Indivisible Mixed Manna: On the Computability of MMS+PO Allocations

RUCHA KULKARNI, University of Illinois at Urbana-Champaign, USA
RUTA MEHTA, University of Illinois at Urbana-Champaign, USA
SETAREH TAKI, University of Illinois at Urbana-Champaign, USA

In this paper we initiate the study of finding *fair* and *efficient* allocations of an indivisible mixed manna: Divide m indivisible items among n agents under the popular fairness notion of *maximin share* (MMS) and the efficiency notion of *Pareto optimality* (PO). A mixed manna allows an item to be a good for some agents and a chore for others, and hence strictly generalizes the well-studied goods (chores) only manna. For the goods manna, non-existence of an MMS allocation prompted a series of works on finding approximate MMS allocations, and the best factor known to date is $\alpha =\sim 3/4$, while non-existence is only known for α close to 1. The problem of finding α-MMS allocation for the *(near) best* $\alpha \in (0,1]$ for which it exists, remains unresolved even when the number of agents is a constant, while the problem of finding α-MMS + PO allocation is unexplored for *any* $\alpha \in (0,1]$.

We make significant progress on the above questions for the case of mixed manna. First, we show that for any $\alpha > 0$, an α-MMS allocation may not always exist, thus ruling out solving the problem for a fixed α. Second, towards computing α-MMS + PO allocation for the best possible α, we obtain a dichotomous result: We derive two conditions and show that the problem is tractable under these two conditions, while dropping either renders the problem intractable. The two conditions are: (i) number of agents is a constant, and (ii) for every agent, her absolute value for all the items is at least a constant factor of her total (absolute) value for all the goods *or* all the chores.

In particular, first, for instances satisfying (i) and (ii) we design a PTAS – an efficient algorithm to find an $(\alpha - \epsilon)$-MMS and γ-PO allocation when given $\epsilon, \gamma > 0$, for the highest possible $\alpha \in (0,1]$. Second, we show that if either condition is not satisfied then finding an α-MMS allocation for *any* $\alpha \in (0,1]$ is NP-hard, even when a solution exists for $\alpha = 1$. On m item instances our PTAS runs in time $2^{O(1/\min\{\epsilon^2,\gamma^2\})} poly(\text{m})$ for given ϵ and γ, and therefore gives polynomial run-time for ϵ, γ as small as $O(1/\sqrt{\log m})$.

As corollaries, our algorithm resolves the open questions of designing a PTAS for a goods manna and a chores manna with constantly many agents to find an α-MMS allocation for the best possible α; the best known was $\alpha = \sim 3/4$ for goods manna, and $\alpha = 9/11$ for chores manna. To the best of our knowledge, ours is the first algorithm that ensures both approximate MMS and PO guarantees. In terms of techniques, for the first time, we use an LP-rounding through envy-cycle elimination as a tool to solve an MMS problem and ensure PO, which may be of independent interest.

1 INTRODUCTION

Finding fair and economically efficient allocations of indivisible items is a fundamental problem that arises naturally in various multi-agent systems [Ste48, BT96, Vos02, Mou04, EPT07, Bud11, GHS+18], for example, school seats assignment, spectrum allocation, air traffic management, allocating computing resources on a network, splitting assets and liabilities in partnership dissolution, and office tasks. Many of these involve both goods that are freely disposable and chores that *have* to be assigned. In this paper we study the problem of finding fair and efficient allocations of a *mixed manna*, i.e., a set $\mathcal{M}$ of discrete items that are goods/chores, among a set $\mathcal{N}$ of agents with additive valuations. We note that a mixed manna allows an item to be a good (positively valued) for some agents, and a chore (negatively valued) for others, and thereby strictly generalizes the extensively studied goods (chores) manna (See Appendix E for a detailed discussion on related works).

To measure *fairness* and *efficiency* we consider the popular and well studied notions of maximin-share (MMS) (*e.g.,* see [Bud11, KPW18, AMNS17, GHS+18, FGH+19, GT20]) and Pareto optimality (PO) respectively. Pareto optimality is a sought after notion in economics, and when achieved means that there is no other allocation that makes all the agents better off and at least one of them strictly better off. The fairness notion of maximin-share is inspired from the classical cut-and-choose mechanism[1]. The MMS value of agent i is the value that she can guarantee herself if she is to partition (cut) $\mathcal{M}$ into $n = |\mathcal{N}|$ bundles, given that she is the last agent to choose her favorite bundle. Naturally, she will try to maximize the minimum valued bundle in the partition. Formally, if $\Pi_n(\mathcal{M})$ represents all possible partitions $(A_1, \dots, A_n)$ of $\mathcal{M}$ into n bundles, and v_i is her valuation function, then

$$\mathsf{MMS}_i(\mathcal{M}) = \max_{(A_1,\dots,A_n)\in\Pi_n(\mathcal{M})} \min_{k\in[n]} v_i(A_k) \ . \tag{1}$$

An MMS *allocation* is one where every agent gets at least her MMS value. The problem of finding an MMS allocation has seen extensive work in the case of a goods (chores) only manna, while no results are known for the mixed manna. Even for the goods (chores) manna, no work has explored the PO guarantee in addition to MMS, to the best of our knowledge; finding fair+(approximate) PO allocations has been studied for other fairness notions like EF1 and Prop1 [BKV18, ZP20, AMS20]. In this paper we initiate the study of finding an MMS + PO allocation for a mixed manna.

For the goods manna, the notable result of Kurokawa, Procaccia and Wang [KPW18] showed that an MMS allocation may not always exist, but α-MMS allocations, where every agent gets at least α times her MMS value, exist for $\alpha = 2/3$. This prompted works on efficient computation of an α-MMS allocation for progressively better $\alpha \in [0, 1]$ [AMNS17, BKM17, GMT18, GHS+18]; the best factor known so far is $\alpha = (3/4 + 1/(12n))$ by Garg and Taki [GT20] for $n \geq 5$ agents. With a chores manna, MMS values are negative, and an α-MMS allocation gives each agent i a bundle of value at least $\frac{1}{\alpha} \cdot \mathsf{MMS}_i$. For this case too, starting from the work [ARSW17] for $\alpha = 1/2$, a series of works improved it to $9/11$ [BKM17, HL19].

With a mixed manna we show that, for any fixed $\alpha \in (0, 1]$, an α-MMS allocation may not always exist (see Appendix B); in contrast, non-existence with a goods manna is known for α close to one [KPW18]. This rules out efficient computation for any fixed α, and naturally raises the following problem.

*Design an efficient algorithm to find an α-*MMS + PO *allocation for the best possible α, i.e., the maximum $\alpha \in (0, 1]$ for which it exists.*

This *exact* problem is intractable: In the case of identical agents, an $(\alpha = 1)$-MMS allocation exists by definition. However, finding one is known to be NP-hard for a goods manna.[2] On the positive side, a polynomial-time approximation scheme (PTAS) is known for this case due to [Woe97]; given a *constant* $\epsilon \in (0, 1]$, the algorithm finds a $(1 - \epsilon)$-MMS allocation in polynomial time. No such result is known when the agents are not identical. Guaranteeing PO in addition adds to the complexity, since even checking if a given allocation is PO is coNP-hard even with two identical agents [ABL+16]. In light of these results, we ask,

Question. *Can we design a* PTAS, *namely an efficient algorithm to find an $(\alpha - \epsilon)$-*MMS + γ-PO *allocation, given $\epsilon, \gamma > 0$, for the best possible α?*

[1]In case of divisible items and two agents, one agent cuts so that she is okay with both the bundles and the other person chooses (mentioned in the Bible).

[2]Checking if a given instance admits an MMS allocation is known to be in $\mathsf{NP}^{\mathsf{NP}}$, but not known to be in NP [BL16].

Our Contribution. In this paper we make significant progress towards this question for mixed manna by showing the following dichotomy result: We derive two conditions and show that the problem is tractable under these conditions, while dropping either renders the problem intractable. The two conditions are: (i) number of agents n is a constant, and (ii) for every agent i, her total (absolute) value for all the items ($|v_i(\mathcal{M})|$) is significantly greater than the minimum of her total value of goods (v_i^+) and her total (absolute) value for chores (v_i^-), i.e., for a constant $\tau > 0$, $|v_i(\mathcal{M})| \geq \tau \cdot \min\{v_i^+, v_i^-\}$.

In particular, first, for instances satisfying (i) and (ii), we design a PTAS (as asked in the above question). Second, we show that if either condition is not satisfied, then finding an α-MMS allocation for *any* $\alpha \in (0, 1]$ is NP-hard, even with *identical agents* where a solution exists for $\alpha = 1$. This hardness is striking because it shows inapproximability within *any* non-trivial factor when either (i) or (ii) is not satisfied. This also indicates that the two conditions are unavoidable.

Our algorithm, in principle, gives a little more than a PTAS. It runs in time $2^{O(1/\min\{\epsilon^2,\gamma^2\})}poly(\mathrm{m})$ for given ϵ, γ, thus gives polynomial run-time for ϵ, γ as small as $O(1/\sqrt{\log m})$, where $m = |\mathcal{M}|$.

α-MMS + PO for goods (chores) manna. As a corollary, we obtain a PTAS for finding α-MMS + PO allocations of a goods manna and a chores manna when the number of agents is a constant. This improves the previous results for these settings in two aspects: (i) provides the best possible approximation factor; factors better than the general case known for good manna are $4/5$ for $n = 4$ [GHS+18], $8/9$ for $n = 3$ [GM19], and 1 for $n = 2$ [BL16], and (ii) provides an additional (approximate) PO guarantee.

Challenges. The key challenge in solving this question is handling items of high value to any agent. In the goods or chores mannas, these items can be greedily assigned, for example as singleton bundles. But in a mixed manna, *high valued* goods (chores) may have to be bundled with specific sets of chores (goods) or low valued items to form lesser valued bundles. Secondly, the MMS values of the agents, and the α for which α-MMS allocation exist, both are not known. In fact, computing the exact MMS values is NP-hard (even with a goods manna).

PTAS to find MMS values. As the first key step for our main algorithm, we design a PTAS that returns $(1-\epsilon)$ approximate MMS values of agents, which may be of independent interest.

A new technique to prove PO. Since certifying a PO allocation is a coNP-hard problem [ABL+16], known works maintain a PO allocation with market equilibrium as a certificate [BKV18, MG20, GM20]. We develop a novel approach to ensure PO with α-MMS through LP rounding. The LP itself is intuitive, however the rounding is involved. It makes use of *envy-graph* and properties of the MMS in a novel way. This approach may be of independent interest.

Organization. Section 2 gives a formal definition of the problem and notations. Section 3 discusses the main result of PTAS for the α-MMS + PO problem with the best possible α; the formal proofs missing from this section due to space limitations are in Appendix A. The formal and complete discussion of the PTAS for computing MMS values for the case when MMS ≥ 0 is in Section 4 and for the MMS < 0 case is in Appendix C. Appendix B discusses the non-existence of α-MMS allocation for any $\alpha \in (0, 1]$. Finally, the discussion of NP-hardness results is in Appendix D.

2 PROBLEM DEFINITION AND NOTATIONS

Notations. We use $[k]$ to denote the set $\{1, 2 \cdots, k\}$. For $c \in \mathbb{R}$, c^+ denotes $\max\{c, 0\}$.

We consider the problem of allocating a set $\mathcal{M}$ of m indivisible items among a set $\mathcal{N}$ of n agents in a *fair* and *efficient* manner, with the fairness notion of *maximin share* (MMS) and the efficiency notion of *Pareto-optimality* (PO). Each agent $i \in \mathcal{N}$ has an additive valuation function $v_i : 2^{\mathcal{M}} \to \mathbb{R}$ over sets of items. For a set $S \subseteq \mathcal{M}$, her value is $v_i(S) = \sum_{j \in S} v_{ij}$. Agents are called *identical* if their v_is are the same function; in this case, the valuation function is denoted by v.

The set of items valued non-negatively (negatively) by an agent i are called her *Goods* (*Chores*), and denoted by $\mathcal{M}_i^+ = \{j \mid v_{ij} \geq 0\}$ ($\mathcal{M}_i^- = \{j \mid v_{ij} < 0\}$). The sets of all the goods and all the chores of the instance are defined as respectively $\mathcal{M}^+ := \cup_i \mathcal{M}_i^+$, and $\mathcal{M}^- := \mathcal{M} \backslash \mathcal{M}^+$. We refer to an item j as a *good* if $v_{ij} \geq 0$ for *some* agent and as a *chore* if $v_{ij} < 0$ for all agents.

MMS **values and allocation.** Let $A^\pi = \{A_1, A_2, \cdots, A_n\}$ denote a partition of all the items among the n agents, referred as an *allocation*, i.e., $A_i \cap A_{i'} = \emptyset$ for all distinct i, i' in $\mathcal{N}$, and $\cup_i A_i = \mathcal{M}$. And let $\Pi_n(\mathcal{M})$ be the set of all possible allocations of $\mathcal{M}$ among n agents. The maximin share (MMS) value of an agent i is defined as

$$\mathsf{MMS}_i^n(\mathcal{M}) = \max_{(A_1,\ldots,A_n) \in \Pi_n(\mathcal{M})} \min_{k \in [n]} v_i(A_k).$$

We refer to $\mathsf{MMS}_i^n(\mathcal{M})$ by MMS_i when the qualifiers n and $\mathcal{M}$ are clear, and by MMS when agents are identical. Note that MMS_i can be negative too.

Definition 2.1 (α-MMS allocation). *A^π is called an α-MMS allocation for an $\alpha \in (0, 1]$, if for each agent $i \in \mathcal{N}$ we have $v_i(A_i) \geq \alpha \mathsf{MMS}_i$ if $\mathsf{MMS}_i \geq 0$, $v_i(A_i) \geq (1/\alpha)\mathsf{MMS}_i$, if $\mathsf{MMS}_i < 0$. Equivalently, $v_i(A_i) \geq \min\{\alpha \mathsf{MMS}_i, (1/\alpha)\mathsf{MMS}_i\}$. When $\alpha \leq 0$, for simplicity, we define any allocation as α-MMS.*

γ-Pareto optimal (γ-PO) **and γ-Pareto dominating allocations.** An allocation A^π is said to be γ-PO if there does not exist any $B^\pi \in \Pi_n(\mathcal{M})$, called an allocation γ-Pareto dominating A^π, such that $\forall i \in \mathcal{N},\ v_i(B_i) \geq (1+\gamma)v_i(A_i)$ if $v_i(A_i) \geq 0$, and $v_i(B_i) \geq \frac{1}{(1+\gamma)} v_i(A_i)$ if $v_i(A_i) < 0$, and for at least one i the inequality is strict.

An allocation is called PO if it is 0-PO. It is easy to see that if there exists an α-MMS allocation for a given instance then there is one that is both α-MMS and PO (and thereby also γ-PO). This is because if an allocation B^π Pareto dominates an α-MMS allocation A^π, then B^π is also α-MMS.

Since the problem of finding α-MMS allocation is NP-hard for any $\alpha \in (0, 1]$, we design a PTAS to compute an α-MMS + PO allocation for a sub-class of instances. To characterize this sub-class, we will need the following definition.

$$\text{For each agent } i \in \mathcal{N}, \text{ define } \quad v_i^+ = \textstyle\sum_{j \in \mathcal{M}_i^+} v_{ij} \quad \text{and} \quad v_i^- = \textstyle\sum_{j \in \mathcal{M}_i^-} |v_{ij}|. \tag{2}$$

Definition 2.2 (α-MMS + PO Problem). *Given an instance $(\mathcal{N}, \mathcal{M}, (v_i)_{i \in \mathcal{N}})$ and $\alpha \in (0, 1]$ where,*

(1) the number of agents n is constant, and

(2) for some constant $\tau > 0$, for every agent $i \in \mathcal{N}$, $|v_i(\mathcal{M})| \geq \tau \cdot \min\{v_i^+, v_i^-\}$,

either find an allocation $A^\pi \in \Pi_n(\mathcal{M})$ that is both α-MMS and PO, also called an α-MMS + PO allocation, or correctly report that such an allocation does not exist for the given instance.

The above problem without the PO guarantee is called the α-MMS problem. Unlike the goods manna or the chores manna, for the mixed manna an α-MMS allocation may not exist for *any* $\alpha > 0$, as shown in Appendix B. Therefore, for a mixed manna, we can only hope to find an α-MMS allocation for the maximum possible α value for the given instance, formally defined below.

Definition 2.3 (OPT-α-MMS+PO Problem.). *Given an instance $(\mathcal{N}, \mathcal{M}, (v_i)_{i \in \mathcal{N}})$, find an allocation which is α-MMS + PO for an $\alpha \in (0, 1]$ such that there is no α'-MMS allocation for any $\alpha' > \alpha$.*

Note that, given an algorithm for the α-MMS + PO problem it is easy to solve OPT-α-MMS + PO by doing a binary search on the value of α. We design a PTAS for the former in Section 3 and thereby solve the latter efficiently up to a small error. By PTAS we mean, given constants $\epsilon, \gamma > 0$, in polynomial-time it either returns an $(\alpha - \epsilon)^{+}$-MMS + γ-PO allocation, or correctly reports that no α-MMS allocation exists.

The following observations will be useful in what follows (See Appendix A for proofs).

