V, C\rangle$ of the fair allocation problem under cardinality constraints, a $(\sqrt{|N|}/(2\sqrt{|N|}-1))$-approximate MMS allocation always exists.*

Proof sketch (full proof in appendix). When only a few bundles have been given away, any MMS-partition of I for any remaining agent contains at least as many bundles with a remaining value of $\alpha\mu_i$ or higher, as there are remaining agents. The goods in the other bundles in the MMS-partition can then arbitrarily be moved to one of these bundles with remaining value $\alpha\mu_i$. On the other hand, since $\alpha < 1$, as the number of allocated bundles increases, each remaining agent's proportional share of the value of the remaining goods increases. Thus, Algorithm 3 will be able to guarantee a partition with higher and higher minimum bundle value. The value of α must then be set so that in any situation, one of the two methods works. It can be shown that $\sqrt{|N|}/(2\sqrt{|N|}-1)$ is the largest value of α that works. $\square$

6 Uniform Matroid Constraints

In this section we deal with the special case of cardinality constraints in which there is only a single category, i.e., $\ell = 1$. In this case, the cardinality constraints are equivalent to simply limiting the maximum number of goods in a bundle, or, equivalently, restricting bundles to be independent sets of a *uniform matroid*. Throughout the section we will assume that for any ordered instance, which provides a total ordering of goods, the goods are numbered in a way such that $v_i(j) \geq v_i(j')$ for all $i \in N$ and $j, j' \in M$ with $j < j'$. In other words, the goods are numbered from most preferred (1) to least preferred ($|M|$). Our main result (Theorem 7) for single-category instances is the existence of (2/3)-approximate MMS allocations and the ability to find these in polynomial time.

Theorem 7. *For an instance $I = \langle N, M, V, \langle (C_1, k_1) \rangle \rangle$ of the fair allocation problem under cardinality constraints, a* (2/3)*-approximate MMS allocation always exists and can be found in polynomial time.*

In order to prove Theorem 7 we need the following observation about the value of certain subsets of goods.

Lemma 4. *Let $I = \langle N, M, V, \langle (C_1, k_1) \rangle \rangle$ be an ordered instance of the fair allocation problem under cardinality constraints. For any $r \in \{1, 2, \ldots, |N|\}$, let $B_r = \{r, r+1, \ldots, \min(|M|, r + k_1(|N| - r + 1) - 1)\}$. Then, for any $i \in N$, $v_i(B_r) \geq (|N| - r + 1)\mu_i$.*

Lemma 4 provides two useful properties. Most importantly, it can be used to show that the bundles created during a bag-filling style algorithm (Algorithm 4) will be worth at least μ_i before running out of improvements. At the same time, it provides a direct, polynomial way to improve our estimate of μ_i (in addition to Theorem 2) to the required accuracy for the algorithm. Lemma 4 can be used to show that 2/3-MMS allocations can be found in polynomial time for a restricted class of instances using Algorithm 4.

Lemma 5. *For an instance I of the fair allocation problem under cardinality constraints satisfying the requirements of Algorithm 4, the algorithm finds a* 2/3*-approximate MMS allocation in polynomial time.*

Proof sketch (full proof in appendix). The correctness of Algorithm 4 follows from two observations about the construction of B'_j. First, the construction guarantees that B'_j is feasible and contains at least the required number of goods so that after allocating B'_j, there are at most $k_1(j-1)$ goods left. Second, Lemma 4, the incremental improvements of B'_j and the distribution of the $|N|$ most valuable

Algorithm 4 Find (2/3)-MMS solution for single-category instance

Input: An ordered instance $I = \langle N, M, V, \langle C_1, k_1 \rangle\rangle$ with $|M| > |N|$, $\mu_i \leq 1$, $v_i(B_r) \geq |N| - r + 1$ (from Lemma 4), $v_i(1) < 2/3$, and $v_i(|N| + 1) < 1/3$ for every $i \in N$, $r \in \{1, 2, \dots, |N|\}$

Output: Allocation A consisting of each bundle B'_j allocated

let $B'_1 = \{1\}, B'_2 = \{2\}, \dots, B'_{|N|} = \{|N|\}$
for $j = |N|$ down to 1
    if $|M| > k_1(j-1)+1$
        add the $|M| - k_1(j-1) - 1$ least-valuable goods in $M \setminus (B'_1 \cup B'_2 \cup \dots \cup B'_j)$ to B'_j
    while $v_i(B'_j) \leq 2/3$ for all $i \in N$ and $|B'_j| < k_1$
        add the least-valuable good in $M \setminus (B'_1 \cup B'_2 \cup \dots \cup B'_j)$ to B'_j
    while $v_i(B'_j) \leq 2/3$ for all $i \in N$
        exchange the least valuable $g \in B'_j$ for the least valuable $g' \in M \setminus (B'_1 \cup B'_2 \cup \dots \cup B'_j)$ with $g' < g$
    find $i \in N$ such that $v_i(B'_j) \geq 2/3$
    allocate B'_j to i and set $N = N \setminus \{i\}$, $M = M \setminus B'_j$.

goods into distinct bundles, together guarantee that when $j = r$, the value of the $\min(k_1, |B_r \cap M|)$ most valuable remaining goods in B_r is at least 1 for each remaining agent. Thus, B'_j will always be able to reach a value of at least $2/3$. □

Proof sketch for Theorem 7 (full proof in appendix). The proof boils down to showing that for any instance I, we can either trivially, if $|M| \leq |N|$, find a (2/3)-approximate MMS allocation through valid reduction, or we can turn I into an instance accepted by Algorithm 4. The latter is achieved through repeated rescaling based on Theorem 2 and Lemma 4, together with applying all possible valid reductions based on Corollaries 1 and 2. □

In addition to existence of (2/3)-approximate MMS allocations, certain restricted classes of single-category instances allow for better approximation or existence guarantees. Specifically, when the number of goods is not much larger than the category threshold, approximation results for unconstrained fair allocation apply under cardinality constraints.

Lemma 6. *For an instance $I = \langle N, M, V, \langle (C_1, k_1)\rangle\rangle$ of the fair allocation problem under cardinality constraints, with $|M| < |N| + k_1$, MMS-approximation reduces to MMS-approximation for unconstrained fair allocation.*

As a result of Lemma 6, the following follows directly from the results of Garg and Taki on MMS approximation in unconstrained fair allocation [15].

Corollary 4. *For an instance $I = \langle N, M, V, \langle (C_1, k_1)\rangle\rangle$ of the fair allocation problem under cardinality constraints, with $|M| < |N| + k_1$, a $(3/4 + 1/(12n))$-*

approximate MMS allocation always exists and a $(3/4)$*-approximate MMS allocation can be found in polynomial time.*

When the threshold is small enough, it is possible to show that MMS allocations always exist. For larger thresholds, on the other hand, it is possible to create instances for which there is no MMS allocation.

Lemma 7. *Let* $I = \langle N, M, V, \langle (C_1, k_1) \rangle \rangle$ *be an instance of the fair allocation problem under cardinality constraints. If* $k_1 \leq 2$*, an MMS allocation always exists. If* $k_1 \geq 4$*, an MMS allocation is not guaranteed to exist.*

7 Fair Allocation of Chores

So far we have only considered instances where the items are goods. In this section we instead consider instances where the items are chores. As our results on chores are similar in scope and technique to our results on goods, the results will only be covered briefly with all proofs given in the appendix. We assume, without loss of generality, that $v_i(M) < 0$.[8] Then concepts of scale invariance and normalization transfer directly to chores.

Theorem 8 (Scale invariance). *If* A *is an MMS allocation for the instance* $I = \langle N, M, V, C \rangle$ *of the fair allocation of chores problem under cardinality constraints, then* A *is also an MMS allocation of* $I' = \langle N, M, V', C \rangle$*, where* $v'_i(S) = a_i v_i(S)$*,* $a_i > 0$*, for some agent* i*.*

Theorem 9 (Normalization). *Let* $I = \langle N, M, V, C \rangle$ *be an instance of the fair allocation of chores problem under cardinality constraints and* $v_i(M) = -|N|$ *for some agent* i*. Then* $\mu_i \leq -1$*.*

Further, the reduction to ordered instances works for chores as well. As with goods, reassigning the valuations of the chores does not change the MMS of any agent. The earlier conversion algorithm for an allocation of the ordered instance provides each agent with a bundle of equal or higher value (less disutility), which provides an equal or better approximation.

Theorem 10. *For fair allocation of chores under cardinality constraints, MMS-approximation reduces to MMS-approximation of ordered instances in polynomial time.*

For chores, the use of valid reductions does not make sense in the same way as for goods. While valid reductions could still exist and be used, there is a lack of simple rules for finding useful valid reductions. However, we can still bound the number of chores that have a large disutility by exploiting the pigeonhole principle on MMS partitions. Note that Theorem 11 provides a stronger upper bound on the number of high-valued chores than the bounds for goods when $\ell \geq 2$.

[8] As with goods, normalization does not work if $v_i(M) = 0$. In this case, i can be removed from the (ordered) instance by allocating i the k_h worst chores in each C_h. This would constitute a valid reduction.

Theorem 11. *Let $I = \langle N, M, V, C\rangle$ be an instance of the fair allocation of chores problem under cardinality constraints, with $|M| \geq |N|r + 1$ for an $r \in \{0, 1, \dots\}$. For agent $i \in N$, let $g_{i_j} \in M$ denote the j-th most valuable chore in M for i. Then,*

$$v_i(\{g_{i_{|N|r+1-r}}, g_{i_{|N|r+2-r}}, \dots, g_{i_{|N|r+1}}\}) \geq \mu_i$$

Theorems 8, 9 and 11 allow for an easy adjustment of the valuation functions such that for each agent $i \in N$, $\mu_i \leq -1$, $v_i(M) \geq -|N|$ and there are at most $r|N|$ chores that i values at less than $-1/(r+1)$. Crucially, this guarantees that no chore is valued at less than -1, allowing a variant of the bag-filling algorithm used for goods to find 2-approximate MMS allocations.

Theorem 12. *For an instance $I = \langle N, M, V, C\rangle$ of the fair allocation of chores problem under cardinality constraints, a $((2|N| - 1)/|N|)$-approximate MMS allocation always exists and can be found in polynomial time.*

For single-category instances we can also for chores find much better MMS approximate allocations using an algorithm similar to Algorithm 4.