Lemma 2.1. $v_i(\mathcal{M}) \geq 0$ *iff* $\mathsf{MMS}_i \geq 0$.

Lemma 2.2. $\mathsf{MMS}_i \leq v_i(\mathcal{M})/|\mathcal{N}|$ *for all* $i \in \mathcal{N}$.

Lemma 2.3. *[Scale Invariance]* α-MMS + PO *allocations for the instances* $(\mathcal{N}, \mathcal{M}, (v_i)_{i \in \mathcal{N}})$ *and* $(\mathcal{N}, \mathcal{M}, (v'_i)_{i \in \mathcal{N}})$ *are the same when for all* i, j, $v'_{ij} = c_i \cdot v_{ij}$ *for some constants* $c_i > 0$.

3 PTAS FOR α-MMS + PO WITH NON-IDENTICAL AGENTS

In this section, we present our main result, namely a PTAS for the OPT-α-MMS + PO problem.

For OPT-α-MMS + PO the crucial step is to get a PTAS for the α-MMS + PO problem, discussed next. Recall that the definition of the latter, namely Definition 2.2, assumes two conditions on the input instance $(\mathcal{N}, \mathcal{M}, (v_i)_{i \in \mathcal{N}})$: (i) number of agents is a constant, and (ii) for each $i \in \mathcal{N}$, $|v_i(\mathcal{M})| \geq \tau \cdot \min\{v_i^+, v_i^-\}$, where $\tau > 0$ is a constant. Let us first briefly discuss why both of these conditions are unavoidable.

Hardness of approximation. In Appendix D, we show the following theorem by proving that if either condition is dropped then the problem is intractable for *any* $\alpha \in (0, 1]$, even when exact MMS allocation exists.

Theorem 3.1. *For any instance* $(n, \mathcal{M}, v)$ *with identical agents and* $v(\mathcal{M}) > 0$ *such that exactly one of the following two holds: (a) either* $n = 2$ *or (b)* $|v(\mathcal{M})| \geq \tau \cdot \min\{v(\mathcal{M}^+), |v(\mathcal{M}^-)|\}$ *for a constant* τ*, finding an* α-MMS *allocation of* $(n, \mathcal{M}, v)$ *for any* $\alpha \in (0, 1]$ *is* NP*-hard.*

To prove the above theorem, we design two reductions from a well-known NP-hard problem PARTITION to the problem of finding an α-MMS allocation of an instance $(n, \mathcal{M}, v)$ for any $\alpha \in (0, 1]$,. The tricky part in these reductions is to guarantee that an α-MMS allocation for *any* $\alpha > 0$ maps to a solution of PARTITION.

Computing the MMS values. The first step in our PTAS is to compute the MMS values of the agents, which is equivalent to finding an MMS allocation with identical agents. The above hardness result rules out even approximating the MMS values within any non-trivial factor in polynomial time if either condition is not satisfied. For the instances satisfying both, in Section 4 we design an efficient algorithm to compute the MMS values up to a small multiplicative error. We need to tackle the cases with MMS ≥ 0 and MMS < 0 separately; note that the sign of MMS can be easily determined using Lemma 2.1. Formally, we show the following (see Section 4 and Appendix C).

Theorem 3.2. *Given an instance* $(n, \mathcal{M}, v)$ *and a constant* $\epsilon > 0$*, if* (i) n *is a constant and* (ii) *for each* $i \in [n]$, $|v_i(\mathcal{M})| \geq \tau \cdot \min\{v_i^+, v_i^-\}$*, where* $\tau > 0$ *is a constant. Then, there is a* PTAS *to compute a* $(1 - \epsilon)$-MMS *allocation.*

Our PTAS for the α-MMS + PO problem takes as input the instance $(\mathcal{N}, \mathcal{M}, (v_i)_{i \in \mathcal{N}})$, a parameter $\alpha \in (0, 1]$, and constants $\epsilon, \gamma > 0$, and it either finds an allocation that is $(\alpha - \epsilon)^{+}$-MMS + γ-PO allocation, or correctly reports that an α-MMS allocation does not exist; the latter may very well be the case for *any* $\alpha \in (0, 1]$ as shown in Appendix B.

Pre-processing. First, note that the problem is non-trivial only if $\alpha > \epsilon$, otherwise since $(\alpha - \epsilon)^+ = 0$, thus an allocation that gives every item to the agent with the highest value for it is $(\alpha - \epsilon)^+$-MMS + PO, and returned. Therefore, now on we assume that $\alpha > \epsilon$.

Next we re-define ϵ as $\min\{\epsilon, \frac{\gamma\alpha}{(1+\gamma)}\}$. This is done for technical reasons to ensure that the final allocation is also γ-PO. It does not harm the MMS guarantee, as an $(\alpha - \epsilon)^+$-MMS allocation with a smaller ϵ is also an $(\alpha - \epsilon)^+$-MMS allocation with respect to the given ϵ. Note that when α and γ are constants, so is the new value of ϵ. Finally, we assume there are no agents with $v(\mathcal{M}) = 0$. Note that because of condition 2 of the problem, when $v(\mathcal{M}) = 0$ then the value of every item for this agent is 0. Also note that their MMS = 0. Thus, we can allocate all the chores arbitrarily among agents with $v(\mathcal{M}) = 0$, and remove them. It is easy to see that the MMS value of the remaining agents can only improve, and all α-MMS allocations are retained, by the removal of all the chores and a subset of agents. The problem then reduces to a goods manna case with no agents with $v(\mathcal{M}) = 0$, which is solved as a special case of the PTAS we will describe.

Due to the pre-processing step, now on we assume that $(\mathcal{N}, \mathcal{M}, (v_i)_{i\in\mathcal{N}})$, the given fair division instance, satisfies $v_i(\mathcal{M}) \neq 0$ for every agent $i \in \mathcal{N}$. We first scale the valuations so that $|v_i(\mathcal{M})| = n$ since the problem is scale free by Lemma 2.3. Without loss of generality, we assume that the given constants $\alpha, \epsilon, \gamma > 0$ are such that $\alpha > \epsilon$, and $\epsilon \leq \frac{\gamma\alpha}{(1+\gamma)}$. The algorithm first applies the relevant PTAS from Section 4 or Appendix C to compute the MMS value of every agent approximately up to a factor $(1 - \epsilon/2)$. If $\overline{\mathsf{MMS}_i}$ is the value returned by the algorithm for agent i, we know $\overline{\mathsf{MMS}_i} \geq \min\{(1-\epsilon/2)\mathsf{MMS}_i, (1/(1-\epsilon/2))\mathsf{MMS}_i\}$. The algorithm then tries to find an $(\alpha-\epsilon/2)^+$-$\overline{\mathsf{MMS}_i}$ allocation, and fails only when an α-MMS allocation does not exist.

High-level Approach. At a high level, the algorithm to find an $(\alpha - \epsilon/2)^+$-$\overline{\mathsf{MMS}_i}$ allocation is as follows. We will classify all items as BIG, based on if they are highly valued by any agent relative to her MMS value, or SMALL otherwise. Although the MMS values of agents can be arbitrarily small, we show that the number of BIG items is a function of n, hence constant from condition 1. Therefore, we can efficiently enumerate all partitions of the BIG items.

For each partition, we allocate the SMALL items by solving an LP and rounding its solution. The LP ensures a fractional solution where every agent gets at least an α-$\overline{\mathsf{MMS}}$ valued bundle. Next, through a careful rounding, we show that if there is an α-MMS allocation where the BIG items are allocated according to the current partition, then the allocation of all items obtained after rounding the LP solution is $(\alpha - \epsilon/2)^+$-$\overline{\mathsf{MMS}_i}$. Among all the fractional α-MMS allocations found by combining some BIG item partition with the allocation of SMALL as per the LP solution, we find the one, say $\mathcal{A} = [\mathcal{A}_1, \cdots, \mathcal{A}_n]$, with the highest value for the sum of valuations of all the agents, i.e., $\sum_i v_i(\mathcal{A}_i)$. That is, we find a fractional allocation,

$$\mathcal{A} \in \underset{B\in\Pi_n(\mathsf{BIG})}{\operatorname{argmax}} \max_{A_i\supseteq B_i, A \text{ is } \alpha\text{-MMS}} \sum_{i\in\mathcal{N}} v_i(A_i).$$

Finally, we show that the rounded solution, call it $\mathcal{A}^r$, is γ-PO, by showing that for an allocation to γ-Pareto dominate $\mathcal{A}^r$, it must be an α-MMS allocation and have higher welfare than $\mathcal{A}$. This proof is quite involved and uses several new ideas, including the way we round the LP solution, to show Pareto optimality of the integral allocation[3]. The following bound on the MMS_i values will be useful in the analysis, and follows from Lemma 2.2 and $|v_i(\mathcal{M})| = n,\ \forall i \in \mathcal{N}$.

Lemma 3.1. *For each agent $i \in \mathcal{N}$, $\mathsf{MMS}_i \leq 1$ if $v(\mathcal{M}) \geq 0$, otherwise $\mathsf{MMS}_i \leq -1$.*

In the remaining section, we will discuss the details and formalize these ideas, with some proofs moved to Appendix A in order to convey the main ideas within the page limit. Also, for brevity, at times we will refer to $\overline{\mathsf{MMS}_i}$ as $\tilde{\mu}_i$ and to MMS_i as μ_i.

3.1 BIG and SMALL Items

Next we classify items into sets BIG and SMALL, and show bounds on the size of the BIG items set.

Definition 3.1 (Big and Small items). *The sets of all BIG goods (BIG_i^+) and BIG chores (BIG_i^-) of agent i are defined as,*

$$\mathsf{BIG}_i^+ := \{j \in \mathcal{M}^+ \mid (\tilde{\mu}_i \geq 0 \text{ and } v_{ij} > \epsilon\tilde{\mu}_i/(2n)) \text{ or } (\tilde{\mu}_i < 0 \text{ and } v_{ij} > \epsilon/(2n))\}, \text{ and}$$
$$\mathsf{BIG}_i^- := \{j \in \mathcal{M}^- \mid -v_{ij} > \epsilon/(2n)\}.$$

The union of all sets BIG_i^+ is called $\mathsf{BIG}^+ = \cup_i \mathsf{BIG}_i^+$, and of all BIG_i^- sets is called $\mathsf{BIG}^- = \cup_i \mathsf{BIG}_i^-$. Finally, the set of all BIG items is called $\mathsf{BIG} := \mathsf{BIG}^+ \cup \mathsf{BIG}^-$.

Any item that is not in BIG is called a SMALL item. We define SMALL goods and chores for agent i as $\mathsf{SMALL}_i^+ = \mathcal{M}_i^+ \backslash \mathsf{BIG}_i^+$, and $\mathsf{SMALL}_i^- = \mathcal{M}_i^- \backslash \mathsf{BIG}_i^-$. Similarly, the sets of SMALL goods, SMALL chores, and SMALL items are respectively $\mathsf{SMALL}^+ = \mathcal{M}^+ \backslash \mathsf{BIG}^+$, $\mathsf{SMALL}^- = \mathcal{M}^- \backslash \mathsf{BIG}^-$, and $\mathsf{SMALL} = \mathsf{SMALL}^+ \cup \mathsf{SMALL}^-$.

In the remaining section, we will show the size of BIG is constant. For this, we make two useful observations, then show the bound on BIG.

Claim 3.1. *For the approximate MMS values $\tilde{\mu}_i$, we have, if $\mu_i > 0$, then $\tilde{\mu}_i \in [(1 - \epsilon/2)\mu_i, \mu_i]$, if $\mu_i = 0$ then $\tilde{\mu}_i = 0$ and if $\mu_i < 0$, then $\tilde{\mu}_i \in [\mu_i/(1 - \epsilon/2), \mu_i]$.*

The claim follows from the guarantees of Theorems 4.1 and C.1. Next, recall the definitions of v_i^+ and v_i^- from equation (2). The next claim follows from condition 2 of the problem.

Claim 3.2. *For all agents i, $v_i^+ \leq O(n), v_i^- \leq O(n)$.*

Next lemma shows a bound on $|\mathsf{BIG}|$ (proof in Appendix A). For this we show the bound of $O(n^2/\epsilon)$ on $|\mathsf{BIG}_i^+|$ and $|\mathsf{BIG}_i^-|$ for each agent i. Note that if $\tilde{\mu}_i$ is big enough then it is easy to prove that $|\mathsf{BIG}_i^+|$ is a constant. The difficulty is when $\tilde{\mu}_i$ is arbitrarily small, in which case $|\mathsf{BIG}_i^+|$ can potentially be large – a tricker case. The bound on $|\mathsf{BIG}_i^-|$ follows from the definition of BIG_i^- together with Claim 3.2.

Lemma 3.2. *The number of big items, i.e., $|\mathsf{BIG}| \leq O(n^3/\epsilon)$.*

3.2 LP for Allocating SMALL Items, and Rounding

Given a partition $B^\pi = (B_1, \ldots, B_n)$ of BIG items, next we write an LP to find a *fractional* allocation of SMALL items such that together with B^π this allocation gives at least α-$\overline{\mathsf{MMS}}$ value to every

[3]Recall that testing PO is coNP-hard [ABL[+]16], and market equilibrium (or highest sum of valuations for the trivial PO allocation) is the only technique in all known literature to certify that an allocation is PO.

agent. If there exists an α-MMS allocation where the BIG items are allocated as per B^{π} then we show that the LP has to be feasible.

For every agent i, denote by c_i the value from SMALL that i needs for her bundle's value to be at least $\alpha \cdot \tilde{\mu}_i$ if $\tilde{\mu}_i \geq 0$ or $(1/\alpha) \cdot \tilde{\mu}_i$ otherwise, i.e., $c_i = \min\{(1/\alpha)\tilde{\mu}_i, \alpha\tilde{\mu}_i\} - v_i(B_i)$.

$$\max \quad \sum_{i \in \mathcal{N}} \left(\sum_{j \in \mathsf{SMALL}_i^+} v_{ij} x_{ij} - \sum_{j \in \mathsf{SMALL}_i^-} |v_{ij}| x_{ij} \right) \tag{3}$$

$$\text{s.t.} \quad \sum_{j \in \mathsf{SMALL}_i^+} v_{ij} x_{ij} - \sum_{j \in \mathsf{SMALL}_i^-} |v_{ij}| x_{ij} \geq c_i, \qquad \forall i \in \mathcal{N} \tag{4}$$

$$\sum_{i \in \mathcal{N}} x_{ij} \leq 1, \qquad \forall j \in \mathsf{SMALL}^+ \tag{5}$$

$$\sum_{i \in \mathcal{N}} x_{ij} \geq 1, \qquad \forall j \in \mathsf{SMALL}^- \tag{6}$$

$$x_{ij} \geq 0, \qquad \forall i \in \mathcal{N}, j \in \mathcal{M}. \tag{7}$$

We now prove two properties (Lemmas 3.3 and 3.4) that will help in obtaining an integral $(\alpha - \epsilon/2)$-$\overline{\mathsf{MMS}}$ allocation of items from a fractional α-$\overline{\mathsf{MMS}}$ allocation. Let us assume the LP has a solution, say $x = [x_{ij}]_{i \in \mathcal{N}, j \in \mathsf{SMALL}}$. We define a bipartite graph, called the *allocation graph*, corresponding to x as follows. There is a vertex corresponding to each agent in $\mathcal{N}$ in one part of vertices, and to each item in SMALL in the other part, and for all $i \in \mathcal{N}$ and $j \in \mathsf{SMALL}$, edge (i, j) exists if $x_{ij} > 0$. We show the following property of the allocation graph.

Lemma 3.3. *The allocation graph of* any *LP solution x can be made acyclic in such a way that in the allocation corresponding to the new graph, say $x' = [x'_{ij}]_{i \in \mathcal{N}, j \in \mathsf{SMALL}}$, every agent receives a bundle of the same or better value as in x.*

To prove the lemma, we show re-allocations can be done along any cycle in a certain way without any agent losing any value that eliminates at least one edge. For every cycle, we define a particular scaled valuation function, and define weights for the edges to reflect the values to agents from the adjacent items. Then we add and subtract weights in a certain way along the cycle, taking into consideration if the adjacent item is a good or a chore, so that the allocation corresponding to the new weights, or equivalently (scaled) values to agents, does not contain this cycle.

The next lemma follows since an undirected, acyclic graph forms a tree.

Lemma 3.4. *The number of shared items in any acyclic allocation graph is at most $n - 1$.*

Next we describe the notion of envy graph [LMMS04], a directed graph corresponding to any allocation, that will be used to round the LP solutions.

Envy Graph and Cycle Elimination. Given a set of agents $\mathcal{N}$ and an *integral* allocation $\mathcal{A}$ of a set of items among them, each node in the graph corresponds to an agent in $\mathcal{N}$. There is a directed edge $(i \to k)$ corresponding to agents i and k if agent i values agent k's allocation more than her own. It is shown in [LMMS04] that the allocation can be modified so that its corresponding envy graph is acyclic, and no agent's valuation decreases. This is done by giving each agent in a cycle the bundle of her successor. The graph is updated and the process repeated until all cycles are eliminated. This process can be done efficiently [LMMS04].

Claim 3.3. *In an allocation of $\mathcal{M}$ among n agents, every sink agent i corresponding to an acyclic envy graph has value at least* 1 *for her own bundle if $v_i(\mathcal{M}) > 0$, and at least* -1 *otherwise.*

Rounding the LP. Using Lemmas 3.3 and 3.4, we first modify the allocation graph of the LP solution so that it is a forest graph with at most $n-1$ shared items. Let S be the set of all the shared items, $S^{-\epsilon}$ the set of all the shared chores whose absolute value is more than $\epsilon|\tilde{\mu}_i|/(2n)$ for at least one agent, that is, $S^{-\epsilon} := \{j \in S \mid \exists i \in \mathcal{N}, |v_{ij}| > \epsilon|\tilde{\mu}_i|/(2n)\}$, and $S^{+} := S \setminus S^{-\epsilon}$. Allocate each item j in S^{+} to any agent i in $\text{argmax}_i\, v_{ij}$. Then consider the envy graph corresponding to this allocation of $\mathcal{M} \setminus S^{-\epsilon}$, and modify the allocation by eliminating all the cycles in the envy graph. Allocate all the items in $S^{-\epsilon}$ to a sink agent in the acyclic envy graph, and denoted it as i^t.

The following claim will be useful in proving the final allocation of the algorithm is γ-PO.

Claim 3.4. *If $S^{-\epsilon} \neq \emptyset$ then there exists an $i \in \mathcal{N}$ such that $v_i(\mathcal{M}) > 0$.*

Proof. Every agent with $v(\mathcal{M}) < 0$ has $\mu \leq -1$, from Lemma 3.1. The value of any chore in SMALL for any such agent is at most $\epsilon/2n \leq \epsilon|\mu|/2n \leq \epsilon|\tilde{\mu}|/2n$, as from Claim 3.1, $|\tilde{\mu}| \geq |\mu|$. Hence, if $S^{-\epsilon} \neq \emptyset$, then the agent who values any item in $S^{-\epsilon}$ more than $\epsilon\tilde{\mu}/(2n)$ has $v(\mathcal{M}) > 0$. □

Finally, we show the maximum loss in value of each agent in the rounding process, which will be used to ensure that the algorithm returns an $(\alpha - \epsilon)^{+}$-MMS allocation.

Lemma 3.5. *In the rounding process, i^t loses at most $\epsilon/2$ value and every other agent i loses at most $\epsilon|\tilde{\mu}_i|/2$ value.*

Proof. Every agent except i^t, in the worst case, loses all her shared goods and gains all her shared chores in S^{+}, and has no shared chores in $S^{-\epsilon}$, as she only gains from the rounding of items in $S^{-\epsilon}$. Her maximum loss from the items in S^{+} is at most $(n-1) \cdot \epsilon\tilde{\mu}/(2n) \leq \epsilon\tilde{\mu}/2$, as $|S^{+}| \leq |S| \leq n-1$, from Lemma 3.4. For agent i^t, her loss from S in the worst case is at most $(n-1) \cdot \epsilon/(2n) \leq \epsilon$, as each item in S has absolute value at most $\epsilon/(2n)$ for her. □

3.3 PTAS for α-MMS + PO

Algorithm 1 combines the ideas in the previous sections and finds an $(\alpha - \epsilon)^{+}$-MMS allocation if an α-MMS allocation exists, else returns an empty allocation to indicate that no α-MMS allocation exists. The algorithm works as follows. First, it finds the approximate MMS values of all agents using the algorithms from Section 4 and Appendix C, and classifies all items as BIG or SMALL. Then among all allocations of BIG items where the corresponding LP has a solution, if any, it finds the combined allocation of BIG and SMALL, called $\mathcal{A}$, with the highest social welfare where SMALL may be fractionally allocated.

From constraint 4 of the LP, $\mathcal{A}$ is α-MMS, and as shown in Lemma 3.6, its rounded allocation, denoted as $\mathcal{A}^r$, is $(\alpha - \epsilon)^{+}$-MMS. To ensure $\mathcal{A}^r$ is also γ-PO, for technical reasons we require the sum of absolute values of all agents to be at least α whenever there is at least one agent with $v_i(\mathcal{M}) > 0$. If $\mathcal{A}^r$ does not satisfy this condition, Algorithm 1 ensures a stronger guarantee, namely at least one agent values her own bundle at least 1, by modifying $\mathcal{A}^r$ as follows. Let $\mathcal{N}^{+}$ be the set of agents with $v_i(\mathcal{M}) > 0$. Note that, for an $i \in \mathcal{N}^{+}$, $v_i(\mathcal{M}) = n$ but $v_i(A_i^r) < \alpha$, and hence there exists an agent $k \neq i$ such that i values k's bundle more than 1. We consider two cases based on if k is in $\mathcal{N}^{+}$ or not. If $k \notin \mathcal{N}^{+}$, then k's MMS value is negative, thus even if we re-allocate her bundle to i and give k nothing (lines 18-19), her $(\alpha - \epsilon)^{+}$-MMS guarantee is maintained. If no such (i, k) pair is found, then we go to the other case, where k has to be given something. For this (lines 21-22), we construct a graph on $\mathcal{N}^{+}$ where there is an edge from i to k if $v_i(A_i^r) < \alpha$ and $v_i(A_k^r) \geq 1$. This graph has to have a cycle (See proof of Claim 3.6), and swapping bundles along the cycle gives value more than 1 to every agent along the cycle.

ALGORITHM 1: $(\alpha-\epsilon)^+$-MMS + γ-PO allocation of mixed items to non-identical agents

Input : Instance $(\mathcal{N}, \mathcal{M}, (v_i)_{i\in\mathcal{N}})$, $\alpha \in (\epsilon, 1]$, $\gamma > 0$ and $\epsilon > 0$
Output : $(\alpha-\epsilon)^+$-MMS + γ-PO allocation or report α-MMS allocation does not exist

1 $\epsilon \leftarrow \min\{\epsilon, \frac{\gamma\alpha}{(1+\gamma)}\}$, $flag \leftarrow false$
2 For all $i \in \mathcal{N}$, $\mathcal{A}^r_i \leftarrow \emptyset$ // initialize $\mathcal{A}^r$ as the empty allocation
3 $\mathcal{A} \leftarrow$ lowest social welfare allocation, i.e., give every item j to agent i with smallest v_{ij}
4 For all i, $\overline{\text{MMS}_i} \leftarrow (1-\frac{\epsilon}{2})$-MMS$_i$ value of agent i // use Algorithm from Section 4
5 Define BIG and SMALL according to Definition 3.1
6 **for** *all allocations* $B^\pi = [B_1, B_2 \ldots, B_n]$ *of* BIG **do**
7   Solve the LP (Equations (3)-(7)) for allocating SMALL items
8   **if** *LP has a solution* **then**
9         $flag \leftarrow true$
10         $A^\pi \leftarrow$ Allocation of SMALL in optimal LP combined with B^π
11         $\mathcal{A} \leftarrow$ Allocation from $(\mathcal{A}, A^\pi)$ with higher welfare, i.e., $\sum_i v_i(\mathcal{A}_i)$
12 **if** $flag = true$ **then**
13   Make allocation graph of $\mathcal{A}$ acyclic using Lemma 3.3
14   Round off $\mathcal{A}$ and obtain $\mathcal{A}^r$ by applying the rounding method from Section 3.2
15   **if** $\sum_i |v_i(\mathcal{A}^r_i)| < \alpha$, *and* $\exists i : v_i(\mathcal{M}) > 0$ // technical steps to get γ-PO
16   **then**
17         $\mathcal{N}^+ \leftarrow$ set of agents i with $v_i(\mathcal{M}) > 0$
18   **if** $\exists i \in \mathcal{N}^+$, $k \notin \mathcal{N}^+ : v_i(\mathcal{A}^r_k) \geq 1$ **then**
19   modify $\mathcal{A}^r$ by giving $\mathcal{A}^r_k$ to i, and giving k nothing // note: k has negative MMS
20   **else**
21   Construct the following directed graph $G(V, E)$. $V = \mathcal{N}^+$, directed edge $(i \longrightarrow k) \in E$ if $v_i(\mathcal{A}^r_i) < \alpha$ and $v_i(\mathcal{A}^r_k) \geq 1$ // G has a cycle as for all $i \in V : v_i(\mathcal{A}^r_i) < \alpha \leq 1$
22   Swap bundles along any 1 cycle in G by giving every agent her successor's bundle
23 **return** $\mathcal{A}^r$

We will prove the correctness of the algorithm in the remaining section. In what follows, we denote by $\mathcal{A} = [\mathcal{A}_1 \cdots, \mathcal{A}_n]$ and $\mathcal{A}^r = [\mathcal{A}^r_1 \cdots, \mathcal{A}^r_n]$ respectively the fractional allocation that is rounded after Line 12 and its rounded allocation. First we show that the algorithm returns an $(\alpha-\epsilon)^+$-MMS allocation if an α-MMS allocation exists.

Lemma 3.6. *If the LP has a solution for any partition of* BIG*, then* $\mathcal{A}^r$ *is an* $(\alpha-\epsilon)^+$-MMS *allocation.*

Proof. First, we argue for the allocation obtained after the rounding step on Line 14. Consider agent i^t. Since i^t corresponds to a sink node in the envy graph, from Claim 3.3 and Lemmas 3.5 and 3.1, her value for her bundle in $\mathcal{A}^r$ is at least $1-\epsilon/2 \geq 1-\epsilon \geq (\alpha-\epsilon)\mu_{i^t}$ if $\tilde{\mu}_{i^t} \geq 0$, and $-1-\epsilon/2 \geq -1-\epsilon \geq (1+\epsilon)\mu_{i^1} \geq \frac{1}{(\alpha-\epsilon)}\mu_{i^1}$ otherwise. Next, every agent i except i^t, according to constraint (4) of the LP, receives a bundle of value at least c_i from SMALL in the fractional allocation of SMALL corresponding to $\mathcal{A}$. Thus, for all $i \neq i^t$, their value for their bundle in $\mathcal{A}^r$ is at least $v_i(B_i) + c_i - n\epsilon \cdot |\tilde{\mu}_i|/(2n) \geq \min\{(1/\alpha)\tilde{\mu}_i, \alpha\tilde{\mu}_i\} - \epsilon \cdot |\tilde{\mu}_i|/2$, from Lemma 3.5 and by definition of c_i. Combined with Claim 3.1, when $\tilde{\mu}_i \geq 0$, this is at least $(\alpha-\epsilon/2)\tilde{\mu}_i \geq (\alpha-\epsilon/2)(1-\epsilon/2)\mu_i \geq (\alpha-\epsilon)\mu_i$. When $\tilde{\mu}_i < 0$, this value is at least $\frac{1}{\alpha}\tilde{\mu}_i + \epsilon\tilde{\mu}_i/2$. As $(\frac{1}{\alpha}+\epsilon/2) \leq \frac{1}{(\alpha-\epsilon/2)}$, and $\tilde{\mu}_i < 0$, along with Claim 3.1, $(\frac{1}{\alpha}+\epsilon/2)\tilde{\mu}_i \geq \frac{1}{(\alpha-\epsilon/2)(1-\epsilon/2)}\mu_i \geq \frac{1}{(\alpha-\epsilon)}\mu_i$.

Any further modifications of the allocation can occur when (a) there is a pair of agents i, k with $v_i(\mathcal{A}^r_k) > 1$ and $v_k(\mathcal{M}) < 0$, or when (b) there is a cycle of agents with value at most α for their own bundle and value at least 1 for the next. The only agents whose value decreases in these steps are those with $v_k(\mathcal{M}) < 0$, who after the swap receive no item. As $v_k(\mathcal{M}) < 0$, then $\mu_k < 0$, hence they receive at least $\mu_k \geq \frac{1}{(\alpha-\epsilon)}\mu_k$ valued bundle. As no other agents lose, they still retain an $(\alpha-\epsilon)^+$-MMS bundle. □

Note that, the steps after rounding maintains the MMS guarantee. Since by construction, the LP has to be feasible whenever BIG items are allocated as per an α-MMS allocation, we get as a corollary,

Corollary 3.1. *If an* α-MMS *allocation exists, Algorithm 1 returns an* $(\alpha-\epsilon)^+$-MMS *allocation.*

Finally, we show the approximate Pareto optimality of the $(\alpha-\epsilon)^+$-MMS allocation returned. This is the more involved part. For this, we use the notion of *social welfare* of any allocation, defined as the sum of values of all agents. Formally, for an allocation $A = [A_1 \cdots, A_n]$ among n agents, define its social welfare as $w(A) = \sum_i v_i(A_i)$.

Lemma 3.7. *If an* α-MMS *allocation exists, then* $\mathcal{A}$ *has the highest welfare among all the* α-MMS *allocations of* $\mathcal{M}$ *among* $\mathcal{N}$ *obtained by allowing* SMALL *to be fractionally allotted.*

Proof. Let $\mathcal{S}^B \subseteq \Pi_n(\mathsf{BIG})$ be the set of all partitions of BIG corresponding to which there is a fractional α-MMS allocation, or in other words, for which the LP has a solution. For every partition B^π in $\mathcal{S}^B$, the objective function of the LP ensures that the allocation of SMALL returned by the algorithm has the highest social welfare among all allocations that satisfy the LP constraints. Hence, among all α-MMS allocations where the partition of BIG is B^π, the allocation returned, say A^π, has the highest social welfare. Formally, let $\mathcal{S}^{A,B^\pi}$ be the set of all α-MMS allocations corresponding to the partition B^π. Then $A^\pi \in \operatorname{argmax}_{A\in\mathcal{S}^{A,B^\pi}} \sum_i w(A)$.

Let the set of allocations A^π, one corresponding to each partition $B^\pi \in \mathcal{S}^B$, be $\mathcal{S}^A$. From Line 11 of the algorithm, $\mathcal{A}$ has the highest social welfare among all. Formally, $\mathcal{A}$ is in $\operatorname{argmax}_{A\in\mathcal{S}^A}\{w(A)\}$. Combining with the above characterization of the allocations in $\mathcal{S}^A$, we have,

$$\mathcal{A} \in \max_{B^\pi\in\mathcal{S}^B} \operatorname*{argmax}_{A\in\mathcal{S}^{A,B^\pi}} \{w(A)\},$$

thus proving the lemma. □

Next, we prove two key properties (Lemmas 3.8 and 3.9) of any *integral* allocation that γ-Pareto dominates $\mathcal{A}^r$. Suppose $\mathcal{A}^*$ is such an allocation.

Lemma 3.8. $\mathcal{A}^*$ *is an integral* α-MMS *allocation.*

Proof. We will use the following relation between ϵ, α and γ. We have,

$$\epsilon \leq \frac{\gamma\alpha}{(1+\alpha)} \Rightarrow \epsilon \leq \gamma(\alpha-\epsilon) \Rightarrow \gamma \geq \frac{\epsilon}{(\alpha-\epsilon)} \Rightarrow (1+\gamma) \geq \frac{\alpha}{(\alpha-\epsilon)}. \tag{8}$$

We know that $\mathcal{A}^r$ is an $(\alpha-\epsilon)^+$-MMS allocation. Hence, agents i with $\mu_i \geq 0$ get a bundle of value at least $(\alpha-\epsilon)\mu_i \geq 0$ in $\mathcal{A}^r$, hence get a bundle of value at least $(1+\gamma)(\alpha-\epsilon)\mu_i$ in $\mathcal{A}^*$. From equation (8), this is at least $\alpha\mu_i$. Next, consider agents with $\mu_i < 0$. If they receive a bundle of positive value in $\mathcal{A}^r$, then they also receive a positive valued, hence a bundle of value more than $\mu_i \geq \frac{1}{\alpha}\mu_i$, in $\mathcal{A}^*$. And if they get a negative valued bundle of value at least $\frac{1}{(\alpha-\epsilon)}\mu_i$ in $\mathcal{A}^r$, then they get a bundle of value at least $\frac{1}{(1+\gamma)(\alpha-\epsilon)}\mu_i$, which from equation (8) and the fact that $\mu_i < 0$, is at least $\frac{1}{\alpha}\mu_i$. Hence, $\mathcal{A}^*$ is an integral α-MMS allocation. □

The next property of $\mathcal{A}^*$ is that it will have higher social welfare than (fractional allocation) $\mathcal{A}$. To prove this, we first prove two technical claims.

Claim 3.5. *$w(\mathcal{A}^r) \geq w(\mathcal{A}) - \epsilon$ if $S^{-\epsilon} \neq \emptyset$, else $w(\mathcal{A}^r) \geq w(\mathcal{A})$.*

Proof. We consider each step in Algorithm 1 that changes the allocation from $\mathcal{A}$ to $\mathcal{A}^r$, and see how it changes the social welfare. The first step is making the allocation graph of $\mathcal{A}$ acyclic. Here every agent's value, hence the social welfare also remains the same. The next step is the rounding process. Here, first the items in S^+ are allotted to the agents with the highest value for them, hence the sum of values of the items from S^+ in $\mathcal{A}^r$ is at least as much as that in $\mathcal{A}$. As no other item's allocation changes, the social welfare from them remains the same. Hence this part only improves the social welfare. Then the envy graph cycle elimination only improves the value of every agent, hence does not reduce the social welfare. At this point, $w(\mathcal{A}^r) \geq w(\mathcal{A})$. If $S^{-\epsilon} = \emptyset$, the rounding process ends, hence this inequality holds, otherwise from Lemma 3.5, allocating the items from $S^{-\epsilon}$ reduces the sink agent's value, hence the social welfare, by at most ϵ, giving $w(\mathcal{A}^r) \geq w(\mathcal{A}) - \epsilon$. Now we show the next part of the algorithm does not reduce the social welfare of $\mathcal{A}^r$, hence these relations remain true, thus proving the claim.