Theorem 13. *For an instance $I = \langle N, M, V, \langle(C_1, k_1)\rangle\rangle$ of the fair allocation of chores problem under cardinality constraints, a $(3/2)$-approximate MMS allocation always exists and can be found in polynomial time.*

8 Discussion

We improved the currently best known MMS approximation guarantees for cardinality constraints by extending the concepts of ordered instances and valid reductions to this setting. Cardinality constraints do, however, impose additional challenges that do not exist in the unconstrained setting, limiting the achievable approximation guarantees. The apparent lack of a common preference ordering between distinct categories limits the degree to which the number of and maximum value of high-valued goods can be restricted—an important factor in improving the approximation guarantee of bag-filling style algorithms. Cardinality constraints also restrict the usability of other types of MMS-approximation algorithms. For example, the lone-divider method may easily allocate bundles that contain many items from a single category and few from others, which in turn can make all further feasible divisions very unbalanced.

Experiments. An earlier version of this preprint (v1) contains some preliminary experimental results, along with source code.

Acknowledgements. The authors wish to acknowledge valuable input from anonymous reviewers.

References

1. Aigner-Horev, E., Segal-Halevi, E.: Envy-free matchings in bipartite graphs and their applications to fair division. Information Sciences **587**, 164–187 (Mar 2022). https://doi.org/10.1016/j.ins.2021.11.059, https://www.sciencedirect.com/science/article/pii/S0020025521011816
2. Amanatidis, G., Markakis, E., Nikzad, A., Saberi, A.: Approximation Algorithms for Computing Maximin Share Allocations. ACM Transactions on Algorithms **13**(4), 52:1–52:28 (Dec 2017). https://doi.org/10.1145/3147173, https://doi.org/10.1145/3147173
3. Babaioff, M., Nisan, N., Talgam-Cohen, I.: Competitive Equilibrium with Indivisible Goods and Generic Budgets. Mathematics of Operations Research **46**(1), 382–403 (Jan 2021). https://doi.org/10.1287/moor.2020.1062, https://pubsonline.informs.org/doi/abs/10.1287/moor.2020.1062, publisher: INFORMS
4. Barman, S., Krishna Murthy, S.K.: Approximation Algorithms for Maximin Fair Division. In: Proceedings of the 2017 ACM Conference on Economics and Computation. pp. 647–664. EC '17, Association for Computing Machinery, Cambridge, Massachusetts, USA (Jun 2017). https://doi.org/10.1145/3033274.3085136, https://doi.org/10.1145/3033274.3085136
5. Bilò, V., Caragiannis, I., Flammini, M.c., Igarashi, A., Monaco, G., Peters, D., Vinci, C., Zwicker, W.S.: Almost Envy-Free Allocations with Connected Bundles. In: Blum, A. (ed.) 10th Innovations in Theoretical Computer Science Conference (ITCS 2019). Leibniz International Proceedings in Informatics (LIPIcs), vol. 124, pp. 14:1–14:21. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany (2018). https://doi.org/10.4230/LIPIcs.ITCS.2019.14, http://drops.dagstuhl.de/opus/volltexte/2018/10107, iSSN: 1868-8969
6. Biswas, A., Barman, S.: Fair Division Under Cardinality Constraints. In: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence. pp. 91–97. International Joint Conferences on Artificial Intelligence Organization, Stockholm, Sweden (Jul 2018). https://doi.org/10.24963/ijcai.2018/13, https://www.ijcai.org/proceedings/2018/13
7. Bouveret, S., Cechlárová, K., Elkind, E., Igarashi, A., Peters, D.: Fair division of a graph. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence. pp. 135–141. International Joint Conferences on Artificial Intelligence Organization, Melbourne, Australia (Aug 2017). https://doi.org/10.24963/ijcai.2017/20, https://www.ijcai.org/proceedings/2017/20
8. Bouveret, S., Chevaleyre, Y., Maudet, N.: Fair Allocation of Indivisible Goods. In: Handbook of Computational Social Choice, pp. 285–310. Cambridge University Press, 32 Avenue of the Americas, New York, NY 10013-2473, USA, 1 edn. (2016), https://www.cambridge.org/no/academic/subjects/computer-science/artificial-intelligence-and-natura
9. Bouveret, S., Lemaître, M.c.: Characterizing conflicts in fair division of indivisible goods using a scale of criteria. Autonomous Agents and Multi-Agent Systems **30**(2), 259–290 (Mar 2016). https://doi.org/10.1007/s10458-015-9287-3, http://link.springer.com/10.1007/s10458-015-9287-3
10. Budish, E.: The Combinatorial Assignment Problem: Approximate Competitive Equilibrium from Equal Incomes. Journal of Political Economy **119**(6), 1061–1103 (Dec 2011). https://doi.org/10.1086/664613,

https://www.journals.uchicago.edu/doi/full/10.1086/664613, publisher: The University of Chicago Press
11. Chiarelli, N., Krnc, M., Milanič, M., Pferschy, U., Pivač, N., Schauer, J.: Fair packing of independent sets. In: Combinatorial Algorithms. pp. 154–165. Lecture Notes in Computer Science, Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-48966-3_12
12. Feige, U., Sapir, A., Tauber, L.: A tight negative example for MMS fair allocations. arXiv:2104.04977 [cs] (Apr 2021), http://arxiv.org/abs/2104.04977, arXiv: 2104.04977
13. Ferraioli, D., Gourvès, L., Monnot, J.: On Regular and Approximately Fair Allocations of Indivisible Goods. In: Proceedings of the 2014 International Conference on Autonomous Agents and Multi-Agent Systems. pp. 997–1004. AAMAS '14, International Foundation for Autonomous Agents and Multiagent Systems, Paris, France (2014). https://doi.org/10.5555/2615731.2617405
14. Garg, J., McGlaughlin, P., Taki, S.: Approximating Maximin Share Allocations. In: Fineman, J.T., Mitzenmacher, M. (eds.) 2nd Symposium on Simplicity in Algorithms (SOSA 2019). OpenAccess Series in Informatics (OASIcs), vol. 69, pp. 20:1–20:11. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany (2019). https://doi.org/10.4230/OASIcs.SOSA.2019.20, http://drops.dagstuhl.de/opus/volltexte/2018/10046
15. Garg, J., Taki, S.: An Improved Approximation Algorithm for Maximin Shares. In: Proceedings of the 21st ACM Conference on Economics and Computation. pp. 379–380. EC '20, Association for Computing Machinery, New York, NY, USA (Jul 2020). https://doi.org/10.1145/3391403.3399526, https://doi.org/10.1145/3391403.3399526, arXiv: 1903.00029
16. Ghodsi, M., Hajiaghayi, M., Seddighin, M., Seddighin, S., Yami, H.: Fair Allocation of Indivisible Goods: Improvements and Generalizations. In: Proceedings of the 2018 ACM Conference on Economics and Computation. pp. 539–556. EC '18, Association for Computing Machinery, Ithaca, NY, USA (Jun 2018). https://doi.org/10.1145/3219166.3219238, https://doi.org/10.1145/3219166.3219238
17. Gourvès, L., Monnot, J.: On maximin share allocations in matroids. Theoretical Computer Science **754**, 50–64 (Jan 2019). https://doi.org/10.1016/j.tcs.2018.05.018, http://www.sciencedirect.com/science/article/pii/S0304397518303384
18. Gourvès, L., Monnot, J., Tlilane, L.: Near fairness in matroids. In: Proceedings of the Twenty-first European Conference on Artificial Intelligence. pp. 393–398. ECAI'14, IOS Press, Prague, Czech Republic (Aug 2014)
19. Greco, G., Scarcello, F.: The Complexity of Computing Maximin Share Allocations on Graphs. Proceedings of the AAAI Conference on Artificial Intelligence **34**(02), 2006–2013 (Apr 2020). https://doi.org/10.1609/aaai.v34i02.5572, https://ojs.aaai.org/index.php/AAAI/article/view/5572, number: 02
20. Hummel, H., Hetland, M.L.: Fair allocation of conflicting items. Autonomous Agents and Multi-Agent Systems **36**(1), 8 (Dec 2021). https://doi.org/10.1007/s10458-021-09537-3, https://doi.org/10.1007/s10458-021-09537-3
21. Hummel, H., Hetland, M.L.: Guaranteeing half-maximin shares under cardinality constraints. In: Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems. pp. 1633–1635. AAMAS '22, International Foundation for Autonomous Agents and Multiagent Systems (2022)

22. Kurokawa, D., Procaccia, A.D., Wang, J.: When can the maximin share guarantee be guaranteed? In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence. pp. 523–529. AAAI'16, AAAI Press, Phoenix, Arizona (Feb 2016)
23. Kurokawa, D., Procaccia, A.D., Wang, J.: Fair Enough: Guaranteeing Approximate Maximin Shares. Journal of the ACM **65**(2), 8:1–8:27 (Feb 2018). https://doi.org/10.1145/3140756, https://doi.org/10.1145/3140756
24. Li, Z., Vetta, A.: The Fair Division of Hereditary Set Systems. ACM Transactions on Economics and Computation **9**(2), 1–19 (Jun 2021). https://doi.org/10.1145/3434410, https://dl.acm.org/doi/10.1145/3434410
25. Lonc, Z., Truszczynski, M.: Maximin Share Allocations on Cycles. In: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18. pp. 410–416. International Joint Conferences on Artificial Intelligence Organization (2018). https://doi.org/10.24963/ijcai.2018/57, https://doi.org/10.24963/ijcai.2018/57
26. Procaccia, A.D., Wang, J.: Fair Enough: Guaranteeing Approximate Maximin Shares. In: Proceedings of the fifteenth ACM conference on Economics and computation. pp. 675–692. EC '14, Association for Computing Machinery, Palo Alto, California, USA (Jun 2014). https://doi.org/10.1145/2600057.2602835, https://doi.org/10.1145/2600057.2602835
27. Shoshan, H., Segal-Halevi, E., Hazon, N.: Efficient Nearly-Fair Division with Capacity Constraints. arXiv:2205.07779 [cs] (May 2022), http://arxiv.org/abs/2205.07779, arXiv: 2205.07779
28. Suksompong, W.: Constraints in fair division. ACM SIGecom Exchanges **19**(2), 46–61 (Dec 2021). https://doi.org/10.1145/3505156.3505162, https://doi.org/10.1145/3505156.3505162
29. Woeginger, G.J.: A polynomial-time approximation scheme for maximizing the minimum machine completion time. Operations Research Letters **20**(4), 149–154 (May 1997). https://doi.org/10.1016/S0167-6377(96)00055-7, http://www.sciencedirect.com/science/article/pii/S0167637796000557

Appendix

A Omitted Proofs for Section 2

Proof (for Theorem 1). Assume that A is not an MMS allocation of I'. Then, there is an agent $i' \in N$ such that $v'_{i'}(A_{i'}) < \mu_{i'}^{I'}$. However, since $v'_{i'}(j) = a_{i'}v_{i'}(j)$, it follows that

$$\mu_{i'}^{I'} = \max_{A' \in \mathcal{F}_{I'}} \min_{A_j \in A'} v'_{i'}(A_j) = \max_{A' \in \mathcal{F}_{I'}} \min_{A_j \in A'} a_{i'}v_{i'}(A_j) = a_{i'}\mu_{i'}^{I}$$

Consequently, $v'_{i'}(A_{i'}) = a_{i'}v_{i'}(A_{i'}) \geq a_{i'}\mu_{i'}^{I} = \mu_{i'}^{I'}$. This is a contradiction, and there can be no i' with $v'_{i'}(A_{i'}) < \mu_{i'}^{I'}$. Hence, A is an MMS allocation of I'.