First consider the if statement on Line 18. For agents i, k when k's bundle is given to i, only the allocation of items in $\mathcal{A}^r_k$ changes and the allocation of $\mathcal{M}\backslash\mathcal{A}^r_k$ remains the same. Now $v_k(\mathcal{A}^r_k) < \alpha$ and $v_i(\mathcal{A}^r_k) > 1$, otherwise this step would not be executed. Hence the social welfare changes by $v_i(\mathcal{A}^r_k) - v_k(\mathcal{A}^r_k) > 1 - \alpha \geq 0$. Finally, suppose some bundles are swapped along some cycle by executing Line 22. Every agent had value at most α for their bundle, and received a bundle of value at least 1. Thus, the value of every agent in the cycle changes by at least $1-\alpha > 0$, and every other agent's bundle, hence its value, remains the same. Thus, the social welfare does not decrease in this step as well. □

Claim 3.6. *If there is an agent i with $v_i(\mathcal{M}) > 0$, then there exists some agent i' with $v_{i'}(\mathcal{A}^r_{i'}) \geq \alpha$.*

Proof. If the claim is true for the allocation obtained after rounding $\mathcal{A}$, then this is the allocation returned, hence we are done. Otherwise, the if condition of Line 15 is executed. If the condition of Line 15 is true, then as the allocation before this line was $(\alpha-\epsilon)^+$-MMS, $v_i(\mathcal{A}^r_i) \geq 0$, and after obtaining k's bundle, $v_i(\mathcal{A}^r_i \cup \mathcal{A}^r_k) \geq 0 + 1 = 1$.

Otherwise, for every agent i in $\mathcal{N}^+$ we have $v(\mathcal{M}) = n$, thus there is at least one agent $k \in \mathcal{N}^+$ such that $v_i(\mathcal{A}^r_k) \geq 1$. The graph $G(V, E)$ has an edge $i \longrightarrow k$ in this case. As $\sum_i |v_i(\mathcal{A}^r_i)| < \alpha$, for every i we have $v_i(\mathcal{A}^r_i) < \alpha \leq 1$. Thus for every edge $i \neq k$. Hence, G has at least one cycle. After swapping along a cycle, all agents along the cycle receive the bundle of their successor, hence have value at least 1 for their own bundle. Thus, the final allocation has some agent with value at least $1 \geq \alpha$ for her bundle. □

Lemma 3.9. *$\mathcal{A}^*$ has higher social welfare than $\mathcal{A}$.*

Proof. For cleaner exposition, we will denote $v_i(A_i)$ by $v_i(A)$ for any allocation A.

By definition of a γ-Pareto dominating allocation, the social welfare of allocation $\mathcal{A}^*$ is,

$$\begin{aligned} w(\mathcal{A}^*) &> (1+\gamma)\cdot \sum_{i:v_i(\mathcal{A}^r)\geq 0} v_i(\mathcal{A}^r) + \frac{1}{1+\gamma}\cdot \sum_{i:v_i(\mathcal{A}^r)<0} v_i(\mathcal{A}^r) \\ &= \sum_i v_i(\mathcal{A}^r) + \gamma \cdot \sum_{i:v_i(\mathcal{A}^r)\geq 0} v_i(\mathcal{A}^r) - \frac{\gamma}{(1+\gamma)}\cdot \sum_{i:v_i(\mathcal{A}^r)<0} v_i(\mathcal{A}^r) \\ &= w(\mathcal{A}^r) + \gamma\cdot \sum_{i:v_i(\mathcal{A}^r)\geq 0} v_i(\mathcal{A}^r) + \frac{\gamma}{(1+\gamma)}\cdot \sum_{i:v_i(\mathcal{A}^r)<0} |v_i(\mathcal{A}^r)|. \end{aligned} \tag{9}$$

Let $\mathbb{1}_{S^{-\epsilon}}$ be an indicator variable for whether $S^{-\epsilon}$ is non-empty. Substituting the relation between $w(\mathcal{A})$ and $w(\mathcal{A}^r)$ from Claim 3.5 in equation (9) we get, $w(\mathcal{A}^*) > w(\mathcal{A}) - \epsilon\cdot\mathbb{1}_{S^{-\epsilon}} + \gamma\cdot\sum_{i:v_i(\mathcal{A}^r)\geq 0} v_i(\mathcal{A}^r) + \frac{\gamma}{(1+\gamma)}\cdot\sum_{i:v_i(\mathcal{A}^r)<0}|v_i(\mathcal{A}^r)|$. Hence, to prove the lemma, it suffices to show,

$$\gamma\cdot \sum_{i:v_i(\mathcal{A}^r)\geq 0} v_i(\mathcal{A}^r) + \frac{\gamma}{(1+\gamma)}\cdot \sum_{i:v_i(\mathcal{A}^r)<0} |v_i(\mathcal{A}^r)| - \epsilon\cdot \mathbb{1}_{S^{-\epsilon}} \geq 0. \tag{10}$$

If $S^{-\epsilon} = \emptyset$, we are done, as all values on the left hand side in the above equation, say L, are non-negative. Hence, we now prove the equation when $S^{-\epsilon} \neq \emptyset$, that is, $\mathbb{1}_{S^{-\epsilon}} = 1$.

We know $\epsilon \leq \frac{\gamma\alpha}{(1+\gamma)}$, and as $\gamma \geq 0$, $\gamma \geq \frac{\gamma}{1+\gamma}$. Substituting these relations in L, we have,

$$L \geq \frac{\gamma}{1+\gamma}\cdot \sum_{i:v_i(\mathcal{A}^r)\geq 0} v_i(\mathcal{A}^r) + \frac{\gamma}{(1+\gamma)}\cdot \sum_{i:v_i(\mathcal{A}^r)<0} |v_i(\mathcal{A}^r)| - \frac{\gamma\alpha}{(1+\gamma)} = \frac{\gamma}{1+\gamma}\left(\sum_i |v_i(\mathcal{A}^r)| - \alpha\right).$$

But as $S^{-\epsilon} \neq \emptyset$, from Claim 3.4, there is some agent i with $v_i(\mathcal{M}) > 0$. Then from Claim 3.6, $v_{i'}(\mathcal{A}^r) \geq \alpha$ for at least one agent i'. Hence, $\frac{\gamma}{1+\gamma}\left(\sum_i |v_i(\mathcal{A}^r)| - \alpha\right) \geq 0$, thus proving equation (10), hence the lemma. □

Corollary 3.2. *If an α-*MMS *allocation exists, Algorithm 1 returns a γ-*PO *allocation.*

Proof. For contradiction, suppose $\mathcal{A}^r$ is not γ-PO. Then there is another allocation, say $\mathcal{A}^*$, that γ-Pareto dominates $\mathcal{A}^r$. From Lemmas 3.8 and 3.9, $\mathcal{A}^*$ is an integral α-MMS allocation with higher social welfare than $\mathcal{A}$. From Lemma 3.7, this is a contradiction. □

Using Corollaries 3.1 and 3.2, the next theorem obtains the main result.

Theorem 3.3. *Given an instance $(\mathcal{N}, \mathcal{M}, (v_i)_{i\in\mathcal{N}})$ and constants $\alpha, \epsilon, \gamma > 0$, that is, an instance of the α-*MMS + PO *problem, Algorithm 1 returns an $(\alpha-\epsilon)^+$-*MMS *+ γ-*PO *allocation if an α-*MMS *allocation exists, else reports it does not exist, in time $O(2^{O((n^3\log n)/\min\{\epsilon^2,\gamma^2/2\})}m^3)$ where $n = |\mathcal{N}|$ is a constant. Thus, it is a* PTAS.

3.4 PTAS for OPT-α-MMS + PO

Our final goal is to solve OPT-α-MMS + PO problem, that is to find α-MMS + PO allocation for the highest possible α. In this section we design a PTAS for this problem: given $(\mathcal{N}, \mathcal{M}, (v_i)_{i\in\mathcal{N}})$, and constants $\epsilon, \gamma > 0$, we design a polynomial-time algorithm to find $(\alpha-\epsilon)^+$-MMS + γ-PO allocation for the highest possible α. For this we will use the PTAS for α-MMS + PO problem described in the previous section.

Note that, for a given α, Algorithm 1 either returns $(\alpha-\epsilon)^+$-MMS + γ-PO allocation, or returns an *empty allocation.* And by Theorem 3.3, whenever it returns an empty allocation no α-MMS allocation exists. Using this, we run a simple binary search to find the highest value of $\alpha \in [\epsilon, 1]$

(up to a polynomial precision) for which Algorithm 1 returns a non-empty allocation. If empty allocation is returned for every α in our search, then we only need to ensure PO and therefore we return the social welfare maximizing allocation obtained by giving every item to the agent who values it the most.

We stop when the range of α values under consideration is $[c-\delta, c+\delta]$ for $\delta = \frac{1}{2^{poly(m)}}$ for some constant c, where $m = |\mathcal{M}|$. Clearly the number of iterations the binary search will take to get within such a range is at most $poly(m)$. Each iteration runs Algorithm 1 once, and hence finishes in $O(m^3)$ time (Theorem 3.3). Thus the overall running time of the algorithm is $poly(m)$, and the next theorem follows.

THEOREM 3.4. *Given an instance* $(\mathcal{N}, \mathcal{M}, (v_i)_{i\in\mathcal{N}})$ *and constants* $\epsilon, \gamma > 0$, *that is, an instance of the* OPT-α-MMS + PO *problem, there is a* PTAS *that runs for* $(2^{O(1/\min\{\epsilon^2,\gamma^2\})} poly(m))$ *time and returns an* $(\alpha-\epsilon)^+$-MMS + γ-PO *allocation such that for any* $\alpha' > \alpha + \frac{1}{2^{m^c}}$, *no* α'-MMS *allocation exists, where* $c > 0$ *is a constant.*

This completes the discussion of the OPT-α-MMS + PO problem. Next, we describe a PTAS for finding the MMS value of an agent for distributing a set of items $\mathcal{M}$ into n bundles according to a valuation function $v : \mathcal{M} \to \mathbb{R}$. We refer to this problem as the α-MMS problem with (n) identical agents. Using Lemma 2.1 we can find the sign of the MMS value. Hence, we describe two algorithms, one for each case when MMS ≥ 0 and otherwise. The following section discusses the algorithm for the former case.

4 FINDING MMS VALUES OF AGENTS WHEN MMS ≥ 0

In this section we prove Theorem 3.2 for the case when MMS ≥ 0, i.e., given an instance $(n, \mathcal{M}, v)$, we describe an algorithm to find a $(1-\epsilon)$-MMS allocation for any constant $\epsilon > 0$. Using scale invariance (Lemma 2.3), here on we assume $v(\mathcal{M}) = n$ without loss of generality. Due to Lemma 2.2, this implies MMS $\leq v(\mathcal{M})/n = 1$.

The high level ideas used in the algorithm are as follows. First is a classification of all the items into two sets, Big and Small, based their value (Section 4.1). Using this we prove that |Big| is constant, which allows the enumeration of all allocations of Big, referred as *partitions* of Big to avoid confusion with allocations of $\mathcal{M}$. Next in Section 4.2 we explain a short procedure which allows us to characterize partitions of Big as *valid* or *invalid*. We show that there is at least one valid partition corresponding to which there is an MMS allocation, hence all invalid partitions can be discarded. Finally in Section 4.3, we describe a sub-routine called Bag-Fill, that greedily allocates or 'fills' the items from Small upon partitions or 'bags' of Big that satisfy certain constraints to obtain a $(1-\epsilon)$-MMS allocation. The main algorithm (Section 4.4) enumerates all partitions of Big, discards the invalid partitions, and applies Bag-Fill if its constraints are satisfied. If the constraints are not satisfied, we show that we can apply the PTAS for obtaining MMS allocations with identical agents for a goods manna, and obtain a $(1-\epsilon)$-MMS allocation.

We now discuss these key ideas formally followed by the algorithm in separate subsections.

4.1 Big and Small items

Given an instance $(n, \mathcal{M}, v)$ and a constant $\epsilon > 0$, let Big be the set of items in $\mathcal{M}$ which have absolute value higher than $\frac{\epsilon}{2}$, i.e.,

$$\text{Big} = \{j \in \mathcal{M} : |v_j| \geq \tfrac{\epsilon}{2}\}.$$

Let Big^+ and Big^- respectively be the sets of the goods and the chores in Big, i.e., $\text{Big}^+ = \text{Big} \cap \mathcal{M}^+$, and $\text{Big}^- = \text{Big} \cap \mathcal{M}^-$. Let Small be the set of small items, i.e., $\mathcal{M} \setminus \text{Big}$, and similarly define the sets of goods (Small^+) and chores (Small^-) in Small.

We abuse notation slightly and call items in the set Big (or Small) as Big (resp. Small) items.

Lemma 4.1. $|\text{Big}| = O(n/\epsilon)$.

Proof. As $v(\mathcal{M}) = n$, we have $n = v(\mathcal{M}^+) - |v(\mathcal{M}^-)|$. Then, as $v(\mathcal{M}^+) \geq (1+\tau)|v(\mathcal{M}^-)|$,

$$n \geq v(\mathcal{M}^+) - \frac{v(\mathcal{M}^+)}{1+\tau} \implies v(\mathcal{M}^+) \leq \frac{n(1+\tau)}{\tau}. \tag{11}$$

Finally, by the definition of Big^+ we have

$$|\text{Big}^+| \leq \frac{v(\mathcal{M}^+)}{\frac{\epsilon}{2}} \leq \frac{2n(1+\tau)}{\epsilon\tau}.$$

Similarly, we have

$$n \geq (1+\tau)|v(\mathcal{M}^-)| - |v(\mathcal{M}^-)| \Rightarrow |v(\mathcal{M}^-)| \leq n/\tau. \tag{12}$$

Thus, the number of Big chores is bounded as $|\text{Big}^-| \leq \frac{2n}{\epsilon\tau}$. Hence, $|\text{Big}| = |\text{Big}^+ \cup \text{Big}^-| = O(\epsilon)$. □

As n and ϵ are constant, Lemma 4.1 implies that all partitions of Big can be enumerated in constant time.

4.2 Valid and Invalid partitions of Big

Given an allocation $A^\pi = [A_1 \cdots, A_n]$ of $\mathcal{M}$, denote by $B^\pi = [B_1 \cdots, B_n]$ the allocation from $\mathcal{M} \setminus \text{Small}^+$. Classify the bundles of A^π based on their value from B^π into sets $\mathcal{B}_1$, $\mathcal{B}_2$, $\mathcal{B}_3$, and $\mathcal{B}_4$, together called the $\mathcal{B}$-sets, as follows.

$$\begin{array}{llll} \mathcal{B}_1 & := \{A_k \in B^\pi : v(B_k) > 1\} & \mathcal{B}_2 & := \{A_k \in B^\pi : 1-\epsilon \leq v(B_k) \leq 1\} \\ \mathcal{B}_3 & := \{A_k \in B^\pi : 0 \leq v(B_k) < 1-\epsilon\} & \mathcal{B}_4 & := \{A_k \in B^\pi : v(B_k) < 0\} \end{array} \tag{13}$$

We will abuse notation to denote all items in the sets in any $\mathcal{B}_i$, i.e. $\cup_{\mathcal{A}_k \in \mathcal{B}_i} \mathcal{A}_k$, by $\mathcal{B}_i$.

Given a partition of Big B^π, we now explain a procedure using which we classify the partition as valid or invalid. First classify the bundles of B^π into four sets as per equation (13). Initially, all Small items are unallocated. Then while $\mathcal{B}_1 \neq \emptyset$ and $\text{Small}^- \neq \emptyset$, assign any item from Small^- to any agent that has a bundle from $\mathcal{B}_1$. Re-classify the bundles using equation (13) and remove the assigned item from Small^- after every assignment. This procedure ends when either $\mathcal{B}_1$ or Small^- becomes empty, or both. If in the end $\mathcal{B}_1 \neq \emptyset$, and we also have (a) $\mathcal{B}_4 \neq \emptyset$ and (b) $v(\mathcal{B}_3 \cup \mathcal{B}_4 \cup \text{Small}^+) < (1-\frac{\epsilon}{2})(|\mathcal{B}_3| + |\mathcal{B}_4|)$ then we call B^π *invalid*. All partitions of Big that are not invalid are called valid.

Lemma 4.2. *There exists an* MMS *allocation where the* Big *items are allocated according to a valid partition.*

Proof. Let A^π be an MMS allocation with the lowest value of $|\mathcal{B}_1| + |\mathcal{B}_4|$, and let B^π be its corresponding partition of Big. Suppose B^π is invalid. Then the agents with bundles from $\mathcal{B}_3$ or $\mathcal{B}_4$ can only receive items from $\mathcal{B}_3 \cup \mathcal{B}_4 \cup \text{Small}^+$ in A^π. From Lemma 2.2 and the definition of invalid partitions giving $v(\mathcal{B}_3 \cup \mathcal{B}_4 \cup \text{Small}^+) < (1-\frac{\epsilon}{2})(|B_3| + |B_4|)$, we have $\mu < (1-\frac{\epsilon}{2})$.

As both $\mathcal{B}_1, \mathcal{B}_4 \neq \emptyset$, consider any bundles B and B' respectively from $\mathcal{B}_1$ and $\mathcal{B}_4$. Let $v(B) = (1+x)$ and $v(B') = -y$ for some $x, y > 0$. In A^π, B' must be bundled with a set of items from Small^+, denoted as $\mathcal{S}^+$, of value at least $y+\mu$. We form two new bundles A_1 and A_2 of at least MMS value using B, B' and $\mathcal{S}^+$ as follows. First merge B and B' into one bundle (with value $1+x-y$). If $1+x-y \geq \mu$, call this bundle A_1 and add all the remaining items from $\mathcal{S}^+$ to A_2. Each bundle thus has value at least μ. Otherwise, if $1+x-y < \mu$, add items from $\mathcal{S}^+$ one by one, each time to the bundle with the lower value before adding the item. Let A_1 and A_2 be the resulting bundles after adding all items in $\mathcal{S}^+$. Without loss of generality, let $v(A_1) \geq v(A_2)$. As each item in $\mathcal{S}^+$ has value at most $\frac{\epsilon}{2}$ and is always added to the lower valued bundle, $v(A_1) - v(A_2) \leq \frac{\epsilon}{2}$. Thus,

$$v(A_1) + v(A_2) \geq 1 + x - y + y + \mu = 1 + x + \mu \text{ and } v(A_1) - v(A_2) \leq \tfrac{\epsilon}{2}$$
$$\implies v(A_2) \geq \tfrac{1}{2}(1 + x + \mu - \tfrac{\epsilon}{2}) > \tfrac{1}{2}(1 + \mu - \tfrac{\epsilon}{2}) > \mu \implies v(A_1) > \mu.$$

No item from Big is assigned to A_2. Thus, $A_2 \notin \mathcal{B}_1 \cup \mathcal{B}_4$, and A_1 and A_2 combined with the allocations of the remaining agents who did not get B or B' in A^π form an MMS allocation with a smaller value of $|\mathcal{B}_1| + |\mathcal{B}_4|$ than A^π, a contradiction. Thus, B^π is valid. □

4.3 Algorithm Bag-Fill

In this section we design the algorithm Bag-Fill (Algorithm 2) that generalizes algorithms in [GHS⁺18, GMT18, GT20] to the mixed setting. Bag-Fill (Algorithm 2) takes as input an MMS instance $(n, \mathcal{M}, v)$, and a partition of the Big items of $\mathcal{M}$, denoted by $B^\pi = [B_1, B_2, \ldots, B_n]$ such that they satisfy one of the two condition sets (14) or (15). It outputs an allocation of items $A^\pi = [A_1, \ldots, A_n]$ where $v(A_i) \geq 1 - \epsilon$, for all $i \in [n]$.

$$\begin{aligned} &v(\mathsf{Small}) + \sum_{k=1}^{n} v(B_k) \geq n. \\ &v(B_k) \leq 1 \;\; \forall k \in [n]. \\ &|v_j| < \epsilon \;\; \forall j \in \mathsf{Small}. \end{aligned} \tag{14}$$

$$\begin{aligned} &v(\mathsf{Small}) + \sum_{k=1}^{n} v(B_k) \geq n(1 - \tfrac{\epsilon}{2}). \\ &v(B_k) \leq 1 - \tfrac{\epsilon}{2} \;\; \forall k \in [n]. \\ &|v_j| < \tfrac{\epsilon}{2} \;\; \forall j \in \mathsf{Small}. \end{aligned} \tag{15}$$

Algorithm 2 works as follows. It has $n-1$ rounds. Each round starts with a bundle ('bag') from B^π. If the bag is valued at least $(1-\epsilon)$, then it is assigned to some agent. If not, we first add all the unallocated Small chores to this bag. Then one by one we add the unallocated goods from Small until it is valued at least $(1-\epsilon)$, and assign to some agent. After all rounds are done, in the last step, all remaining items from Small are added to the bag B_n. The next lemma proves the correctness of the algorithm.

ALGORITHM 2: Bag-filling to find $(1-\epsilon)$-MMS allocation of identical agents

Input : $(n, \mathcal{M}, v)$, Partition of Big $\subseteq \mathcal{M}$: $B^\pi = [B_1, B_2, \ldots, B_n]$. Input satisfies Condition set (14) or (15)
Output: $A^\pi = \{A_1, \ldots, A_n\}$ such that $v(A_i) \geq 1 - \epsilon$, $\forall i \in [n]$.

```
A^π ← ∅, Small ← M\Big
for k ∈ {1,...,n−1} do
    while v(B_k) < 1 − ϵ do
        j ← argmin_{j∈Small} v_j
        B_k ← B_k ∪ {j}, Small ← Small \ {j}
    A^π ← A^π ∪ {B_k}
B_n ← B_n ∪ Small,  A^π ← A^π ∪ {B_n}
return A
```

Lemma 4.3. *If an* MMS *instance with identical agents satisfies condition set* (14) *or* (15)*, then Algorithm 2 gives a* $(1-\epsilon)$*-Allocation.*

Proof. By induction on $k \in \{0, 1, 2, \cdots, n-1\}$, we prove that the value of each assigned bundle after k rounds is in $[1-\epsilon, 1]$ if the instance satisfies condition set (14), and in $[1-\epsilon, 1-\epsilon/2]$ if it satisfies condition set (15). The base case when $k = 0$ is trivial.

First consider the case when condition set (14) is satisfied. Assume the value of all bundles assigned to the first $k-1$ agents are in this range. Now $v(B_j) \leq 1$ for all $j \in \{k, .., n\}$. If $v(B_k) \geq 1-\epsilon$, we are done. If not, then while the value of B_k is less than $1-\epsilon$, the value of the unallocated items from Small is at least the value of all items minus that of all the allocated bundles and unallocated bags of Big items. This can be bounded as,

$$v(\mathcal{M}) - \sum_{i<k} v(A_i) - v(B_k) - \sum_{i>k} v(B_i) > n - (k-1) - (1-\epsilon) - (n-k) > \epsilon.$$

Hence, there is at least one unallocated Small good. Before adding the last Small good to A_k, its value was strictly less than $1-\epsilon$. Adding the last item increases the value by at most ϵ. Hence, the value of A_k is at most 1. Thus, $v(A_k) \in [1-\epsilon, 1]$, for all $k \in [n-1]$. As the total value of all items is at least n, and the total value of the $n-1$ assigned bundles is at most $n-1$, the last agent also gets a bundle of value at least 1.

Now suppose the instance satisfies condition set (15). In every round, while this is not true, the value of unallocated goods is at least $\epsilon/2$. Thus, there is at least one Small good. Finally, after assigning $n-1$ bundles, the total value remaining is at least $n(1-\epsilon/2) - (n-1)(1-\epsilon/2)$, hence the last bundle also has value at least $(1-\epsilon)$. □

4.4 The PTAS

We use the notions from the previous subsections and derive the PTAS, shown in Algorithm 3. The PTAS works as follows. It first enumerates all the partitions of Big. For each partition B^π, it first classifies the bundles into the $\mathcal{B}$-sets as per equation (13). If $\mathcal{B}_1$ is not empty, then add items from Small$^-$ to any bag in $\mathcal{B}_1$, re-defining the sets and removing the assigned item from Small$^-$ after each assignment. This process ends when either $\mathcal{B}_1 = \emptyset$ or Small$^- = \emptyset$. In the first case, condition set (14) of the Bag-Fill algorithm is satisfied, and we run Algorithm 2 (Line 8) and return its output.

Otherwise when $\mathcal{B}_1 \neq \emptyset$, if $|\mathcal{B}_4| = 0$, then reduce to the following goods manna α-MMS problem instance $(\mathcal{N}', \mathcal{M}', v')$. $\mathcal{N}'$ is the set of agents who received bundles from $\mathcal{B}_3$ or $\mathcal{B}_4$. $\mathcal{M}'$ has (a) Small$^+$, with each item having the same value in v' as in v, and (b) for each bundle $B \in \mathcal{B}_3$, $\mathcal{M}'$ has a new item b with value $v'_b = v(B)$. Run the PTAS from [JKV16] on $(\mathcal{N}', \mathcal{M}', v')$ to find a $(1-\epsilon)$-MMS allocation of $\mathcal{M}'$ among the $n' = n - (|\mathcal{B}_1| + |\mathcal{B}_2|)$ agents, and store its output in $\mathbb{A}$.

For the final case when $\mathcal{B}_1 \neq \emptyset$ and $|\mathcal{B}_4| > 0$, first check if the remaining unallocated goods and agents in $\mathcal{B}_3 \cup \mathcal{B}_4$ fulfill the condition set 15. If they do, apply the Bag-Fill and return the $(1-\epsilon)$-MMS allocation. If not, then B^π is invalid, hence discarded. After enumerating all the partitions, the algorithm returns the best allocation from $\mathbb{A}$.

Let us now discuss the analysis of the PTAS. To prove correctness when $\mathcal{B}_1 \neq \emptyset$ and $\mathcal{B}_4 = \emptyset$, we first show in Lemma 4.4 a relation between the MMS values of the given instance and the reduced goods manna instance. Let $A^{\pi*}$ be some MMS allocation, and $B^{\pi*} = \{B^*_1, B^*_2, \cdots, B^*_n\}$ be the allocation of Big items according to $A^{\pi*}$.

ALGORITHM 3: $(1-\epsilon)$-MMS allocation for identical agents with MMS ≥ 0

Input : $(n, \mathcal{M}, v)$ such that $v(\mathcal{M}) = n$, $\epsilon \in [0, 1]$
Output: $(1-\epsilon)$-MMS Allocation

$\mathbb{A} \leftarrow \emptyset$. $\Pi_n(\text{Big}) \leftarrow$ all partitions of Big into n sets.
for $B^\pi \in \Pi_n(\text{Big})$ **do**
  Define $\mathcal{B}$-sets as per equation (13).
  while $\mathcal{B}_1 \neq \emptyset$ *and* $\text{Small}^- \neq \emptyset$ **do**
    Remove any item j from Small^- and assign to any agent with a $\mathcal{B}_1$ bundle
    Re-define the $\mathcal{B}$-sets for the new allocation
  if $\mathcal{B}_1 = \emptyset$ **then**
    $A^\pi \leftarrow$ Bag-Fill($(\mathcal{N}, \mathcal{M}, v)$, $\mathcal{B}$-sets)
    return A^π
  $\mathcal{A}^{1,2} \leftarrow$ allocation of all bundles from $\mathcal{B}_1$ and $\mathcal{B}_2$ to distinct agents
  $\mathcal{N}' \leftarrow$ set of remaining agents, $\mathcal{M}' \leftarrow \cup_{B \in \mathcal{B}_3 \cup \mathcal{B}_4} B \cup \text{Small}^+$, $n' = |\mathcal{N}'|$
  if $\mathcal{B}_4 \neq \emptyset$ **then**
    if $v(\mathcal{M}') \geq n'(1 - \frac{\epsilon}{2})$ **then**
      $A^\pi \leftarrow \mathcal{A}^{1,2} \cup$ Bag-Fill($(\mathcal{N}', \mathcal{M}', v), \mathcal{B} - sets = \mathcal{B}_3 \cup \mathcal{B}_4$)
      return A^π
    else
      continue `// B^π is invalid`
  else
    $\mathcal{M}' \leftarrow \text{Small}^+$
    for $B \in \mathcal{B}_3$ **do**
      introduce a new good b with $v(b) = v(B)$; $\mathcal{M}' \leftarrow \mathcal{M}' \cup \{b\}$
    $A^\pi \leftarrow \mathcal{A}^{1,2} \cup (1-\epsilon)$-MMS allocation for $(\mathcal{N}', \mathcal{M}', v')$ using the algorithm in [JKV16]
    $\mathbb{A} \leftarrow \mathbb{A} \cup \{A^\pi\}$
return $\text{argmax}_{A^\pi \in \mathbb{A}} \min_{A_i \in A^\pi} v(A_i)$

Lemma 4.4. *If for $B^{\pi *}$, the subsequent allocation of* $\text{Big} \cup \text{Small}^-$ *in Algorithm 3 has $\mathcal{B}_4 = \emptyset$, then,*

$$\text{MMS}^n(\mathcal{M}) \leq_p \text{MMS}^{n-|\mathcal{B}_1|-|\mathcal{B}_2|}\Big(\bigcup_{B \in \mathcal{B}_3} B \cup \text{Small}^+\Big) .$$

Proof. We form an allocation of $\cup_{B \in \mathcal{B}_3} B \cup \text{Small}^+$ among $n - |\mathcal{B}_1| - |\mathcal{B}_2|$ agents with the smallest bundle's value at least $\text{MMS}^n(\mathcal{M})$, thus proving the lemma. Consider the allocation of Small in $A^{\pi *}$. Allocate the items from $\mathcal{M}' = \cup_{B \in \mathcal{B}_3} B \cup \text{Small}^+$ among the set $\mathcal{N}'$ of agents who have received bundles in $\mathcal{B}_3$, as they are allocated in $A^{\pi *}$. Call this allocation $A^{\pi \prime}$. Now the allocation $A^{\pi *}$ may also have some Small chores assigned to agents in $\mathcal{N}'$, but no other goods. The lowest valued bundle in $A^{\pi \prime}$ thus has value at least that of the lowest valued bundle in $A^{\pi *}$ (since no SMALL chore is added to these bundles in $A^{\pi \prime}$). The MMS value of agents in $\mathcal{N}'$, when partitioning $\mathcal{M}'$ among them, is at least that of the lowest valued bundle of $A^{\pi \prime}$, hence is at least $\text{MMS}^n(\mathcal{M})$. □

Next we state and prove the main theorem of this section.

Theorem 4.1. *Given an instance $(n, \mathcal{M}, v)$ with* MMS ≥ 0, *Algorithm 3 returns a* $(1-\epsilon)$-MMS *allocation in $O(m)$ time.*

Proof. First we prove the correctness of the algorithm. Note that no valid partition is discarded, as the procedure before deciding to discard a partition is exactly the procedure to determine if the partition is invalid. Consider a valid partition $B^{\pi*}$ corresponding to an MMS allocation $A^{\pi*}$, and its $\mathcal{B}$-sets as per (13). From Lemma 4.2, such a partition exists. After executing the while loop on Line 4, as every Small chore has absolute value at most $\epsilon/2$, upon adding the last chore before the value falls below 1, the value of every bundle to which a chore was added is still at least $1-\epsilon/2$. After this, one of the cases based on which conditions from $\mathcal{B}_1 = \emptyset$ and $\mathcal{B}_4 = \emptyset$ are true gets executed. In every case, there is some allocation generated, as the partition is valid.

If the Bag-Fill algorithm is called, then every agent gets a bundle of value $1-\epsilon$. As MMS ≤ 1, the allocation returned is $(1-\epsilon)$-MMS.

If the PTAS of [JKV16] is called, then first, the agents receiving bundles from $\mathcal{B}_1, \mathcal{B}_2$, by definition of these sets, have value at least $1-\epsilon \geq (1-\epsilon)$-MMS for their bundle. Also, as $B^{\pi*}$ corresponds to an MMS allocation, the MMS value for allocating the remaining items among the remaining agents, from Lemma 4.4, is at least the original MMS value. Hence, a $(1-\epsilon)$-MMS allocation of the goods manna instance, combined with the allocations to the agents with the $\mathcal{B}_1$ and $\mathcal{B}_2$ bundles, is $(1-\epsilon)$-MMS.

As $B^{\pi*}$ is considered when enumerating all the Big item partitions, this allocation will be stored in $\mathbb{A}$. Hence, the allocation returned has value at least $(1-\epsilon)$-MMS for the smallest valued bundle.

For running time, note that every iteration of the for loop first allocates all Small chores, then either runs a bag-filling algorithm which takes $O(m)$ time, discards the iteration, or runs the PTAS of [JKV16] which takes $O(2^{\tilde{O}(1/\epsilon)} n \log m) = o(2^{(1/\epsilon^2)} n \log m)$ time. In the worst case, every iteration takes $O(m + O(2^{1/\epsilon^2} n \log m))$ time. The for loop runs for $n^{|\mathrm{BIG}|}$ iterations, which from Lemma 4.1 is $O(n^{n/\tau\epsilon}) = 2^{O(n \log n/\tau\epsilon)}$. Hence, the total run time of the algorithm is $O(2^{n \log n/\tau\epsilon}(2^{1/\epsilon^2} n \log m + m)) = O(m)$ time. □

Acknowledgments. We would like to thank Prof. Jugal Garg for several valuable discussions.

REFERENCES

[ABL+16] Haris Aziz, Péter Biró, Jérôme Lang, Julien Lesca, and Jérôme Monnot. Optimal reallocation under additive and ordinal preferences. In *International Conference on Autonomous Agents & Multiagent Systems*, pages 402–410. ACM, 2016.

[ACIW19] Haris Aziz, Ioannis Caragiannis, Ayumi Igarashi, and Toby Walsh. Fair allocation of indivisible goods and chores. In *IJCAI*, pages 53–59, 2019.

[ACL19a] Haris Aziz, Hau Chan, and Bo Li. Maxmin share fair allocation of indivisible chores to asymmetric agents. In *International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '19*, pages 1787–1789, 2019.