Proof (for Theorem 2). Since the valuations are additive, for any allocation A of I, we have that $v_i(M) = \sum_{A_j \in A} v_i(A_j)$. Consequently, the value of the least valuable bundle $B^* \in A$ must be such that $|N|v_i(B^*) \leq v_i(M)$, as otherwise, $\sum_{A_j \in A} v_i(A_j) > v_i(M)$. Therefore,

$$\mu_i = \max_{A \in \mathcal{F}_I} \min_{A_j \in A} v_i(A_j) \leq \max_{A \in \mathcal{F}_I} \frac{v_i(M)}{|N|} = 1$$

□

B Omitted Proofs for Section 3

Proof (for Lemma 1). First, note that creating an ordered instance does not affect any agent's MMS: For any given agent, Algorithm 1 implicitly defines a one-to-one mapping on each category C_h, corresponding to a permutation of the good values. Because the valuations are only interchanged within each C_h, the map can be used to convert any allocation between the two instances, preserving feasibility, without changing the value of any individual bundle. The MMS of the agent is independent of the valuations of other agents, and so the MMS must be the same in both instances.

As algorithms 1 and 2 are equivalent to those of Bouveret and Lemaître within each individual category, it follows that the value an agent receives from each category is at least as high in allocation A for the original instance as in allocation A' for the ordered instance. Thus, the agent's total value in A for the original instance must at least be as high as in A' for the ordered instance. Neither algorithm will introduce violations of the cardinality constraints, and since the MMS is unchanged between the two instances, it follows that the new allocation is also a feasible α-approximate MMS allocation. □

C Omitted Proofs from Section 4

Proof (for Theorem 4). For B' and i to be a valid reduction we must show that (i) B' is a feasible bundle, (ii) $v_i(B') \geq \alpha\mu_i$, (iii) $I' = \langle N \setminus \{i\}, M \setminus B', V', C'\rangle$ has at least one feasible allocation and (iv) $\mu_{i'}^{I'} \geq \mu_{i'}^{I}$ for all $i' \in N \setminus \{i\}$.

For B' to be feasible, then it must hold that $|C_h \cap B'| \leq k_h$ for any category $C_h \in C$. Since B is feasible ($|C_h \cap B| \leq k_h$), we have that:

$$\begin{aligned} |C_h \cap B'| &= |C_h \cap B| + \max(0, |C_h \setminus B| - (|N| - 1)k_h) \\ &= |C_h \cap B| + \max(0, |C_h| - |C_h \cap B| - (|N| - 1)k_h) \\ &\leq |C_h \cap B| + |N|k_h - |C_h \cap B| - (|N| - 1)k_h \\ &\leq k_h. \end{aligned}$$

Consequently, (i) holds. Further, since $B \subseteq B'$ we have $v_i(B') \geq v_i(B) \geq \alpha\mu_i$ and (ii) also holds. For (iii), note that if $|C_h \setminus B| > (|N|-1)k_h$ for some $C_h \in C$, B' contains $|C_h \setminus B| - (|N| - 1)k_h$ additional goods from C_h. Consequently, $|C_h \setminus B'| \leq (|N| - 1)k_h$ and $C_h \setminus B'$ does not contain more goods than can be given to $|N| - 1$ agents. Therefore, I' has a feasible allocation and (iii) holds.

For simplicity, we now assume without loss of generality that if $j \in B$ and $j \in B_{i'}$, then $f(j) = j$. If this does not hold for f, then we can define a new injective function $f' : B \to B_{i'}$ satisfying the criteria of the theorem by

$$f'(j') = \begin{cases} j & \text{if } j' = j \\ f(j) & \text{if } f(j') = j \\ f(j') & \text{otherwise} \end{cases}$$

for all $j' \in B$. The requirements of the theorem also hold for f', as it is injective, j, j' and $f(j)$ all belong to the same category and $v_{i'}(j') \leq v_{i'}(j) \leq v_{i'}(f(j))$.

For (iv) we want to show that for any agent $i' \in N \setminus \{i\}$, we can modify their MMS partition such that $B_{i'}$ is converted into B while maintaining a feasible partition and without decreasing the value of any other bundle in the partition. In other words, we want to show that there is a feasible partition containing B, and $(|N| - 1)$ bundles with a value of no less than $\mu_{i'}^{I}$ to i'. The transformation can be achieved by performing three steps in order:

1. For each $j \in B$, exchange the placement of j and $f(j)$ in the MMS partition.
2. While $|C_h \cap B'| < |C_h \cap B_{i'}|$, move a good in $(C_h \cap B_{i'}) \setminus B'$ to any other bundle $B_{i''}$ with $|C_h \cap B_{i''}| < k_h$.
3. While $C_h \cap B' \neq C_h \cap B_{i'}$, exchange any good in $(C_h \cap B_{i'}) \setminus B'$ for a good of equivalent or lower value in $(C_h \cap B') \setminus B_{i'}$.

Note that non of the operations can cause infeasibility, as both goods in steps 1 and 3 belong to the same category. Additionally, in step 2, $B_{i''}$ has space for at least one more good from C_h.

The value of any other bundle than $B_{i'}$ will not decrease during the transformation. In steps 1 and 3 the good removed from $B_{i'}$ has a value that is either equivalent to or higher than the one it is exchanged for. As each good has a non-negative value, the value of the receiving bundle cannot decrease in step 2. Since each of the bundles in the MMS partition had a value of at least μ_i^I prior to the transformation, there are at least $|N| - 1$ bundles with a value of at least μ_i^I after the transformation.

We now need to show that each step can be performed, and that $B' = B_{i'}$ after step 3. Exchanging the position of two goods from the same category is always possible, and step 1 can always be performed. Additionally, as f is injective and $f(j) = j$ for any $j \in B$ with $j \in B_{i'}$, we have that for $j \in B$ either $f(j) \notin B$ or $f(j) = j$. Consequently, each exchange either brings j into $B_{i'}$ without removing any item from B or exchanges j for itself within $B_{i'}$. After step 1, we thus have $B \subseteq B_{i'}$.

In step 2, we know that $|(C_h \cap B_{i'}) \setminus B'| \geq |C_h \cap B_{i'}| - |C_h \cap B'| > 0$ whenever the condition of the step holds. The only way the step can fail is thus that there is no bundle $B_{i''}$. However, by the construction of B' and as I is feasible, we have that when selecting $B_{i''}$:

$$|C_h| \leq (|N| - 1)k_h + |C_h \cap B'| \leq (|N| - 1)k_h - 1 + |C_h \cap B_{i'}|,$$

and one of the other bundles contains less than k_h goods from C_h.

We claim that after step 2, $|C_h \cap B'| = |C_h \cap B_{i'}|$ for each category $C_h \in C$. If this does not hold, then $|C_h \cap B'| > |C_h \cap B_{i'}|$ after step 1. We know that either $B \cap C_h = B' \cap C_h$ or $|C_h \setminus B'| = (|N| - 1)k_h$. If $B \cap C_h = B' \cap C_h$, then $(B' \cap C_h) \subseteq B_{i'}$ and if $|C_h \setminus B'| = (|N| - 1)k_h$ then, $|B' \cap C_h| > |B_{i'} \cap C_h|$ would imply infeasiblity. Consequently, $|B' \cap C_h| = |B_{i'} \cap C_h|$ for each category $C_h \in C$ after step 2. Additionally, no good in B' is removed from $B_{i'}$ in this step and it still holds that $B \subseteq B_{i'}$.

The only way that step 3 can fail is if $(C_h \cap B_{i'}) \setminus B' \neq \emptyset$, but there is no good in $(C_h \cap B') \setminus B_{i'}$ of equivalent or lower value. Note that $(C_h \cap B') \setminus B_{i'}$ always contains the same number of goods as $(C_h \cap B_{i'}) \setminus B'$ as the sets had equivalent size after step 2 and each exchange in step 3 reduces the size of both by 1. Since $B \subseteq B_{i'}$, for any good $j \in (C_h \cap B') \setminus B_{i'}$ we have $j \notin B$ and by construction B' contains all goods in C_h of lower value than j. Therefore, $(C_h \cap B_{i'}) \setminus B'$ cannot contain a good of lower value than j and any good in $(C_h \cap B') \setminus B_{i'}$ can be used in the exchange. In other words, step 3 will never fail. Finally, after step 3 we have that $(C_h \cap B') \setminus B_{i'} = \emptyset = (C_h \cap B_{i'}) \setminus B'$ for each category $C_h \in C$ which implies $B' = B_{i'}$. Thus, (iv) holds. □

Proof (for Corollary 1). Each good must appear in exactly one bundle in any MMS partition. Thus, each agent $i' \in N \setminus \{i\}$ has in each one of their MMS partitions a bundle $B_{i'}$, with $j \in B_{i'}$. Consequently, there exists an injective map $f : \{j\} \to B_{i'}$ satisfying the conditions of Theorem 4, namely $f(j) = j$. Since $\{j\}$ is a feasible bundle (otherwise $\mathcal{F}_I = \emptyset$), all the conditions of Theorem 4 are satisfied and $\{j\}$ can be used to create a valid reduction by setting $B = \{j\}$ and creating B' as in Theorem 4. □

Proof (for Corollary 2). Since there are exactly $|N|$ bundles in any MMS partition and $|C_h| \geq |N| + 1$, any agent $i' \in N \setminus \{i\}$ must by the pigeonhole principle in any one of their MMS partitions have a bundle $B_{i'}$ that contains at least two goods from $\{c_1, c_2, \ldots, c_{|N|+1}\}$. Since $c_{|N|}$ and $c_{|N|+1}$ are the two least valuable goods in $\{c_1, c_2, \ldots, c_{|N|+1}\}$, $B_{i'}$ contains goods $j, j' \in \{c_1, c_2, \ldots, c_{|N|+1}\}$, $j \neq j'$, such that $v_{i'}(j) \geq v_{i'}(c_{|N|})$ and $v_{i'}(j') \geq v_{i'}(c_{|N|+1})$. Consequently, the map $f : \{c_{|N|}, c_{|N|+1}\} \to B_{i'}$ with $f(c_{|N|}) = j$ and $f(c_{|N|+1}) = j'$ satisfies the conditions of Theorem 4. Since $\mathcal{F}_{\mathcal{I}} \neq \emptyset$, the bundle $\{c_{|N|}, c_{|N|+1}\}$ is feasible and Theorem 4 says that $\{c_{|N|}, c_{|N|+1}\}$ can be used to create a valid reduction. □

D Omitted Proofs from Section 5

Proof (for Theorem 6). To show that a certain approximation guarantee can be fulfilled by the lone-divider technique, it suffice to show that after $b < |N|$ bundles have been allocated, any one of the remaining agents i can partition the remaining goods into $|N| - b$ feasible bundles each with a value of at least $\alpha\mu_i$. We wish to show that this is possible when $\alpha = (\sqrt{|N|})/(2\sqrt{|N|} - 1)$. In the first step this is obviously true, as any MMS partition of the selected agent can be used. In any other step, we will use one of two possible strategies to divide the goods. Note that since each step allocates one or more bundles from a feasible allocation, there at any point remains at most as many goods from each category as can be allocated to the remaining agents.

First, notice that when b bundles have been allocated, the remaining value is at least $|N|\mu_i - b\alpha\mu_i$ for any remaining agent i. By Lemma 3, Algorithm 3 can be used to find a partition of the remaining goods so that the value of each bundle is at least a $(|N| - b)/(2(|N| - b) - 1)$ share of the remaining value, given that any single good is not worth more than this share. If the remaining value is at least $(|N| - b - 1)2\alpha\mu_i + \alpha\mu_i$, then this guarantees that the method gives each remaining agent a bundle of value at least $\alpha\mu_i$. Through valid reductions, the condition on the maximum value of any individual good can easily be achieve before the lone-divider technique is applied. Note that when Algorithm 3 is applicable, a final allocation can immediately be achieved, rather than having to following the lone-divider strategy any further. Thus, we only wish to show that the lone-divider strategy can be performed until the remaining value to agent ratio is high enough.

If $\alpha = (\sqrt{|N|})/(2\sqrt{|N|} - 1)$, the remaining value may initially be less than $2(|N| - b - 1)\alpha\mu_i + \alpha\mu_i$. However, as long as $b\alpha\mu_i \leq (b+1)(1-\alpha)\mu_i$, the partition created by taking one of i's MMS partitions and removing all already allocated goods, contains at least $|N| - b$ bundles with a value of at least $\alpha\mu_i$. This partition can contain more than $|N| - b$ non-empty bundles. To turn it into a partition that contains exactly $|N| - b$ bundles of value at least $\alpha\mu_i$, select a subset of $|N| - b$ bundles, each with a value of at least $\alpha\mu_i$. For any other bundle, transfer the goods to any one of the bundles in the selected subset that has space for more goods. There is always a bundle with space left, as there are at most as many remaining goods in each category as can fit in $|N| - b$ bundles.

If for some remaining agent i, $b\alpha\mu_i \leq (b + 1)(1 - \alpha)\mu_i$, then another step of the lone-divider strategy can be performed by selecting i. We thus need to show that for $\alpha = (\sqrt{|N|})/(2\sqrt{|N|}-1)$, when $b\alpha\mu_i > (b+1)(1-\alpha)\mu_i$ for each remaining agent i, then $|N|\mu_i - b\alpha\mu_i \geq (|N|-b-1)2\alpha\mu_i + \alpha\mu_i$ for all remaining agents i. If $\mu_i = 0$, then $b\alpha\mu_i = 0 = (b+1)(1-\alpha)\mu_i$. So, we can without loss of generality assume that $\mu_i = 1$ for all remaining agents i when $b\alpha\mu_i > (b+1)(1-\alpha)\mu_i$. Since $\alpha > 1/2$, we have the following when $b\alpha > (b+1)(1-\alpha)$:

$$\begin{aligned} b\alpha &> (b+1)(1-\alpha) \\ 2b\alpha - b &> 1-\alpha \\ b &> \frac{1-\alpha}{2\alpha-1} \end{aligned}$$

We now need to show that when $b > \frac{1-\alpha}{2\alpha-1}$, then for $\alpha = (\sqrt{|N|})/(2\sqrt{|N|}-1)$ we have $|N| - b\alpha \geq (|N|-b-1)2\alpha + \alpha$. That is, we need to show that

$$\begin{aligned} |N| - b\alpha &\geq (|N|-b-1)2\alpha + \alpha \\ |N| &\geq (2|N|-b-1)\alpha \\ &\geq \left(2|N| - \frac{1-\alpha}{2\alpha-1} - 1\right)\alpha \\ &\geq 2|N|\alpha - \frac{\alpha-\alpha^2}{2\alpha-1} - \frac{2\alpha^2-\alpha}{2\alpha-1} \\ &\geq 2|N|\alpha - \frac{\alpha^2}{2\alpha-1} \end{aligned}$$

Which can be reorganized to $4\alpha|N| - |N| \geq (4|N|-1)\alpha^2$. Using $\alpha = \frac{\sqrt{|N|}}{2\sqrt{|N|}-1}$, we get that

$$\begin{aligned} 4\alpha|N| - |N| &= \frac{4|N|\sqrt{|N|}}{2\sqrt{|N|}-1} - |N| \\ &= \frac{(4|N|\sqrt{|N|})(2\sqrt{|N|}-1)}{(2\sqrt{|N|}-1)^2} - \frac{|N|(2\sqrt{|N|}-1)^2}{(2\sqrt{|N|}-1)^2} \\ &= \frac{8|N|^2 - 4|N|\sqrt{|N|}}{(2\sqrt{|N|}-1)^2} - \frac{4|N|^2 - 4|N|\sqrt{|N|} + |N|}{(2\sqrt{|N|}-1)^2} \\ &= \frac{4|N|^2 - |N|}{(2\sqrt{|N|}-1)^2} \\ &= \frac{(4|N|-1)|N|}{(2\sqrt{|N|}-1)^2} \\ &= (4|N|-1)\left(\frac{\sqrt{|N|}}{2\sqrt{|N|}-1}\right)^2 \\ &= (4|N|-1)\alpha^2 \end{aligned}$$

Since the left and right hand sides are equal, $4\alpha|N| - |N| \geq (4|N|-1)\alpha^2$ and setting $\alpha = (\sqrt{|N|}))/(2\sqrt{|N|}-1)$ guarantees that at least one of the two methods work. Consequently, a $(\sqrt{|N|}/(2\sqrt{|N|}-1))$-approximate MMS allocation always exists. Since the two sides are equal for all possible values of $|N|$, this is the greatest α this method works for. $\square$

E Omitted Proofs from Section 6

Proof (for Lemma 4). Let A be an MMS partition of i and $S = \bigcup_{A_k \in A^*} A_k$, where A^* is a set of $|N| - r + 1$ bundles in A with $A_k \cap \{1, 2, \ldots, r-1\} = \emptyset$ for each $A_k \in A^*$. Since $0 < r \leq |N|$ and at most $r-1$ bundles can contain goods from $\{1, 2, \ldots, r-1\}$, there are at least $|N| - (r-1) = |N| - r + 1$ bundles that can be used in A^*. Further, there is no good $j \in (M \setminus (\{1, 2, \ldots, r-1\} \cup B_r))$ with $v_i(j) > v_i(j')$ for any good $j' \in B_r$. Thus, we have that for $j \in S$, either $j \in B_r$ or $v_i(j') \geq v_i(j)$ for all $j' \in B \setminus S$. As either $|B_r| = k_1(|N| - r + 1)$ or $|B_r| = |M \setminus \{1, 2, \ldots, r-1\}|$, we have $|B_r| \geq |S|$, and

$$v_i(B_r) \geq v_i(S) = \sum_{A_k \in A^*} v_i(A_k) \geq \sum_{A_k \in A^*} \mu_i = (|N| - r + 1)\mu_i$$

$\square$

Proof (for Lemma 5). To prove that the algorithm finds a (2/3)-MMS allocation in polynomial time, we wish to show that (i) any B'_j allocated is feasible, (ii) all goods are allocated, (iii) both while-loops (lines 5 and 7) finish running for each B'_j, and (iv) the algorithm finishes in polynomial time. Note that (iii) guarantees that there is always an agent that values B'_j at 2/3. As a consequence, as long as (iii) holds, each agent $i \in N$ will receive a bundle worth $(2/3)\mu_i$.

Feasible and complete allocation

For (i) and (ii), we wish to show that at the end of each iteration of the for loop, $|M| \leq k_1|N|$. By the assumption that any instance has at least one feasible allocation, this holds before the first iteration. Assume that this holds at the start of a iteration for a specific j, i.e., $|M| \leq jk_1$. Then, we must show that $|M| - |B'_j| \leq (j-1)k_1$ when B'_j is allocated to an agent. Throughout an iteration, the size of B'_j never decreases. If line 4 is executed, then $|B'_j| > 1 + (|M| - k_1(j-1) - 1) = |M| - k_1(j-1)$ and it follows that $|M| - |B'_j| < |M| - (|M| - k_1(j-1)) = k_1(j-1)$. If on the other hand, line 4 is not executed, then $|M| \leq k_1(j-1) + 1 = k_1(j-1) + |B'_j|$. Consequently, $|M| - |B'_j| \leq (j-1)k_1$ always holds. After the last iteration, $|N| = 0$ implying that $|M| \leq 0k_1 = 0$. Thus, (ii) must hold as long as Algorithm 4 finishes. Since the loop on line 5 only adds items as long as $|B'_j| < k_1$ and the loop on line 7 does not change $|B'_j|$, $|B'_j| \leq k_1$ as long as line 4 does not add too many goods to B'_j. However, since $|M| \leq jk_1$, the line adds at most $jk_1 - k_1(j-1) - 1 = k_1 - 1$ goods to B'_j, which at this point only contains a single good. Thus, (i) holds.