[ACL19b] Haris Aziz, Hau Chan, and Bo Li. Weighted maxmin fair share allocation of indivisible chores. *arXiv preprint arXiv:1906.07602*, 2019.

[AEG+13] Nima Anari, Shayan Ehsani, Mohammad Ghodsi, Nima Haghpanah, Nicole Immorlica, Hamid Mahini, and Vahab S. Mirrokni. Equilibrium pricing with positive externalities. *Theor. Comput. Sci.*, 476:1–15, 2013.

[AGSS17] Nima Anari, Shayan Oveis Gharan, Amin Saberi, and Mohit Singh. Nash Social Welfare, Matrix Permanent, and Stable Polynomials. In *8th Innovations in Theoretical Computer Science Conference (ITCS)*, pages 1–12, 2017.

[AHMSH19] Haris Aziz, Xin Huang, Nicholas Mattei, and Erel Segal-Halevi. The constrained round robin algorithm for fair and efficient allocation. *arXiv preprint arXiv:1908.00161*, 2019.

[Ale20] Martin Aleksandrov. Jealousy-freeness and other common properties in fair division of mixed manna. *arXiv preprint arXiv:2004.11469*, 2020.

[AMGV18] Nima Anari, Tung Mai, Shayan Oveis Gharan, and Vijay V. Vazirani. Nash social welfare for indivisible items under separable, piecewise-linear concave utilities. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018*, pages 2274–2290, 2018.

[AMNS17] Georgios Amanatidis, Evangelos Markakis, Afshin Nikzad, and Amin Saberi. Approximation algorithms for computing maximin share allocations. *ACM Transactions on Algorithms (TALG)*, 13(4):52, 2017.

[AMS20] Haris Aziz, Hervé Moulin, and Fedor Sandomirskiy. A polynomial-time algorithm for computing a pareto optimal and almost proportional allocation. *Operations Research Letters*, 48(5):573–578, 2020.

[ARSW17] Haris Aziz, Gerhard Rauchecker, Guido Schryen, and Toby Walsh. Algorithms for max-min share fair allocation of indivisible chores. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[AW19] Martin Aleksandrov and Toby Walsh. Greedy algorithms for fair division of mixed manna. *CoRR*, abs/1911.11005, 2019.

[AW20a] Martin Aleksandrov and Toby Walsh. Two algorithms for additive and fair division of mixed manna. In *German Conference on Artificial Intelligence (Künstliche Intelligenz)*, pages 3–17. Springer, 2020.

[AW20b] Martin Aleksandrov and Toby Walsh. Two algorithms for additive and fair division of mixed manna, 2020.

[BB18] Arpita Biswas and Siddharth Barman. Fair division under cardinality constraints. In *IJCAI*, pages 91–97, 2018.

[BBKN18] Siddharth Barman, Arpita Biswas, Sanath Kumar Krishnamurthy, and Yadati Narahari. Groupwise maximin fair allocation of indivisible goods. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[BBKS20] Siddharth Barman, Umang Bhaskar, Anand Krishna, and Ranjani G Sundaram. Tight approximation algorithms for p-mean welfare under subadditive valuations. *arXiv preprint arXiv:2005.07370*, 2020.

[BGJ+19] Siddharth Barman, Ganesh Ghalme, Shweta Jain, Pooja Kulkarni, and Shivika Narang. Fair division of indivisible goods among strategic agents. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1811–1813, 2019.

[BILS19] Xiaohui Bei, Ayumi Igarashi, Xinhang Lu, and Warut Suksompong. Connected fair allocation of indivisible goods. *arXiv:1908.05433*, 2019.

[BKM17] Siddharth Barman and Sanath Kumar Krishna Murthy. Approximation algorithms for maximin fair division. In *Proceedings of the 2017 ACM Conference on Economics and Computation*, pages 647–664. ACM, 2017.

[BKV18] Siddharth Barman, Sanath Kumar Krishnamurthy, and Rohit Vaish. Finding fair and efficient allocations. In *Proceedings of the 2018 ACM Conference on Economics and Computation, Ithaca, NY, USA, June 18-22, 2018*, pages 557–574. ACM, 2018.

[BL16] Sylvain Bouveret and Michel Lemaître. Characterizing conflicts in fair division of indivisible goods using a scale of criteria. *Autonomous Agents and Multi-Agent Systems*, 30(2):259–290, 2016.

[BMR+13] Simina Brânzei, Tomasz P. Michalak, Talal Rahwan, Kate Larson, and Nicholas R. Jennings. Matchings with externalities and attitudes. In *International conference on Autonomous Agents and Multi-Agent Systems, AAMAS '13*, 2013.

[BMSY17] Anna Bogomolnaia, Hervé Moulin, Fedor Sandomirskiy, and Elena Yanovskaya. Competitive division of a mixed manna. *CoRR*, 2017.

[BMSY19] Anna Bogomolnaia, Hervé Moulin, Fedor Sandomirskiy, and Elena Yanovskaia. Dividing bads under additive utilities. *Soc. Choice Welf.*, 52(3):395–417, 2019.

[BT96] Steven J Brams and Alan D Taylor. *Fair Division: From cake-cutting to dispute resolution*. Cambridge University Press, 1996.

[Bud11] Eric Budish. The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes. *Journal of Political Economy*, 119(6):1061–1103, 2011.

[CCG+18] Yun Kuen Cheung, Bhaskar Chaudhuri, Jugal Garg, Naveen Garg, Martin Hoefer, and Kurt Mehlhorn. On fair division of indivisible items. In *FSTTCS*, 2018.

[CDG+17] Richard Cole, Nikhil R. Devanur, Vasilis Gkatzelis, Kamal Jain, Tung Mai, Vijay V. Vazirani, and Sadra Yazdanbod. Convex program duality, fisher markets, and nash social welfare. In *Proceedings of the 2017 ACM Conference on Economics and Computation, EC '17*, 2017.

[CFSV19] Vincent Conitzer, Rupert Freeman, Nisarg Shah, and Jennifer Wortman Vaughan. Group fairness for the allocation of indivisible goods. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI*, pages 1853–1860. AAAI Press, 2019.

[CG15] Richard Cole and Vasilis Gkatzelis. Approximating the nash social welfare with indivisible items. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015*, 2015.

[CGM20] Bhaskar Ray Chaudhury, Jugal Garg, and Ruta Mehta. Fair and efficient allocations under subadditive valuations. *arXiv preprint arXiv:2005.06511*, 2020.

[CGMM20] Bhaskar Ray Chaudhury, Jugal Garg, Peter McGlaughlin, and Ruta Mehta. Dividing bads is harder than dividing goods: On the complexity of fair and efficient division of chores. 2020.

[CKMS20] Bhaskar Ray Chaudhury, Telikepalli Kavitha, Kurt Mehlhorn, and Alkmini Sgouritsa. A little charity guarantees almost envy-freeness. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2658–2672. SIAM, 2020.

[EPT07] Raul Etkin, Abhay Parekh, and David Tse. Spectrum sharing for unlicensed bands. *IEEE Journal on selected areas in communications*, 25(3):517–528, 2007.

[FGH+19] Alireza Farhadi, Mohammad Ghodsi, Mohammad Taghi Hajiaghayi, Sébastien Lahaie, David M. Pennock, Masoud Seddighin, Saeed Seddighin, and Hadi Yami. Fair allocation of indivisible goods to asymmetric agents. *J. Artif. Intell. Res.*, 64:1–20, 2019.

[GHM19] Jugal Garg, Martin Hoefer, and Kurt Mehlhorn. Approximating the Nash Social Welfare with budget-additive valuations. arxiv:1707.04428; Preliminary version appeared in the proceedings of SODA 2018, 2019.

[GHS+18] Mohammad Ghodsi, Mohammadtaghi Hajiaghayi, Masoud Seddighin, Saeed Seddighin, and Hadi Yami. Fair allocation of indivisible goods: Improvements and generalizations. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, EC '18, 2018.

[GKK20] Jugal Garg, Pooja Kulkarni, and Rucha Kulkarni. Approximating nash social welfare under submodular valuations through (un) matchings. In *Proceedings of the fourteenth annual ACM-SIAM symposium on discrete algorithms*, pages 2673–2687. SIAM, 2020.

[GM19] Laurent Gourvès and Jérôme Monnot. On maximin share allocations in matroids. *Theor. Comput. Sci.*, 754:50–64, 2019.

[GM20] Jugal Garg and Peter McGlaughlin. Computing competitive equilibria with mixed manna. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, pages 420–428, 2020.

[GMT18] Jugal Garg, Peter McGlaughlin, and Setareh Taki. Approximating maximin share allocations. In *2nd Symposium on Simplicity in Algorithms (SOSA 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

[GT20] Jugal Garg and Setareh Taki. An improved approximation algorithm for maximin shares. In *Proceedings of the 21st ACM Conference on Economics and Computation*, EC '20, page 379–380, 2020.

[HL19] Xin Huang and Pinyan Lu. An algorithmic framework for approximating maximin share allocation of chores. *CoRR*, abs/1907.04505, 2019.

[JKV16] Klaus Jansen, Kim-Manuel Klein, and José Verschae. Closing the gap for makespan scheduling via sparsification techniques. In *43rd International Colloquium on Automata, Languages, and Programming, ICALP*, volume 55, pages 72:1–72:13, 2016.

[KPW16] David Kurokawa, Ariel D Procaccia, and Junxing Wang. When can the maximin share guarantee be guaranteed? In *AAAI*, volume 16, pages 523–529, 2016.

[KPW18] David Kurokawa, Ariel D. Procaccia, and Junxing Wang. Fair enough: Guaranteeing approximate maximin shares. *J. ACM*, 65(2):8:1–8:27, 2018.

[LMMS04] Richard J. Lipton, Evangelos Markakis, Elchanan Mossel, and Amin Saberi. On approximately fair allocations of indivisible goods. In *Proceedings 5th ACM Conference on Electronic Commerce (EC-2004)*, 2004.

[LSZ19] Yin Tat Lee, Zhao Song, and Qiuyi Zhang. Solving empirical risk minimization in the current matrix multiplication time. In *Conference on Learning Theory*, pages 2140–2157, 2019.

[LT19] Zbigniew Lonc and Miroslaw Truszczynski. Maximin share allocations on cycles. *arXiv:1905.03038*, 2019.

[LV18] Zhentao Li and Adrian Vetta. The fair division of hereditary set systems. In *International Conference on Web and Internet Economics*, pages 297–311. Springer, 2018.

[MG20] Peter McGlaughlin and Jugal Garg. Improving nash social welfare approximations. volume 68, pages 225–245, 2020.

[Mou04] Hervé Moulin. *Fair division and collective welfare.* MIT press, 2004.

[PW14] Ariel D Procaccia and Junxing Wang. Fair enough: Guaranteeing approximate maximin shares. In *Proceedings of the fifteenth ACM conference on Economics and computation*, pages 675–692. ACM, 2014.

[SSH19] Fedor Sandomirskiy and Erel Segal-Halevi. Fair division with minimal sharing. *arXiv preprint arXiv:1908.01669*, 2019.

[Ste48] Hugo Steinhaus. The problem of fair division. *Econometrica*, 16:101–104, 1948.

[Vos02] Thomas Vossen. *Fair allocation concepts in air traffic management.* PhD thesis, PhD thesis, Supervisor: MO Ball, University of Martyland, College Park, Md, 2002.

[Woe97] Gerhard J Woeginger. A polynomial-time approximation scheme for maximizing the minimum machine completion time. *Operations Research Letters*, 20(4):149–154, 1997.

[ZP20] David Zeng and Alexandros Psomas. Fairness-efficiency tradeoffs in dynamic fair division. EC '20, page 911–912, New York, NY, USA, 2020. Association for Computing Machinery.

A MISSING PROOFS

A.1 Section 2

Lemma 2.1. *$v_i(\mathcal{M}) \geq 0$ iff $\mathsf{MMS}_i \geq 0$.*

Proof. If the sum of valuations of all items $v_i(\mathcal{M})$ is negative, there can be no allocation where every bundle has non-negative valuation. Hence, MMS_i is negative. If the sum of valuations is positive, then adding all items to one bundle and no item in other bundles makes the least-valued bundle have zero value. Thus, in this case, $\mathsf{MMS}_i \geq 0$. □

Lemma 2.2. $\mathsf{MMS}_i \leq v_i(\mathcal{M})/|\mathcal{N}|$ *for all* $i \in \mathcal{N}$.

Proof. If $\mathsf{MMS}_i > v_i(\mathcal{M})/n$, it implies that there exists a partition of items in $\Pi_n(\mathcal{M})$ where all bundles have value greater than $v_i(\mathcal{M})/n$. Therefore, $v_i(\mathcal{M}) \geq n.\mathsf{MMS}_i > n.\frac{v_i(\mathcal{M})}{n} = v_i(\mathcal{M})$, which is a contradiction. □

Lemma 2.3. *[Scale Invariance]* α-MMS + PO *allocations for the instances* $(\mathcal{N}, \mathcal{M}, (v_i)_{i\in\mathcal{N}})$ *and* $(\mathcal{N}, \mathcal{M}, (v'_i)_{i\in\mathcal{N}})$ *are the same when for all i, j,* $v'_{ij} = c_i \cdot v_{ij}$ *for some constants* $c_i > 0$.

Proof. For any agent i, the value of any bundle of items $\mathcal{S}$ according to the two valuation function are related as $v'_i(\mathcal{S}) = c_i \cdot v_i(\mathcal{S})$. Thus, by definition of MMS, her MMS values according to the two valuation functions are also related as $\mathsf{MMS}'_i = c_i \cdot \mathsf{MMS}_i$, where MMS'_i is agent i's MMS value according to v'_i.

This implies that a set of items has α-MMS value for i according to $(v_i)_{i\in\mathcal{N}}$ if and only if it has α-MMS value according to $(v'_i)_{i\in\mathcal{N}}$. Hence all α-MMS allocations according to valuations $(v_i)_{i\in\mathcal{N}}$ are also α-MMS according to $(v'_i)_{i\in\mathcal{N}}$ and vice versa.

The α-MMS+PO allocation according to $(v_i)_{i\in\mathcal{N}}$, say $\mathcal{A}$, is α-MMS and must also be PO according to $(v'_i)_{i\in\mathcal{N}}$, as otherwise the Pareto dominating allocation will Pareto dominate $\mathcal{A}$ according to $(v_i)_{i\in\mathcal{N}}$ too. □

A.2 Section 3

Claim 3.1. *For the approximate* MMS *values* $\tilde{\mu}_i$, *we have, if* $\mu_i > 0$, *then* $\tilde{\mu}_i \in [(1-\epsilon/2)\mu_i, \mu_i]$, *if* $\mu_i = 0$ *then* $\tilde{\mu}_i = 0$ *and if* $\mu_i < 0$, *then* $\tilde{\mu}_i \in [\mu_i/(1-\epsilon/2), \mu_i]$.

Proof. We know from condition 2 of the α-MMS + PO problem that $v_i(\mathcal{M}) \geq \tau \cdot \min\{v_i^+, v_i^-\}$. After scaling, we have $|v_i(\mathcal{M})| = n$. For agents where $v_i(\mathcal{M}) \geq 0$, $n \geq \tau \cdot v_i^- \Rightarrow v_i^- \leq n/\tau = O(n)$. Also, $n = v_i^+ - v_i^- \Rightarrow v_i^+ \leq n(1+1/\tau) = O(n)$. Analogously we prove the claim when $v_i(\mathcal{M}) < 0$. □

Lemma 3.2. *The number of big items, i.e.,* $|\mathsf{BIG}| \leq O(n^3/\epsilon)$.

Proof. Since $\mathsf{BIG} = \cup_{i\in\mathcal{N}}\mathsf{BIG}_i^+ \cup_{i\in\mathcal{N}} \mathsf{BIG}_i^-$, to prove the lemma it suffices to show that the number of BIG goods and chores of every agent $i \in \mathcal{N}$ is $|\mathsf{BIG}_i^+|, |\mathsf{BIG}_i^-| \leq O(n^2/\epsilon)$. Fix an agent $i \in \mathcal{N}$. First we will show bound on $|\mathsf{BIG}_i^+|$.