Bundle value always reaches 2/3

For (iii) we wish to show that during the execution of the first loop (line 5), either $v_i(B'_j) \geq 2/3$ for some $i \in N$ or $|B'_j| = k_1$. Additionally, we wish to show that during the second loop, $v_i(B'_j) \geq 2/3$ for some $i \in N$. In order to show this, let B_r^j denote the bundle consisting of the $\min(|B_r \cap M|, k_1(j-r+1))$ most valuable remaining goods in B_r (i.e., in $B_r \cap M$) at the start of iteration j for all $r \leq j$. We wish to show that in any iteration j, for all $i \in N$ we have $v_i(B_r^j) \geq j - r + 1$. This will in turn guarantee that $v_i(B_r^j) \geq 1$ for all remaining agents i when $j = r$, and the bundle B'_j will be able to reach a value of $2/3$ either in the first loop when $|M \setminus (B'_1 \cup B'_2 \cup \cdots \cup B'_{j-1})| \leq k_1$, or otherwise by the time the second loop runs out of improvements to make, as $|B_r^r| \leq k_1$.

By Lemma 4, it follows that when $j = |N|$ in the first iteration, $B_r = B_r^j$. Thus, $v_i(B_r^j) = v_i(B_r) \geq j - r + 1$ and the invariant holds initially. Now assume that $v_i(B_r^j) \geq j - r + 1$ for some $j > 1$ and all $r \leq j$. We wish to show that the invariant then also holds for $j' = j - 1$. In other words, we wish to show that $v_i(B_r^{j'}) > (j' - r + 1)$ for all $r \leq j'$ and remaining agents i. There are two possible situations that need to be accounted for, depending on if $v_i(B'_j) > 1$ or $v_i(B'_j) \leq 1$ for agent i.

Since any good $g \in (M \setminus (B'_1 \cup B'_2 \cup \cdots \cup B'_j))$ has $v_i(g) < 1/3$, the only way for $v_i(B'_j)$ to be greater than 1 is if this occurs when the bundle is modified on line 4. In that case, the bundle will not be modified further before it is allocated and we have that $|M \setminus B'_j| = j'k_1 \implies |B_r^{j'}| = k_1(j' - r + 1)$. Further, since B'_j consist of j and the at most $k_1 - 1$ least valuable goods in M, we know that $B_r^{j'}$ consists of, in addition to $\{r, r+1, \ldots j'\}$, $(k_1 - 1)(j' - r + 1)$ goods in M, each with at least the same value as any of the maximum $k_1 - 1$ goods in $B'_j \setminus \{j\}$. In other words,

$$\begin{aligned} v_i(B_r^{j'}) &= v_i(\{r, r+1, \ldots, j'\}) + v_i(B_r^{j'} \setminus \{r, r+1, \ldots, j'\}) \\ &\geq (j' - r + 1)v_i(j) + (j' - r + 1)v_i(B'_j \setminus \{j\}) \\ &= (j' - r + 1)v_i(B'_j) \\ &> j' - r + 1 \end{aligned}$$

This leaves the case where $v_i(B'_j) \leq 1$. In this case we distinguish between two cases, depending on if $B_r^{j'} = B_r^j \setminus B'_j$ or not. If this holds true, then

$$v_i(B_r^{j'}) = v_i(B_r^j \setminus B'_j) \geq v_i(B_r^j) - v_i(B'_j) > (j - r + 1) - 1 = j' - r + 1$$

If $B_r^{j'} \neq B_r^j \setminus B'_j$, we claim that $B'_j \setminus \{j\}$ does not contain any good better than the $(j-r)k_1 + 3$ most valuable good in B_r^j. If a good g, that is the $(j-r)k_1 + 2$ most valuable good in B_r^j or better, is added to B'_j on line 4 or in the first loop (line 5), then B'_j contains all goods in M that are worse than g, and by definition all goods in B_r^j that are worse than g. This is a contradiction, as then $|B_r^j \setminus B'_j| \leq k_1(j-1)$ and $M \setminus (B'_1 \cup B'_2 \cup \cdots \cup B'_j) = B_r^j \setminus B'_j$, which implies that $B_r^{j'} = B_r^j \setminus B'_j$. Thus, the only way that such a g could be added to B'_j and maintain $B_r^{j'} \neq B_r^j \setminus B'_j$, is in the second loop (line 7). However, when a good g is added to B'_j in the second loop, $|B'_j| = k_1$ and B'_j contains the goods $\{g+1, g+2, \ldots, g+k_1-2\}$. Thus, since g is the $(j-r)k_1 + 2$ most valuable good in B_r^j or better, $g + k_1 - 2 \in B_r^j$ as B_r^j by definition contains as many goods as possible up to $(j-r+1)k_1$ goods and $g + k_1 - 2$ is the $(j-r)k_1 + 2 + k_1 - 2 = (j-r+1)k_1$ good in B_r^j. Consequently, $B'_j \in B_r^j$ and $B_r^{j'} = B_r^j \setminus B'_j$ which is another contradiction and the claim holds.

It remains to show that when B'_j contains j and no better than the $(j-r)k_1 + 3$ good in B_r^j, then $v_i(B_r^{j'}) \geq (j' - r + 1)$. We can divide B_r^j into $j - r + 1$ bundles of at most k_1 goods each, by creating the bundles $\{r\}, \{r+1\}, \ldots, \{j\}$ and then as long as there remains goods in B_r^j, placing the most valuable remaining good into the first of the bundles which does not have k_1 goods yet. Then the bundle that started with the good j contains, except for j, no good better than the $(j-r)k_1 + 2$ most valuable good in B_r^j and since j is the least valuable good of the ones initially placed in the bundles, the last bundle is the least valuable of any of the bundles. In other words, the value of all but the last bundle is at least $((j-r)/(j-r+1))v_i(B_r^j) = j - r = j' - r + 1$. Since B'_j only intersects this last bundle, and there are no more than $(j-r)k_1$ goods in the other bundles, there remains at least $(j'-r+1)k_1$ goods in B_r^j with a combined value of at least $j' - r + 1$ and the invariant holds for j'. By induction it holds for all values of j and (iii) holds.

Polynomial run time

It remains to show that (iv) holds, namely that the algorithm uses polynomial time. The outer loop has as many iteration as there are agents, so it suffice to show that each iteration is polynomial. The loop on line 5 runs for at most $k_1 - 1$ iterations, as $|B'_j| \geq 1$ prior to the first iteration and each iteration increases the size of B'_j by 1 up to a maximum of k_1. The loop on line 7 also runs for a maximum of $|M|$ iterations, as each iteration exchanges the least valuable good $g \in B'_j$ for the least valuable, but better good in $M \setminus (B'_1 \cup B'_2 \cup \cdots \cup B'_j)$. Since g now is worse than all goods in B'_j, it will never be picked again. Thus, the size of the set of goods in $M \setminus (B'_1 \cup B'_2 \cup \cdots \cup B'_j)$ that will never be picked again increases by 1 each iteration and after (less than) $|M|$ iterations there is no good to pick. Since each loop has a polynomial number of iterations and each individual operation can be performed in polynomial time, the algorithm must finish in polynomial time and (iv) holds. $\square$

Proof (of Theorem 7). We wish to show that for any instance I we can in polynomial time either convert the instance into one that Algorithm 4 accepts or failing that we can directly create a (2/3)-approximate MMS allocation.

We can by Theorem 3 convert I to an ordered instance. The other requirements for Algorithm 4 can be achieved by performing the following steps. Note that step 3 may cause some remaining agent i' to have $v_{i'}(M) = 0$, in which case step 1 cannot possibly rescale the agent's valuations so that $v_{i'} = |N|$. In this case, just skip the rescaling of agent i''s valuations, and agent i' will at some point be reduced away in step 3. The same may happen for i' and step 2 if $v_{i'}(|N|) = 0$. In which case step 3 will find a valid reduction with i'.

1. For all $i \in N$, rescale i's valuations so that $v_i(M) = |N|$.
2. If for any $i \in N, r \in \{1, 2, \ldots, |N|\}$, $v_i(B_r) < |N| - r + 1$, rescale i's valuations so that $v_i(B_r) = |N| - r + 1$.
3. If $v_i(1) \geq 2/3$ or $v_i(\{|N|, |N|+1\}) \geq 2/3$ for $i \in N$, construct a valid reduction with, respectively, $\{1\}$ or $\{|N|, |N|+1\}$ and agent i, and go back to step 1.

First, we wish to show that after step 2, it holds that $\mu_i \leq 1$ and $v_i(B_r) \geq |N| - r + 1$ for all i and r. By Theorem 2 it holds that $\mu_i \leq 1$ for all $i \in N$ after step 1. In step 2, the rescaling increases the value of all goods (of non-zero value). Thus, rescaling for a specific i and r does not decrease the value of $v_i(B_{r'})$ for $r' \in \{1, 2, \ldots, |N|\}$. Hence, after step 2 it holds that $v_i(B_r) \geq |N| - r + 1$ for all $i \in N$, $r \in \{1, 2, \ldots, |N|\}$. By Lemma 4, we know that $v_i(B_r) \geq (|N| - r + 1)\mu_i$. Since each rescaling sets $v_i(B_r) = |N| - r + 1$, it follows that $\mu_i \leq 1$.

Since $\mu_i \leq 1$, Corollaries 1 and 2 guarantee that if one of the conditions in step 3 hold, a valid reduction can be created for $\alpha = 2/3$. Thus, it also holds that when no valid reduction is found in step 3, then for all $i \in N$ we have $v_i(1) < 2/3$ and $v_i(\{|N|, |N|+1\}) < 2/3 \implies v_i(|N|+1) < 1/3$.

The only missing condition of Algorithm 4 is $|M| > |N|$. If $|M| \leq 2|N|$ at any point, it must hold for any $i \in N$ that either $v_i(1) \geq 2/3$ or $v_i(|N|+1) \geq 1/3$ when $v_i(M) \geq |N|$ and $|M| > 0$. Therefore, when step 3 finishes without going back to step 1, either $|M| = 0$ or $|M| > 2|N| > |N|$. In the first case, we already have a (2/3)-approximate MMS allocation and in the latter case the missing condition of Algorithm 4 holds for the instance.

Since both r and i are bounded in the number of agents, it can easily be verified that each individual step can be performed in polynomial time. As each valid reduction removes an agent, the number of times the steps are performed is also bound in the number of agents. The preprocessing can therefore be done in polynomial time. □

Proof (for Lemma 6). Any feasible allocation for an instance I of the fair allocation problem under cardinality constraints is also a feasible allocation for the unconstrained instance $I' = \langle N, M, V \rangle$. By definition we thus have that $\mu_i^{I'} \geq \mu_i^I$. If no agent $i \in N$ has $\mu_i^I = 0$, then for any $\alpha > 0$, any α-approximate MMS allocation of I' must allocate at least one good to each agent i. Consequently, no agent may receive more than $|M| - |N| + 1 < |N| + k_1 - |N| + 1 = k_1 + 1$ goods. Any α-approximate MMS allocation of I' is then a feasible α-approximate MMS allocation for I.

By Theorem 3 we can assume that I is ordered. If $\mu_i = 0$ for an agent $i \in N$, then $v_i(j) \geq \mu_i$ for all $j \in M$ and Corollary 1 can be used to reduce away i. Since the reductions remove one item along with every agent removed, the conditions of the lemma still hold for the reduced instance. The check can be performed in polynomial time, as $\mu_i = 0 \Leftrightarrow v_i(|N|) = 0$. □

Proof (for Lemma 7). To show the existence of an MMS allocation when $k_1 \leq 2$, we wish to show that the instance I can be reduced to an ordered instance $I' = \langle N', M', V', \langle (C_1', k_1) \rangle \rangle$ with $|M'| = 2|N'|$, in which the allocation $A =$

$\langle\{1, 2|N'|\}, \{2, 2|N'|-1\}, \ldots, \{|N'|, |N'|+1\}$ is an MMS partition for all agents in I'.

By Theorem 3 we can assume that I is ordered. If $|M| < 2|N|$, then in any allocation there is at least one bundle containing only a single item. In other words, Corollary 1 allows for a valid reduction with any agent $i \in N$ and the good 1 for $\alpha = 1$. Thus, repeated reductions can be performed until we have an instance I' where either $|M'| = 0$ and we have found an MMS allocation or $|N'| > 0$ and $|M'| = 2|N'|$. In the second case, for any agent $i \in N'$ let A' be any MMS partition of I for i. We wish to show that A' can be turned into A without reducing the value of the least-valuable bundle in A'.

Let g be the first good in $\{1, 2, \ldots, |N|\}$ such that $\{g, 2|N'| - g + 1\} \notin A'$. Then A' contains distinct bundles $B_g = \{g, g'\}$ and $B_{2|N'|-g+1} = \{g'', 2|N'| - g + 1\}$. Since g was selected to be the smallest g for which this holds, all less-valuable goods than $2|N'| - g + 1$ and more valuable than g appear in other bundles than B_g and $B_{2|N'|-g+1}$. Consequently, we have $v_i(g') \geq v_i(2|N'| - g + 1)$, $v_i(g'') \geq v_i(2|N'| - g + 1)$ and $v_i(g) \geq v_i(g'')$. Thus, $v_i(\{g'', g'\}) \geq v_i(B_{2|N'|-g+1})$ and $v_i(\{g, 2|N'| - g + 1\}) \geq v_i(B_{2|N'|-g+1})$. We can swap the location of g' and $2|N'| - g + 1$ to create an allocation where the worst bundle is no worse than $B_{2|N'|-g+1}$ and that shares one more bundle with A. This can be repeated until the allocation shares all its bundles with A. In other words, A is an MMS partition of i. Since all agents share the same MMS partition, it is also an MMS allocation.

For $k_1 \geq 4$, we will show that there for any $k_1 \geq 4$ exists an instance of the problem for which no MMS allocation exists. The instance will be created by introducing cardinality constraints to the unconstrained instance of Feige et al. with 3 agents and 9 goods for which they showed that the best possible allocation achieves no more than an approximation ratio of 39/40 [12]. Feige et al.'s proof used MMS partitions that contained no more than four goods in each bundle. By introducing cardinality constraints with a single category and a threshold of $k_1 \geq 4$, these MMS partitions remain feasible and each agent's MMS stays the same. By introducing cardinality constraints, the set of feasible allocations is a subset of the set of allocations for the unconstrained instance. Consequently, no feasible allocation can provide all the agent's with more value than the best allocation in the unconstrained instance, and there does not exist any MMS allocation. □

F Omitted Proof from Section 7

Proof (for Theorem 8). Exactly as the proof of Theorem 1. □

Proof (for Theorem 9). Exactly as the proof of Theorem 2, except that the 1 in the last equation is exchanged for -1. □

Proof (for Theorem 10). Exactly as the proof of Theorem 3. □

Proof (for Theorem 11). Let $B = \{g_{i_{|N|r+1-r}}, g_{i_{|N|r+2-r}}, \ldots, g_{i_{|N|r+1}}\}$. By the pigeonhole principle, at least one bundle A_j in i's MMS partition must contain at

Algorithm 5 Find a α-MMS solution to ordered chore instance

Input: An ordered instance $I = \langle N, M, V, C\rangle$ with all $v_{ij} \geq -1$, $v_i(M) > -|N|$ and $\mu_i \leq -1$

Output: Allocation A consisting of each bundle B allocated

while there is more than one agent left
    $B = \cup_{h=1}^{\ell} C_h^H$
    while $v_i(B) < -\alpha$ for all agents i
        if $B \cap C_h^H \neq \emptyset$ for some C_h
            $j =$ any element of $C_h^L \setminus B$
            $j' =$ any element of $B \cap C_h^H$
            $B = (B \setminus \{j'\}) \cup \{j\}$
        else $j =$ any $c_{\lceil |C_h|/n \rceil}$ in B for C_h with $|C_h|/n < \lceil |C_h|/n \rceil$
            $B = B \setminus \{j\}$
    allocate B to some agent i with $v_i(B) \geq -\alpha$
    remove B and i from I and update n, and C_h^H and C_h^L for all h
allocate the remaining chores to the last agent

least $r+1$ chores from $\{g_{i_1}, g_{i_2}, \ldots, g_{i_{|N|r+1}}\}$. Since B contains the $r+1$ least valuable goods in this set, we have $v_i(B) \geq v_i(A_j) \geq \mu_i$. □

F.1 Proof for Theorem 12

To prove Theorem 12, we will show that Algorithm 5, a variation of Algorithm 3, finds $(|N|/(2|N|-1))$-approximate MMS allocations for ordered instances where no chore is worth less than -1, $\mu_i \leq -1$ and $v_i(M) > -|N|$ for all $i \in N$. Algorithm 5 works in a similar manner to Algorithm 3, it starts by creating a bundle B consisting of the $\lceil |C_h|/n \rceil$ worst chores in each category C_h (denoted by C_h^H). It then gradually, as to not improve the value of the bundle by more than 1, improves the value of the bundle by exchanging a good in some C_h^H for one of the $\lceil |C_h|/n \rceil$ best chores in the same category (denoted by C_h^L). To mitigate the effects of rounding C_h/n, it can also remove the $\lceil |C_h|/n \rceil$ best chore in any C_h where $\lceil |C_h|/n \rceil > |C_h|/n$. This strategy guarantees, as for goods, that the bundle created is feasible and that there at any point remains at most as many chores as can be allocated to the remaining agents. Additionally, it makes sure that the bundle initially contains at least $1/n$ of the remaining disutility and in the end at most $1/n$ of the remaining disutility. Thus, a similar argument can be made about the upper bound on the remaining disutility, as for the lower bound on remaining value for goods.

Lemma 8. *Let $I = \langle N, M, V, C\rangle$ be an ordered instance of the fair allocation of chores problem under cardinality constraints where each chore is worth no less than -1 and $v_i(M) \geq -|N|$ for each $i \in N$. Let n denote the number of remaining agents at any point during the execution of Algorithm 5. Then for* *any* $\alpha \in (1, 2]$*, each remaining agent assigns a value of at least* $-|N| + (|N| - n)(\alpha - 1)$ *to the set of unallocated chores at any point during the execution of the algorithm.*

Proof. Since $v_i(M) \geq -|N|$, this holds at the start of the algorithm. Assume that there are n remaining agents at the start of an iteration and for each remaining agent i, $v_i(M) \geq -|N| + (|N| - n)(\alpha - 1)$. Let i' be the agent receiving B in the iteration. For any remaining agent $i \neq i'$, we wish to show that $v_i(M \setminus B) \geq -|N| + (|N| - n + 1)(\alpha - 1)$. Due to the additive valuations, the only way that $v_i(M \setminus B) < -|N| + (|N| - n + 1)(\alpha - 1)$ is if $v_i(B) > -\alpha + 1$. Since any change to B after the initial creation removes a chore from B or exchanges a chore in B for another, any individual change cannot increase the value of B by more than 1. Thus, the only way for $v_i(B) > -\alpha + 1$ is if $B = \bigcup_{h=1}^{\ell} C_h^H$ and $v_i(\bigcup_{h=1}^{\ell} C_h^H) > -\alpha + 1$. However, by definition $v_i(C_h^H) \leq v_i(C_h)/n$ which implies $v_i(B) \leq v_i(M)/n$. Consequently, $v_i(M \setminus B) \geq (n-1)v_i(B) > -(n-1)(\alpha - 1) \geq -|N| + |N|(\alpha - 1) - (n-1)(\alpha - 1) = -|N| + (|N| - n + 1)(\alpha - 1)$. □

With Lemma 8 we have a sufficient upper guarantee for the remaining disutility. We are now ready to show the guarantees of Algorithm 5.

Lemma 9. *Given a normalized ordered instance* $I = \langle N, M, V, C \rangle$ *of the fair allocation problem under cardinality constraints where* $\mu_i \leq -1$, $v_{ij} \geq -1$ *and* $v_i(M) > -|N|$ *for all* $i \in N$, $j \in M$*, and* $\alpha = (2|N| - 1)/|N|$*, Algorithm 5 finds a feasible* $(2|N| - 1)/|N|$*-approximate MMS allocation in polynomial time in the number of agents and chores.*

Proof. When allocating the remaining chores to the last agent, Lemma 8 guarantees that the chores are worth at least $-\alpha$, if $-|N| + (|N| - 1)(\alpha - 1) \geq -\alpha$, which holds for $\alpha \geq (2|N| - 1)/|N|$. Additionally, as long as B reaches a value of $-\alpha$ before running out of improvement operations, any other agent is also guaranteed to receive a bundle they value at no less than $-\alpha$. Since B contains the $\lfloor C_h/n \rfloor$ best chores in each category C_h when the algorithm runs out of operations, B will contain chores of no more than $1/n$ of the remaining disutility. We thus only need to show that the remaining value is always at least $-n\alpha$ for any remaining agent. Lemma 8 guarantees that the remaining value is at least $-|N| + (|N| - n)\alpha$. Since, this is at least $-\alpha$ for $n = |N| - 1$ for $\alpha \geq (2|N| - 1)/|N|$, the value is at least $-(n-1)(\alpha - 1) - \alpha \geq -n\alpha$ for any other n, and we are guaranteed that the value of B reaches at least $-\alpha$ in any iteration. Since $\mu_i \leq -1$ for $i \in N$, each agent i receives at least $-\alpha\mu_i$ value.