Case 1: $\tilde{\mu}_i \geq 0$.

$$\text{If } \tilde{\mu}_i \geq 1/3,\ |\mathsf{BIG}_i^+| \leq |\{j \in \mathcal{M} : v_{ij} > \epsilon/(6n)\}| \leq \frac{v_i^+}{\epsilon/(6n)} \leq \frac{6n}{\epsilon}O(n) = O(n^2/\epsilon) \tag{16}$$

The last inequality follows by Claim 3.2. Otherwise, if $\tilde{\mu}_i < 1/3$, then $\mu_i \leq \tilde{\mu}_i/(1-\epsilon/2) < 1/(3-3\epsilon/2) < 2/3$. Divide BIG_i^+ into two sets as follows.

$$\mathsf{BIG}_i^+ = \{j : v_{ij} > \epsilon/(6n)\} \cup \{j : \epsilon\tilde{\mu}_i/(2n) < v_{ij} \leq \epsilon/(6n)\}. \tag{17}$$

Let us call the first set in Equation (17) LARGE and the second set MEDIUM. Similarly as for the case of $\tilde{\mu}_i \geq 1/3$, we can prove the size of LARGE is at most $O(n^2/\epsilon)$. We now prove that the number of items in MEDIUM is at most $\frac{2n(n-1)}{(1-\epsilon/2)\epsilon} + (n-2)$. We show that if this is not true then there is a partition of all the items where all parts have value strictly more than μ_i for agent i which is a contradicts that μ_i being her MMS value. The partition is as follows. Add all items except the goods from MEDIUM to the first bundle. If i's value for this bundle is more than 1, divide MEDIUM to make $n-1$ bundles with at least $2n/((1-\epsilon/2)\epsilon)+1$ items in each. Then all the remaining bundles have value at least,

$$\left(\frac{2n}{(1-\epsilon/2)\epsilon}+1\right)\left(\frac{\epsilon}{2n}\right)\tilde{\mu}_i > \frac{\tilde{\mu}_i}{(1-\epsilon/2)} \geq \mu_i.$$

If the value of the first bundle is less than 1 for i, then we add enough goods from MEDIUM to each bundle one by one (first bundles and all remaining empty bundles) so that their value is at least $2/3 > \mu_i$. Since every item in MEDIUM has value at most $\epsilon/(6n)$, the value of the each bundle is less than $2/3$ before adding the last item and less than $2/3 + \epsilon/(6n) < 1$ later. As each bundle's value is at most 1 and $v(\mathcal{M}) = n$, there are enough items to make $(n-1)$ bundles, each of value at least $2/3$ which is greater than μ_i. This is a contradiction to definition of MMS_i.

Therefore, $|\mathsf{MEDIUM}| \leq \frac{2n(n-1)}{(1-\epsilon/2)\epsilon} + (n-2) = O(n^2/\epsilon)$. Hence $|\mathsf{BIG}_i^+| = |\mathsf{LARGE}| + |\mathsf{MEDIUM}| = O(n^2/\epsilon)$, for any i with $\tilde{\mu}_i \geq 0$.

Case 2: $\tilde{\mu}_i < 0$. Then by the definition of a BIG good for this case, $|\mathsf{BIG}_i^+| \leq v_i^+/(\epsilon/(2n)) = (2n/\epsilon)O(n) = O(n^2/\epsilon)$.

Next we show the bound on $|\mathsf{BIG}_i^-|$. By definition of a BIG chore, $|\mathsf{BIG}_i^-| \leq 2n \cdot v_i^-/\epsilon = O(n^2/\epsilon)$, as from Claim 3.2 we have $v_i^- \leq O(n)$. □

Lemma 3.3. *The allocation graph of* any *LP solution x can be made acyclic in such a way that in the allocation corresponding to the new graph, say $x' = [x'_{ij}]_{i\in\mathcal{N},j\in\mathsf{SMALL}}$, every agent receives a bundle of the same or better value as in x.*

Proof. First we show how to eliminate one cycle, say C, in the allocation graph of x. That is, we define a new allocation x', that removes one cycle without reducing the value of any agent. Let there be k agents and k items in C, with the edges as,

$$a^1 - o^1 - a^2 - o^2 - \cdots - o^{i-1} - a^i - o^i - a^{(i+1)} \cdots - o^{k-1} - a^k - o^k - a^1.$$

Each agent a^i is partially assigned items o^i and o^{i-1} (by setting $0 \equiv k$) and each item o^i is partially assigned to agents a^i and a^{i+1} (by setting $k+1 \equiv 1$). Without loss of generality, we may assume that items in C are solely considered as either a good or a chore by both agents sharing them, otherwise, we can break the cycle by allocating the share of the other agent for this item to the one who considers it as a good. We call an item a good if both the agents sharing it consider it so, else a chore.

First we argue the case when there is at least one good. Without loss of generality we assume o^k is a good. Let $X^C = [x_{11}, x_{21}, x_{22}, \ldots, x_{k(k-1)}, x_{kk}, x_{1k}]$ be the allocation vector of cycle C. Also let $\tilde{V}^C = [\tilde{v}_{11}, \tilde{v}_{21}, \tilde{v}_{22}, \ldots, \tilde{v}_{k(k-1)}, \tilde{v}_{kk}, \tilde{v}_{1k}]$ be the vector representing the *scaled* values of the agents for the items assigned to them in C, defined as,

$$\tilde{v}_{ij} = \begin{cases} v_{ij} & \text{if } i = 1 \\ v_{ij}\left(\frac{v_{(i-1)(i-1)}}{v_{i(i-1)}}\right) & \text{otherwise.} \end{cases}$$

In $\tilde{V}^C$ the valuations of the agents are scaled in a way so that agents sharing an item in C have the same value for that item (except for item k). Without loss of generality we assume $\tilde{v}_{1k} \leq \tilde{v}_{kk}$. Let $U^C = [u_{11}, u_{21}, u_{22}, \ldots, u_{k(k-1)}, u_{kk}, u_{1k}]$ be the utility vector of C where $u_{ij} = \tilde{v}_{ij}x_{ij}$. Let δ be the minimum of smallest positive $u_{ij}, i \neq j$ (even indexes of U^C) and smallest $|u_{ii}|, u_{ii} < 0$ (odd indexes of U^C). Define

$$u'_{ij} := \begin{cases} u_{ij} - \delta & \text{if } i \neq j \\ u_{ij} + \delta & \text{otherwise.} \end{cases}$$

Then the desired x' is defined as,

$$x'_{ij} := \begin{cases} u'_{ij}/\tilde{v}_{ij} & \text{if } i, j \in C \\ x_{ij} & \text{otherwise.} \end{cases}$$

By choice of δ, at least one x'_{ij} with $x_{ij} > 0$ will be 0 and no new edge is added to the allocation graph so the cycle C is removed. We need to show that the new x'_{ij}'s present a feasible allocation. By choice of δ, we can see that $u'_{ij} \geq 0$ when j is a good for agents sharing it in C, $u'_{ij} \leq 0$ otherwise. For all agents a^i in C, $u_{i(i-1)} + u_{ii} = u'_{i(i-1)} + u'_{ii}$ (by setting $1 - 1 = k$ for agent a^1) so each agent will get the same utility before removing the cycle. Also, we have $u_{ii} + u_{i(i+1)} = u'_{ii} + u'_{i(i+1)}$ and since for all items $O^i, i \in [k-1]$, $\tilde{v}_{ii} = \tilde{v}_{i(i+1)}$ we have $x_{ii} + x_{i(i+1)} = x'_{ii} + x'_{i(i+1)}$. For item k we have,

$$\begin{aligned} u'_{kk} = u_{kk} + \delta &\implies x'_{kk} = x_{kk} + \frac{\delta}{\tilde{v}_{kk}} \\ u'_{1k} = u_{1k} - \delta &\implies x'_{1k} = x_{1k} - \frac{\delta}{\tilde{v}_{1k}} \\ \implies x'_{kk} + x'_{1k} \leq x_{kk} + x_{1k} \;. \end{aligned}$$

The last inequality hold because $\tilde{v}_{1k} \leq \tilde{v}_{kk}$. Therefore, all agents receive the same utility in the new allocation. But there may be an extra amount of good k available; we assign it to the agent who has the highest share of good k.

If all items in C are chores, we define $\tilde{V}$ and U similarly as for the previous case. Without loss of generality, we assume $\tilde{v}_{1k} \leq \tilde{v}_{kk}$ and we choose δ to be the smallest $|u_{ii}|, u_{ii} < 0$ (odd indexes of U^C). With the same analysis we get $u_{ii} + u_{i(i+1)} = u'_{ii} + u'_{i(i+1)}$ for all agents i, $x_{ii} + x_{i(i+1)} = x'_{ii} + x'_{i(i+1)}$ for items $i \neq k$ and $x'_{kk} + x'_{1k} \geq x_{kk} + x_{1k}$. Therefore, agents get the same utility with an extra amount of chore k assigned to some agent. We improve the utility of the agent who gets this share of chore k by reducing her share from chore k by making, $\sum_{i \in \mathcal{N}} x_{ik} = 1$.

We repeat this process for every cycle, removing at least one edge with every removal. Hence, in polynomial time, we get an acyclic allocation graph.

□

Lemma 3.4. *The number of shared items in any acyclic allocation graph is at most $n - 1$.*

Proof. Suppose there are k shared goods. Consider the subgraph of the allocation graph with the $n + k$ nodes corresponding to all the buyers and only the shared goods. As this graph is acyclic, there are at most $n + k - 1$ edges. Further, each item is shared, meaning there are at least two edges incident to each node representing a good. Thus, there are at least $2k$ edges. The inequality $n + k - 1 \geq 2k$ is satisfied only when $k \leq n - 1$, hence there are at most $n - 1$ shared goods. □

Corollary 3.1. *If an α-MMS allocation exists, Algorithm 1 returns an $(\alpha - \epsilon)^+$-MMS allocation.*

Proof. If an α-MMS allocation exists, then for the partition of BIG corresponding to this allocation, say $B^\pi = [B_1 \cdots, B_n]$, there is an integral allocation of SMALL where every agent i gets value $\alpha \cdot \tilde{\mu}_i - v_i(B_i) \geq c_i$ from SMALL. Thus, the LP will have a (fractional) solution. From Lemma 3.6, the resulting allocation obtained by rounding the LP solution is $(\alpha - \epsilon)^+$-MMS. □

THEOREM 3.3. *Given an instance $(\mathcal{N}, \mathcal{M}, (v_i)_{i\in\mathcal{N}})$ and constants $\alpha, \epsilon, \gamma > 0$, that is, an instance of the α-MMS + PO problem, Algorithm 1 returns an $(\alpha - \epsilon)^+$-MMS + γ-PO allocation if an α-MMS allocation exists, else reports it does not exist, in time $O(2^{O((n^3 \log n)/\min\{\epsilon^2,\gamma^2/2\})}m^3)$ where $n = |\mathcal{N}|$ is a constant. Thus, it is a* PTAS.

PROOF. From Corollaries 3.1 and 3.2 the correctness of Algorithm 1 follows. Next we analyze the running time.

The time to compute the approximate MMS values is $O(n \cdot 2^{(n \log n)/\epsilon}(2^{1/\epsilon^2} n \log m + m))$, from the proofs of Theorems 4.1 and C.1. Since $|\mathsf{BIG}| \leq O(n^3/\tau\epsilon)$ by Lemma 3.2, the number of iterations in the for loop enumerating all the allocations of the BIG items is $O(2^{O((n^3 \log n)/\epsilon)})$. Note that we re-define ϵ as $\min\{\epsilon, \frac{\alpha\gamma}{(1+\gamma)}\} \geq \min\{\epsilon, \frac{\gamma^2}{2}\} =: \zeta$, thus $|\mathsf{BIG}| \leq O(n^3/\zeta)$. Each iteration solves an LP of mn variables and $O(mn)$ constraints, hence takes time some polynomial function in (m, n) less than $O((mn)^3)$ [LSZ19]. Finding a cycle in the allocation graph requires time linear in the number of edges, at most $O(mn)$. Eliminating the cycle requires time $O(mn)$, and deletes at least one edge. Repeating the process until the graph is acyclic takes at most $O(mn)$ iterations, hence the making the allocation acyclic and rounding it steps take time at most $O(m^2n^2)$. Hence the total time for the algorithm in the worst case is,

$$O(n \cdot 2^{n \log n/\epsilon}(2^{1/\epsilon^2} n \log m + m)) + O(2^{O(n^3 \log n/\zeta)}m^3n^3 + m^2n^2) \leq O(2^{O((n^3 \log n)/\min\{\epsilon^2,\zeta\})}m^3),$$
$$= O(2^{O((n^3 \log n)/\min\{\epsilon^2,\gamma^2/2\})}m^3) = O(m^3),$$

as n, α, γ and ϵ are constant. □

B NON-EXISTENCE OF α-MMS ALLOCATIONS

In this section, we show an instance for which there is no α-MMS allocation for any $\alpha > 0$. Our instance is a modification of the instance in [KPW16] that shows that an MMS allocation in a goods only manna does not always exist. We take their exact instance, and add three chores to $\mathcal{M}$, each of absolute value equal to a small constant less than the agent's MMS values. For completeness, we discuss all details of the instance.

Let $\mathcal{N} = \{1, 2, 3\}$, $\mathcal{M}^+ = \{(j, k) : j \in [3], k \in [4]\}$, $\mathcal{M}^- = \{(1), (2), (3)\}$, and $\mathcal{M} = \mathcal{M}^+ \cup \mathcal{M}^-$ respectively be the set of agents, goods, chores, and all items. In order to define the valuations of the agents for each of these items, we first define matrices $O, E^{(1)}, E^{(2)}$, and $E^{(3)}$ as follows.

$$O = \begin{bmatrix} 17 & 25 & 12 & 1 \\ 2 & 22 & 3 & 28 \\ 11 & 0 & 21 & 23 \end{bmatrix}$$

$$E^{(1)} = \begin{bmatrix} 3 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad E^{(2)} = \begin{bmatrix} 3 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix} \qquad E^{(3)} = \begin{bmatrix} 3 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}.$$

The valuation of each agent i for each good (j, k) is, $v_i(\{(j, k)\}) = 10^6 + 10^3 \cdot O_{jk} + E^{(i)}_{jk}$, and their value for each chore is -4054999.75.

From [KPW16], every agent can divide all the goods in this instance into three bundles of value 4055000 each. Adding one chore to each of these makes every bundle's value 0.25. It can be verified that the average value of all items is 0.25 for every agent. As MMS cannot be higher than the average, the above allocation shows that every agent's MMS value is 0.25. [KPW16] also show that there is no allocation of the goods were all agents get at least 4055000, and that the sum of any 3 goods is less than 4055000. As the values of goods are integers, every agent must get at least 4 goods for every chore in order to receive a positive valued bundle. If every agent is to get a positive valued bundle, the agent receiving less than 4055000 from the goods must not receive any chore, and must get at least one good. But then there are 3 chores and at most 11 goods remaining to be allotted. Hence, at least one agent will receive a negative valued bundle. Therefore, there is no allocation that can guarantee every agent a positive valued bundle, and the best α for which an α-MMS allocation exists is at most zero.

C COMPUTING MMS WHEN MMS < 0

In this section we introduce the algorithm that finds a $(1-\epsilon)$-MMS allocation of an agent with MMS < 0 for an instance $(n, \mathcal{M}, v)$ and a constant $\epsilon > 0$, or equivalently, a $(1-\epsilon)$-MMS allocation of $(\mathcal{N}, \mathcal{M}, (v_i)_{i\in\mathcal{N}})$ when there are identical agents with valuation function v (Algorithm 4). From Lemmas 2.2 and the normalization $v(\mathcal{M}) = -n$, we have MMS ≤ -1.

From Definition 2.1, a $(1-\epsilon)$-MMS allocation gives each agent a bundle with value at least $(1/(1-\epsilon))$MMS. Let $\sigma := \frac{1}{1-\epsilon} - 1$. Algorithm 4 obtains an allocation where each agent gets a bundle of value at least $(1+\sigma)$MMS $= (1/(1-\epsilon))$MMS. The high level idea of the algorithm is as follows. First we scale the valuations so that $v(\mathcal{M}) = -n$, and classify items as Big or Small. Then similarly as in Algorithm 3, we enumerate all partitions of Big. While there are unallocated Small goods, we add them one by one to the bundle with the least value. Once all the Small goods are exhausted, we iteratively add Small chores to the bundle with the highest value.

ALGORITHM 4: $(1-\epsilon)$-MMS Allocation for identical agents with MMS < 0

Input : $(n, \mathcal{M}, v)$, a constant ϵ
Output: $(1-\epsilon)$-MMS Allocation

Normalize the valuations so that $v(\mathcal{M}) = -n$.
$\sigma \leftarrow \frac{1}{1-\epsilon} - 1$, $\mathbb{A} \leftarrow \emptyset$
Big $:= \{j \in X : |v_j| \geq \sigma\}$, Small $:= \mathcal{M}\backslash$Big, Small$^+$ = Small $\cap \mathcal{M}^+$, Small$^-$ = Small $\cap \mathcal{M}^-$
for $\mathcal{B} \in \Pi_n$(Big) **do**
  while Small$^+ \neq \emptyset$ **do**
    add any Small good to a bundle with the lowest value
  while $X \neq \emptyset$ **do**
    add any Small chore to a bundle with the highest value
  store the allocation to a set $\mathbb{A}$
return $\text{argmax}_{A\in\mathbb{A}} \min_{A_i\in A} v(A_i)$ `// return allocation with highest maximin value`

Theorem C.1. *Algorithm 4 gives a* $(1-\epsilon)$-MMS *allocation when* MMS < 0 *in* $O(m)$ *time.*

Proof. We first prove a helpful lower bound on the value of all Small goods. Let $B^{\pi*}$ be a partition of the Big items corresponding to an MMS allocation. There are enough Small goods to add to each part in $B^{\pi*}$ so that every part has at least MMS value. Specifically, for the set $\mathcal{S} = \{B \in B^{\pi*} : v(B) < \text{MMS}\}$ we have,

$$v(\text{Small}^+) \geq \text{MMS} \cdot |\mathcal{S}| - v(B^{\pi*}) \tag{18}$$

Now, let $A^\pi = \{A_1, \dots, A_n\}$ be the output of Algorithm 4. Suppose for contradiction there exists some $A_i \in A$ such that $v(A_i) < (\frac{1}{1-\epsilon})\mathsf{MMS} = (1+\sigma)\mathsf{MMS}$. Consider each $A_k \supseteq B_k$ with $B_k \in \mathcal{S}$. Note that the algorithm adds Small goods to the bundle with the least value. Because of A_i, before adding the last Small good to any bundle, its value is less than $(1+\sigma)\mathsf{MMS}$. The last good added has value at most σ. Therefore, all the A_ks have value at most MMS. From, (18) and the fact that the algorithm adds goods to the least valued bundle, we have,

$$v(\mathsf{Small}^+ \setminus (\bigcup_{k\in[n]} A_k)) \geq \sum_{B\in\mathcal{S}} \mathsf{MMS} - v(B) + \sigma = \sigma, \tag{19}$$

which is a contradiction.