It remains to show that any bundle allocated is feasible. As long as $|C_h| \leq nk_h$, it holds that $\lceil |C_h|/n \rceil \leq k_h$ and any bundle allocated is feasible. Obviously, $|C_h| \leq nk_h$ holds when $n = |N|$, as all instances are assumed to have at least one feasible complete allocation. Assume that $|C_h| \leq nk_h$ holds at the start of an iteration. The bundle B contains at least $\lfloor |C_h|/n \rfloor \geq |C_h| - (n-1)k_h$ of the chores in C_h at any point during an iteration. Thus, $|C_h \setminus B| \leq (n-1)k_h$ and the condition holds for $n - 1$ after allocating B. Consequently, each allocated bundle, including the bundle allocated to the last agent, is feasible. Since the last agent receives all remaining chores, all chores are allocated.

In each iteration of the algorithm, chores are removed from B and exchanged through a set of operations. As each chore is not added back into B after being removed, the number of operations in each iteration is polynomial in the number of agents and chores. Since there are $|N|-1$ iterations, the running time of the algorithm is also polynomial in the number of agents and chores. □

We can now combine Lemma 3 with rescaling of valuations in order to show that $((2|N|-1)/|N|)$-approximate MMS allocations always exist and can be found in polynomial time.

Proof (for Theorem 12). First of all, if the instance I has any agent $i \in I$ with $v_i(M) = 0$, then we know we can remove i from the instance by allocating the k_h worst chores in each C_h to i. Thus, we can assume $v_i(M) < 0$.

The instance can by Theorem 10 easily be turned into an ordered instance. Further, the valuations of each agent i can be rescaled so that $v_i(M) = -|N|$, which by Theorem 9 gurantees that $\mu_i \leq -1$. Then, if $v_i(1) < -1$, then Theorem 11 allows us to rescale i's valuations so that $v_i(1) = -1$, while maintaining that $v_i(M) \geq -|N|$ and $\mu_i \leq -1$. Consequently, I can be turned into an instance accepted by Algorithm 5 in polynomial time, and Lemma 9 gurantees that a $((2|N|-1)/|N|)$-approximate MMS allocation can be found in polynomial time. □

F.2 Proof for Theorem 13

To prove Theorem 13, we need to develop a result similar to Lemma 4 and show that a similar algorithm to Algorithm 4 can in polynomial time find 3/2-approximate MMS allocations for a restricted class of instances. To simplify notation, we assume that in any ordered instance, the chores are ordered so that the chore numbered 1 is the worst chore (provides most disutility) and the chore numbered $|M|$ is the best chore (provides least disutility).

Lemma 10. *Let $I = \langle N, M, V, \langle (C_1, k_1) \rangle \rangle$ be an ordered instance of the the fair allocation of chores problem under cardinality constraints. Let $B_r = \{1, 2, \ldots, r\}$ along with the $\max(0, |M| - (|N|-r)k_1 - r)$ best chores in M, for any $r \in \{1, 2, \ldots, |N|\}$. Then, for any $i \in N$,*

$$v_i(B_r) \geq r\mu_i$$

Proof. For any agent $i \in N$ and $r \in \{1, 2, \ldots, |N|\}$, let A be an MMS partition of I for i. Let A^* be the union of r bundles in A such that $\{1, 2, \ldots, r\} \subseteq A^*$. At least one such A^* must exist as the r chores in $\{1, 2, \ldots, r\}$ are contained in at most r distinct bundles in A. Since A is an MMS partition, we know that $v_i(A^*) \geq r\mu_i$. Since A is feasible, A^* must contain at least $\max(r, |M| - (|N|-r)k_1)$ chores. Otherwise, there are more chores left in M than can be contained in the $|N|-r$ bundles of A not included in A^*. Since B_r contains the chores $\{1, 2, \ldots, r\}$ along

Algorithm 6 Find (3/2)-MMS solution for single-category instance

Input: An ordered instance $I = \langle N, M, V, \langle C_1, k_1 \rangle \rangle$ with $|M| > |N|$, $\mu_i \leq -1$, $v_i(B_r) \geq -r$ (from Lemma 10), $v_i(1) > -1$, and $v_i(|N|+1) > -1/2$ for every $i \in N$, $r \in \{1, 2, \ldots, |N|\}$

Output: Allocation A consisting of each bundle B'_j allocated

let $n = |N|$ let $B'_1 = \{1\}, B'_2 = \{2\}, \ldots, B'_n = \{n\}$
for $j = 1$ up to n
  if $|M| > |N|$
    add the $\min(|M| - |N|, k_1 - 1)$ worst chores in $M \setminus (B'_j \cup B'_{j+1} \cup \cdots \cup B'_n)$ to B'_j
  while $v_i(B'_j) < -3/2$ for all $i \in N$ and $M \setminus (B'_j \cup B'_{j+1} \cup \cdots \cup B'_n)$ contains a better chore than the worst chore in $B'_j \setminus \{j\}$
    exchange the worst chore $g \in B'_j \setminus \{j\}$ for the worst chore $g' \in M \setminus (B'_j \cup B'_{j+1} \cup \cdots \cup B'_n)$ with $g < g'$
  while $v_i(B'_j) < -3/2$ for all $i \in N$
    remove the worst chore $g \in B'_j \setminus \{j\}$ from B'_j
  find $i \in N$ such that $v_i(B'_j) \geq -3/2$
  allocate B'_j to i and set $N = N \setminus \{i\}$, $M = M \setminus B'_j$.

with $\max(0, |M| - (|N| - r)k_1 - r)$ other chores, it follows that $|B_r| \leq |A^*|$. Combined with the fact that the chores in $B_r \setminus \{1, 2, \ldots, r\}$ are the chores in $M \setminus \{1, 2, \ldots, r\}$ that provide the least disutility, we get that

$$\begin{aligned} v_i(B_r) &= v_i(\{1, 2, \ldots, r\}) + v_i(B_r \setminus \{1, 2, \ldots, r\}) \\ &\geq v_i(\{1, 2, \ldots, r\}) + v_i(A^* \setminus \{1, 2, \ldots, r\}) \\ &= v_i(A^*) \\ &\geq r\mu_i \end{aligned}$$

□

As was the case for goods, Lemma 10 allows us to, in polynomial time, scale valuations such that our estimates of μ_i achieves the required accuracy for the bag-filling style algorithm (Algorithm 6). Additionally, it provides a vital role in showing that for some agent $i \in N$, the bundle created in Algorithm 6 contains a sufficient number of chores when it is worth more than $-3/2$.

With Lemma 10 we can now prove that Algorithm 6 finds a (3/2)-approximate MMS allocation for the instances that fulfills the input requirements.

Lemma 11. *For an instance I of the fair allocation of chores problem under cardinality constraints satisfying the requirements of Algorithm 6, Algorithm 6 finds a* (3/2)*-approximate MMS allocation in polynomial time.*

Proof. To show that the algorithm finds a (3/2)-approximate MMS allocation in polynomial time, we wish to show that (i) any B'_j allocated is feasible, (ii) all chores are allocated, (iii) there is always an agent $i \in N$ with $v_i(B'_j) \geq -3/2$, and (iv) the algorithm finishes in polynomial time.

Feasible bundles

To show (i) it suffice to show that $|B'_j| \leq k_1$ at any point during the algorithm. Initially this holds, as the instance I is assumed to have at least one feasible allocation and each chore must be allocated, hence $k_1 > 1$. Further, on line 4, at most $k_1 - 1$ chores are added to B'_j. Since $|B'_j| = 1$ before this line, the size of B'_j remains at most k_1. In the first loop (line 5), the size of B'_j does not change, as each iteration exchanges one good in B'_j for one outside of B'_j. In the second loop (line 7), chores are removed from B'_j and the size of B'_j decreases. Hence, $|B'_j| \leq k_1$, which guarantees that B'_j is feasible when allocated and (i) holds.

At the end of iteration j, $\mathbf{v_i(B'_j) \geq -3/2}$ for an $\mathbf{i \in N}$

As $v_i(g) > -1$ for all $i \in N$ and $g \in M$, we know that $v_i(\{j\}) \geq -1$. Since the second loop (line 7) removes one and one chore from B'_j, except for j, B'_j must at some point be worth more than $-3/2$ to some agent $i \in N$. Otherwise, B'_j would become $\{j\}$, which is worth at least $-1 > -3/2$ to all agents in N. Consequently, every bundle allocated is worth no less than $-3/2$ to the agent receiving it and a bundle is allocated in every iteration.

All chores are allocated

Showing that all of the chores are allocated boils down to showing that B'_j contains a sufficiently large number of chores when allocated. To show that B'_j has sufficient size, we will use a similar loop invariant argument as used to show that a sufficient amount of value remained for single-category instances of goods. Let B^j_r denote the collection consisting of $\{j, j+1, \ldots, r\}$ and the $\max(0, |M| - (n-r)k_1 - (r-j+1))$ best chores in M at the start of iteration j. In other words, B^j_r contains $\{j, j+1, \ldots, r\}$ and if $|M \setminus \{j, j+1, \ldots, r\}| > (n-r)k_1$, B^j_r contains the exact number of chores needed so that $|M \setminus B^j_r| \leq (n-r)k_1$. These additional chores are the best (least disutility) remaining chores. Note that if $|M| \leq k_1|N|$ after iteration $j-1$, then by definition $|B^j_r| \leq (r-j+1)k_1$.