Now we prove that after adding the Small chores the value of all the bundles is at least $(1+\sigma)\mathsf{MMS}$. This is true because while there exists an unallocated chore, the value of the highest valued bundle is greater than -1, because $v(\mathcal{M}) = -n$. Adding a chore to such bundle will decrease the value by at most σ. Therefore, the value of such bundle is at least $-(1+\sigma) \geq (1+\sigma)\mathsf{MMS}$. By definition of σ, $(1+\sigma) = 1/(1-\epsilon)$.

Finally, $|\mathsf{BIG}| = O(n/\sigma) = O(n/\epsilon)$, from the definition of BIG and σ. As every iteration corresponding to a partition of BIG takes $O(m)$ time, Algorithm 4 runs for $O(m \cdot 2^{O(n \log n/\epsilon)}) = O(m)$ time. □

D HARDNESS OF APPROXIMATION

The α-MMS + PO problem makes two assumptions. First, the number of agents is assumed to be a constant. Second, the sum of absolute values of all the items for every agent is assumed to be at least τ times the minimum of this sum for the goods and the chores, for some constant $\tau > 0$. In this section we show that relaxing either of these two assumptions makes the α-MMS problem NP-hard for any $\alpha \in (0, 1]$, even when agents are identical.

When agents are identical, the allocation that decides the MMS value of the agents is also an MMS allocation for the instance. Thus, for $\alpha = 1$, the GENERAL α-MMS problem should return an MMS allocation. Furthermore, given $v(\mathcal{M}) > 0$, we are guaranteed to have $\mathsf{MMS} \geq 0$ due to Lemma 2.1. However next we show that when either assumption of problem α-MMS is dropped, *deciding* if the inequality is indeed strict is NP-hard.

We separate Theorem 3.1 as two NP-hardness results in Theorems D.1 and D.2. To prove both, we reduce from the known NP-hard PARTITION problem.

PARTITION **Problem.** Given a set of non-negative integers $E = \{e_1, \dots, e_m\}$, output YES if there exists a division of the elements into two sets of equal weight, otherwise output NO.

Theorem D.1. *Given an instance $(n, \mathcal{M}, v)$ with constantly many (two) identical agents and $v(\mathcal{M}) > 0$, checking if MMS > 0 is NP-hard.*

Proof. We reduce an instance of PARTITION to an MMS instance $(n, \mathcal{M}, v)$ with two identical agents. Let $\mathcal{N} = \{1, 2\}$. $\mathcal{M} = [m+2]$, where the first m items are goods and the last two are chores. The valuation function v is defined as follows, where $\beta = 1/4$.

$$v_j = \begin{cases} e_j, & \forall\ j \in [m] \\ -(\sum_i e_i/2) + \beta, & j \in \{m+1, m+2\}. \end{cases}$$

That is, the goods correspond to PARTITION elements, and have the same value as the weight of the element, and the chores are β more than the negated weight of each set in an equal distribution of the elements. Note that, (a) the trivial partition where all items are in the same bundle has the smaller bundle valued zero, and (b) the average of values of all items is β, and MMS cannot be higher than the average (Lemma 2.2). Hence, $0 \le \text{MMS} \le \beta$.

We prove the correctness of the reduction in the following two claims.

Claim D.1. PARTITION *has a solution* $\Rightarrow \text{MMS} \ge \beta$.

Proof. Divide the goods into two bundles as per the PARTITION solution, and add one chore to each set. This gives us two bundles of equal value β, implying that $\text{MMS} \ge \beta$. □

Claim D.2. $\text{MMS} > 0 \Rightarrow$ PARTITION *has a solution.*

Proof. We prove the contrapositive by contradiction. Suppose PARTITION does not have a solution. but $\text{MMS} > 0$ for the instance $(n, \mathcal{M}, v)$. Let $A^\pi = (A_1, A_2)$ be the allocation achieving the MMS value, and let $u_1 = v(A_1)$ and $u_2 = v(A_2)$. Then we have $u_1, u_2 > 0$.

First we prove that both the chores cannot be in the same bundle. If they are, and if all goods are not in this bundle, then the value of the bundle with chores is at most the sum of all except the smallest good. This is $(-\sum_i e_i + 2\beta) + (\sum_i e_i - \min_i e_i) \le 1/4 - 1 < 0$. If every good and chore is in the same bundle, the value of the other bundle is 0. But $v_1 > 0$, hence the chores are in separate bundles.

But then the value of the goods in each bundle is at least the total value minus the chore's value, i.e., for $i = 1, 2$, $v(A_i \cap \mathcal{M}^+) = u_i - (-\frac{1}{2}\sum_{i\in[m]} e_i + \beta) \ge \text{MMS} - \beta + \frac{1}{2}\sum_i e_i > \frac{1}{2}\sum_i e_i - \beta$. Since $\beta = 1/4$ while $v(A_i \cap \mathcal{M}^+)$ and $\frac{1}{2}\sum_i e_i$ are integers, it follows that $v(A_i \cap \mathcal{M}^+) \ge \frac{1}{2}\sum_i e_i$. Then partition $(A_1 \cap \mathcal{M}^+, A_2 \cap \mathcal{M}^+)$ of $E = (e_1, \ldots, e_m)$ is a solution of the PARTITION problem, a contradiction. □

Claims D.1 and D.2 show that $\text{MMS} > 0 \iff$ there is a solution to PARTITION.

When agents are identical, they agree on every item if it is a good or a chore, and therefore $\mathcal{M}^{gc} = \emptyset$. Therefore, v_i^+ and v_i^- as defined in Definition 2.2 are same as $v(\mathcal{M}^+)$ and $|v(\mathcal{M}^-)|$ respectively. □

Theorem D.2. *Given a fixed constant $\tau > 0$, even if an instance $(n, \mathcal{M}, v)$ with identical agents satisfies $|v(\mathcal{M})| \ge \tau \cdot \min\{v(\mathcal{M}^+), |v(\mathcal{M}^-)|\}$, checking if* MMS > 0 *is* NP*-hard.*

Proof. Again, we give a reduction from PARTITION. Let $E = \{e_1, e_2, \cdots, e_m\}$ be the set of elements given as input for PARTITION. Create an instance $(n, \mathcal{M}, v)$ as follows: $\mathcal{N}$ has n agents, where n will be fixed later based on the value of τ. $\mathcal{M} = \{1, 2, \cdots, m+n\}$ where the first $m + (n-2)$ items are goods, and the last 2 are chores. The valuation function v is defined as follows, where $\beta = 1/4$.

$$v_j = \begin{cases} e_j & \forall j \in [m] \\ \beta & \text{for } j \in \{m+1, ..., m+(n-2)\} \\ -\sum_{i\in[m]} e_i/2 + \beta & \text{for } j \in \{m+n-1, m+n\}. \end{cases}$$

That is, the first m goods have values equal to the weights of the corresponding elements of PARTITION. The remaining $(n-2)$ goods have value β each, and both the chores have value $-(\sum_i e_i/2) + \beta$. Fix n to satisfy $|v(\mathcal{M})| \ge \tau \cdot \min\{v(\mathcal{M}^+), |v(\mathcal{M}^-)|\}$, or equivalently $v(\mathcal{M}^+) \ge (1+\tau)|v(\mathcal{M}^-)|$, that is, $((n-2)\beta + \sum_i e_i) \ge (1+\tau)(\sum_i e_i + 2\beta)$.

We again have $0 \leq \mathsf{MMS} \leq \beta$. The lower bound because $v(\mathcal{M}) > 0$ and Lemma 2.1, and the upper bound because the average $v(\mathcal{M})/n$ is β and Lemma 2.2. The correctness is argued in the next two claims.

Claim D.3. PARTITION *has a solution* $\Rightarrow \mathsf{MMS} \geq \beta$.

Proof. Divide the first m goods as per the division of the elements of PARTITION into equal valued sets, and add one chore to each bundle. From the remaining goods $\{m+1, ..., m+(n-2)\}$ give one each to the remaining $(n-2)$ bundles. The value of every bundle created is β. Hence, $\mathsf{MMS} \geq \beta$. □

Claim D.4. $\mathsf{MMS} > 0 \Rightarrow$ PARTITION *has a solution.*

Proof. We prove the contrapositive of the statement, by contradiction. Suppose PARTITION instance $E = e_1, \ldots, e_m$ does not have a solution, but $\mathsf{MMS} > 0$ for $(n, \mathcal{M}, v)$.

Given that there are exactly two chores, at least $(n-2)$ bundles have only goods and has to have at least one good. Furthermore, since e_is are positive integers and $\beta = 1/4$, each of these $(n-2)$ bundles have value at least β. Now, β being the upper bound on the MMS value, wlog we can assume that these $(n-2)$ bundles have exactly one good of the minimum value, namely β. This exhaust the goods $\{m+1, \ldots, m+(n-2)\}$ with value β. Therefore, the two chores and all goods corresponding to the PARTITION problem elements, and no other good, are in the remaining two bundles. Let these be the first two bundles A_1 and A_2.

Now by the same argument as in the proof of Claim D.2, we can show that both A_1 and A_2 have positive value only if each contains exactly one chore and the total value of goods in each, namely $v(A_i \cap E)$ for $i = 1, 2$, is at least $\frac{1}{2}\sum_{i\in[m]} e_i$. Thus, $(A_1 \cap E, A_2 \cap E)$ is a solution to the PARTITION instance E, a contradiction. □

Claims D.3 and D.4 show that $\mathsf{MMS} > 0$ for $(n, \mathcal{M}, v) \iff$ there is a solution to PARTITION. □

Theorems D.1 and D.2 show that even if we know that $\mathsf{MMS} \geq 0$ checking if it is strictly positive is NP-hard. Since for $\alpha \in (0, 1]$, $\mathsf{MMS} > 0 \Leftrightarrow \alpha\mathsf{MMS} > 0$, this essentially means, we can not find an α-MMS allocation for *any* value of $\alpha \in (0, 1]$ if either of the two conditions in α-MMS problem is dropped. The next theorem formalizes this.

Theorem 3.1. *For any instance $(n, \mathcal{M}, v)$ with identical agents and $v(\mathcal{M}) > 0$ such that exactly one of the following two holds: (a) either $n = 2$ or (b) $|v(\mathcal{M})| \geq \tau \cdot \min\{v(\mathcal{M}^+), |v(\mathcal{M}^-)|\}$ for a constant τ, finding an α-MMS allocation of $(n, \mathcal{M}, v)$ for any $\alpha \in (0, 1]$ is* NP*-hard.*

Even though an instance with identical agents is guaranteed to have an allocation where every agent gets at least the MMS value, i.e., 1-MMS allocation exists, Theorem 3.1 ruling out an efficient algorithm for finding α-MMS allocation any $\alpha \in (0, 1]$ is very striking. In light of this result, it is evident that even getting a PTAS, in other words finding $(1-\epsilon)$-MMS allocation, in case of identical agents is non-trivial and important.

E DETAILED RELATED WORK

Fairness and efficiency in mixed manna. While ours is the first work on MMS + PO, finding fair and efficient allocations has been studied for other notions. [ACIW19] initiate the study for a mixed manna, and study the problem of finding EF1 + PO allocations. [AW20a] study fairness properties related to EFX, defined as envy-freeness up to any item along with PO.

Fairness for Mixed Manna. Finding fair allocations of mixed items has recently caught a lot of attention for both divisible [BMSY17, BMSY19] and indivisible [AW19, AW20b, Ale20, GM20], items. However, to the best of our knowledge, ours is the first study on MMS allocations for a mixed manna.

Fairness and efficiency in goods manna. This problem is well-studied for a goods manna. Two popular notions for a goods manna are the Nash social welfare (NSW), and EF1 + PO, defined and discussed below.

NSW. Nash Social Welfare (NSW) is the geometric mean of the valuation of the agents. The NSW problem is to find an allocation of indivisible items that maximizes NSW. This problem is APX-Hard [MG20], and remarkable approximation results for the linear valuations case have been proven by a connection of the problem with markets [CG15, CDG+17, BKV18, CCG+18] or real stable polynomials [AGSS17]. The best known result is a 1.45 approximation factor [BKV18]. Similar results are known, again by exploiting the market connection, for popular valuation functions like budget-additive [GHM19], separable piece-wise linear concave (SPLC) [AMGV18], and their combination [CCG+18]. Recent results give an $O(n)$ approximation when agents have subadditive valuations, a far more general class than all the earlier ones [BBKS20, CGM20]. Recent work has also been done on the general version of the problem with asymmetric agents, where the aim is to maximize the weighted geometric mean, for given weights, and submodular utilities [GKK20]. This notion is not applicable for a mixed manna.

EF1 + PO. EF1 was first introduced by [Bud11] as an relaxation of envy-freeness. An allocation is EF1 if for any two agents i_1 and i_2, agent i_1 prefers (or equally likes) her own bundle to agent i_2's bundle after removing *some item* from the bundle of agent i_2. An EF1 allocation can be found efficiently using envy cycle removal procedure introduced by [LMMS04]. [BKV18] show a pseudo-polynomial time algorithm to obtain an EF1 + PO allocation on a goods manna. A series of works [AMS20, ZP20, CGMM20, SSH19] study special cases of the problem.

Other notions studied for a goods manna are Prop1+PO [AMS20]), or group fairness notions [CFSV19]. When the preferences are ordinal, [AHMSH19] discuss EF1 solutions that satisfy the efficiency notions of utilitarian maximality and rank maximality.

MMS. The study of fair division started with the cake cutting problem [Ste48]. Two popular notions of fairness established here were proportionality, meaning each agent must get a bundle worth at least $1/|\mathcal{N}|$ of her value for all items, and envy-freeness, where each agent must value her own bundle at least as much as any other. However, neither of these can always be attained when the items are indivisible. A simple example is allocating one good between two agents; there is no allocation that is proportional or envy-free. This motivated the search for new fairness notions for indivisible items. One well-studied notion resulting from this investigation is MMS [Bud11]. In recent years, the problem of finding MMS allocations gained a lot of interest, and a series of impressive results were found for various special cases of the problem, as discussed below.

MMS **for Goods.** [BL16] showed that in some restricted cases MMS allocations always exist. A notable result from [PW14] showed that MMS allocations may not always exist but 2/3-MMS allocations always do. A series of works studied the efficient computation of 2/3-MMS allocations for any n [AMNS17, BKM17, GMT18]. [GHS+18] showed that a 3/4-MMS allocation always exists. Most recently [GT20] showed that a $(3/4 + 1/(12n))$-MMS allocation always exists. Finding MMS values is hard but a PTAS for this problem is known [Woe97]. This PTAS can be used to find a $(3/4+1/(12n)-\epsilon)$-MMS allocation for $\epsilon > 0$ in polynomial time. There is also a strongly polynomial time algorithm to find 3/4-MMS allocation [GT20]. Other notable works on the goods only case before being improved by follow-up work are [FGH+19, GMT18, KPW16, KPW18].

Constant number of agents with a goods only manna. For three agents, [AMNS17] showed that a 7/8-MMS allocation always exists. This factor was later improved to 8/9 in [GM19]. For four agents, [GHS+18] showed that a 4/5-MMS allocation always exist.

MMS **for Chores.** [ARSW17] first studied the MMS problem with a chores manna. They introduced an algorithm for finding 2-MMS allocations[4]. [BKM17] improved the previous result by showing an algorithm for a 4/3-MMS allocation. Later, [HL19] improved this result to a 11/9-MMS allocation. They also showed a PTAS to find $(11/9 + \epsilon)$-MMS allocation and a polynomial time algorithm to find a 5/4-MMS allocation.

Other variants of MMS**.** The MMS problem has been studied under various other models in the goods only setting like with asymmetric agents [FGH+19], group fairness [BBKN18, CKMS20], beyond additive valuations [BKM17, GHS+18, LV18], in matroids [GM19], with additional constraints [GM19, BB18], for agents with externalities [BMR+13, AEG+13], with graph constraints [BILS19, LT19], and with strategic agents [BGJ+19]. In the chores only setting too, weighted MMS [ACL19b], and asymmetric agents [ACL19a] notions have been investigated.

[4]Our definition of α-MMS for the mixed manna is consistent for agents with positive as well as negative MMS values. We define α as smaller than 1, and consider $1/\alpha$-MMS valued bundles as α-MMS. Prior results for the chores manna have $\alpha > 1$ and ask for $\alpha \cdot$ MMS valued bundles. We state the approximation factors as defined in the original papers, and ask the reader to invert them when relating with ours.