We wish to show that the two following properties hold for B^j_r for all $r \in \{1, 2, \ldots, n\}$ and $j \leq n$:

1. $v_i(B^j_r) \geq -(r-j+1)$ for all $i \in N$
2. $|B^j_r| \leq k_1(r-j+1)$

Specifically, this would mean that when $r = j$, then $v_i(B^j_r) \geq -1$ for all $i \in N$ and $|M \setminus B^j_r| \leq |N \setminus \{i\}|k_1$. In other words, the bundle B^j_r is such that if B^j_r is allocated, then after allocation there remains at most as many chores as can be given to the remaining agents and the bundle may be given to any one of the agents without violating the MMS approximation guarantee. Especially, when $j = n$, this would mean that all remaining chores can and will be allocated to the remaining agent (line 4 will for $j = n$ add all chores in M to B'_j if $|M| \leq k_1$).

Notice how $B^j_r = B_r$ when $j = 1$. Consequently, by Lemma 10 we have $v_i(B^1_r) \geq -r$. Additionally, since $|M| \leq nk_1$ at the start,

$$|B^1_r| = r + \max(|M| - (n-r)k_1 - r, 0) \leq r + nk_1 - (n-r)k_1 - r = rk_1$$

Thus, the conditions hold for $j = 1$. Assume for some $j < n$ that they hold for all $r \geq j$. We wish to show that they hold for $j' = j+1$ and all $r \geq j'$. For 2., notice that if B'_j is not modified in the second loop (line 7), then either $|B'_j| = |M| - |N| + 1$ and B'_j contains all chores in $M \setminus \{j+1, j+2, \dots, n\}$, or $|B'_j| = k_1$. In the first case, $|M \setminus B'_j| = |\{j', j'+1, \dots, r\}| = r - j' + 1 \leq k_1 |N \setminus \{i\}|$. In the second case, since $|B^j_j| \leq k_1$, $|M| \leq |N| k_1$ and $|M \setminus B'_j| \leq |N \setminus \{i\}| k_1$. Consequently, we know $|B^{j'}_r| \leq k_1(r - j' + 1)$ in either case.

If B'_j is modified in the second loop, then note that before the first iteration of the loop, B'_j consists of j and the $\min(|M| - |N|, k_1 - 1)$ best chores in M. This is the exact same construction as B^j_j, except that B^j_j contains $|B^j_j| - 1 \leq \min(|M| - |N|, k_1 - 1)$ of the best chores in M. Since the loop removes the worst chore in $B'_j \setminus \{j\}$ in each iteration, B'_j will turn into B^j_j at some point. Since $v_i(B^j_j) \geq -1$, $|B'_j| \geq |B^j_j|$ when the second loop finishes. By definition, $|M \setminus B'_j| \leq |M \setminus B^j_j| \leq k_1(n - j) = k_1(|N \setminus \{i\})$ and 2. holds in all cases.

For 1. first note that any change performed in the first or the second loop (lines 5 and 7) either removes a chore in $M \setminus \{1, 2, \dots, n\}$ from B'_j or exchanges a chore from that subset of M for another in the same subset. That is, the value of B'_j changes by at most $1/2$ in each operation. Thus, either $v_{i'}(B'_j) < -1$ for all $i' \in N$ or B'_j is not modified in either loop.

If B'_j is modified in the last loop (line 7), then since $B^j_r \setminus \{j, j+1, \dots, r\}$ and $B'_j \setminus \{j\}$ both consist of some number of the best chores in M, either $B'_j \subseteq B^j_r$ or $(B^j_r \setminus \{j+1, j+2, \dots, r\}) \subseteq B'_j$. In the first case,

$$v_{i'}(B^{j'}_r) = v_{i'}(B^j_r) - v_{i'}(B'_j) \geq -(r - j + 1) - (-1) = -(r - j' + 1)$$

for all $i' \in N \setminus \{i\}$. In the second case, $B^{j'}_r = \{j', j'+1, \dots, r\}$ and $v_{i'}(B^{j'}_r) \geq -(r - j' + 1)$ since $v_{i'}(g) \geq -1$ for $g \in M$.

If B'_j is not modified in the second loop, then we know that either $|B'_j| < k_1$ and $M \setminus B'_j = \{j', j'+1, \dots n\}$ or $|B'_j| = k_1$. In the first case, we have as earlier $B^{j'}_r = \{j', j'+1, \dots r\}$ and $v_i(B^{j'}_r) \geq -(r - j' + 1)$. In the second case, if $|B^j_r| < k_1 - 1 + (r - j + 1)$, then $B^{j'}_r = \{j', j'+1, \dots, n\}$ and $v_{i'}(B^{j'}_r) \geq -(r - j' + 1)$. Otherwise, we know that by the way the chores are exchanged in the loop, we always select the worst $g' \in M \setminus (B'_j \cup B'_{j+1} \cup \dots \cup B'_n)$ that is better than the chore replaced. In other words, there is no chore in $M \setminus B'_j$ that is both better than and worse than two distinct chores in $B'_j \setminus \{j\}$. Thus, either $B'_j \subseteq B^j_r$ or there is no chore $g \in B'_j$ such that there is $g' \in B^j_r \setminus \{j, j+1, \dots, r\}$ with $g' < g$. In other words, $B'_j \setminus \{j\}$ contains $k_1 - 1$ chores such that for $g \in B'_j \setminus \{j\}$, g is either worse than the chores in $B_r^j \setminus \{j, j+1, \ldots, r\}$ or B'_j contains all worse chores in $B_r^j \setminus \{j, j+1, \ldots, r\}$. Consequently, since $|B_r^j| \leq k_1(r-j+1)$, the k_1 chores removed from B_r^j to create $B_r^{j'}$ are j and the $k_1 - 1$ worst chores in $B_r^j \setminus \{j, j+1, \ldots, r\}$. Due to the ordered instance, these k_1 chores must be at least $1/(r-j+1)$ of the disutility in B_r^j. Consequently,

$$v_{i'}(B_r^{j'}) \geq \left(1 - \frac{1}{r-j+1}\right) v_{i'}(B_r^j) \geq \frac{r-j}{r-j+1} \cdot -(r-j+1) = -(r-j'+1)$$

for all $i' \in N \setminus \{i\}$. Consequently, 1. holds and it follows that (ii) holds.

Polynomial run time

It remains to show that (iv) holds, namely that the algorithm run in polynomial time. Since (i), (ii) and (iii) hold, it follows that the algorithm will not be stuck in any loop without any operations to perform. Further, since j is bounded in the number of agents, it suffice to show that each iteration of the outer loop can be performed in polynomial time. Since each iteration of the first loop improves the worst chore in B'_j, the number of iterations of this loop is at most $|M|$. In the second loop, a chore is removed from B'_j in each iteration. Consequently, the loop can at most have $|B'_j \setminus \{j\}| \leq |M|$ iterations. Since each of the individual operations can be performed in polynomial time, it therefore follows that each iteration of the outer loop can be performed in polynomial time and (iv) holds. □

Proof (for Theorem 13). To show that 3/2-approximate MMS allocations can be found in polynomial time, we will show that in polynomial time, I can either be converted into an instance that Algorithm 6 accepts or a 3/2-approximate MMS allocation can trivially be found. First, note that if $v_i(M) = 0$, for some agent $i \in N$, then agent i can be allocated the worst k_1 chores, as agent i assigns each of these a value of 0. The instance without i and these chores is obviously feasible and one in which the MMS of each agent is no worse than in I. Thus, any i with $v_i(M) = 0$ can be reduce away, and we can assume that $v_i(M) < 0$ for each remaining agent $i \in N$. If $|M| \leq |N|$, then any allocation that gives each agent at most one chore is an MMS allocation. Such an allocation can trivially be found in linear time.

If $|M| > |N|$, the only change needed for I is to rescale the valuations of the agents. First, by Theorem 9, we can for each agent $i \in N$ rescale i's valuations so that $v_i(M) = |N|$ which guarantees $\mu_i \leq -1$. Further, by Theorem 11 if $v_i(1) < -1$, then adjusting i's valuations so that $v_i(1) = -1$ maintains $\mu_i \leq -1$. If $v_i(|N|+1) < -1/2$, Theorem 11 also guarantees that $\mu_i \leq -1$ if we rescale i's valuations so that $v_i(|N|+1) = -1/2$. Similarly, by Lemma 10 if $v_i(B_r) < -r$ for some $r \in \{1, 2, \ldots, |N|\}$, then i's valuations can be rescaled so that $v_i(B_r) = -r$ while still guaranteeing $\mu_i \leq -1$. Since each rescale increases $v_i(M)$ and all the properties required for Algorithm 6 require the value of a set of chores to be above a certain threshold, each further rescale does not break any one of the properties that already hold. Thus, after checking all the cases above, all conditions of Algorithm 6 hold. Since both r and i are bound in the number of agents, the conversion can be performed in polynomial time. □

G Omitted Examples

Example 1 (Failing valid reduction with two goods). As mentioned in Section 4, even for instances with a single category, certain types of valid reductions from unconstrained fair allocation fail to be applicable. One of these is the valid reduction created by constructing a bundle B consisting of two goods such that $v_i(B) \geq \alpha\mu_i$ for an agent i and $v_{i'} \leq \mu_{i'}$ for all other agents i'. In unconstrained fair allocation, this is a valid reduction since for any agent i', one can easily show that their MMS remains at least as high. This follows from the fact that the goods in B are contained in either one or two bundle in the agent's MMS partition. Therefore, there is either already $n-1$ bundles valued at $\mu_{i'}$ or higher, or the two bundles containing goods from B can be combined to have a value of at least $\mu_{i'}$. Under cardinality constraints, this last step could lead to infeasibility, and the reduction therefore does not work, as seen below.

Let I be an instance of the fair allocation problem under cardinality constraints, with a single category C_1 with threshold $k_1 = 5$, 11 goods and 3 agents with identical valuation functions given in Table 1. The MMS of each agent is 1, as $v_i(M) = 3$ and the partition $(\langle 1, 8, 9\rangle, \langle 2, 10, 11\rangle, \langle 3, 4, 5, 6, 7\rangle)$ contains three bundles with a value of exactly 1. If $\alpha \leq 19/20$, then the bundle $B = \langle 2, 7\rangle$ would constitute a valid reduction for any agent in the unconstrained setting. However, with the given threshold for C_1, removing B and an agent i would leave us with an instance where the MMS of the agents would be $37/40$ with the MMS partition $(\langle 1, 8, 9, 10\rangle, \langle 3, 4, 5, 6, 11\rangle)$. If $37/40 < \alpha \leq 19/20$, this would even make it impossible to achieve an α-approximate MMS allocation.

j	1	2	3	4	5	6	7	8	9	10	11
v_i	$\frac{3}{4}$	$\frac{3}{4}$	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$

Table 1. Valuations in Example 1