Approximating MMS and (symmetric) APS under Cardinality Constraints: Goods, Bads, and the Best-of-Both-Worlds

Arjun Aggarwal
University of Illinois
Urbana-Champaign
Champaign, United States
arjuna4@illinois.edu

Kyra Gunluk
Cornell University
Ithaca, United States
kg556@cornell.edu

Ruta Mehta
University of Illinois
Urbana-Champaign
Champaign, United States
rutameht@illinois.edu

ABSTRACT

We study the problem of finding fair allocation of indivisible items (goods and chores) among agents with *heterogeneous cardinality constraints* under additive valuations. That is, each agent has their own cardinality constraint on the items they may receive. We consider one of the strongest share-based fairness notions when agents have equal entitlements, namely symmetric Any Price Share (APS). We note that, all the results extend to Maximin Share (MMS) fairness as well since APS is a stronger notion.

We design polynomial time algorithms to find: (i) 1/2-APS allocation for goods, (ii) 2-APS allocation for chores, and (iii) 3/2-APS allocation for chores when agents have symmetric cardinality constraints. Finally, we extend one of our algorithms to provide *best-of-both worlds* guarantee, namely ex-ante proportionality and ex-post 1/2-APS and 2-APS for the cases of goods and chores, respectively.

KEYWORDS

Fair Allocation, APS, MMS, Cardinality Constraints

ACM Reference Format:
Arjun Aggarwal, Kyra Gunluk, and Ruta Mehta. 2024. Approximating MMS and (symmetric) APS under Cardinality Constraints: Goods, Bads, and the Best-of-Both-Worlds. In *ACM Conference, Washington, DC, USA, July 2017*, IFAAMAS, 10 pages.

1 INTRODUCTION

Discrete fair division is a fundamental problem within the social choice theory, where a set M of m indivisible items needs to be allocated *fairly* among n agents. The preferences of each agent i are given by a valuation function $V_i : 2^M \to \mathbb{R}$. Arguably, two of the most popular share-based fairness notions are of Maximin Share (MMS) and Any Price Share (APS). We study MMS and symmetric-APS for allocating both goods and chores to agents with cardinality constraints. That is, agents have additive valuation functions with cardinality constraints which may be heterogeneous across agents, *i.e.,* for any $S \subseteq M$, $V_i(S) = \sum_{j \in S} v_i(j)$ as far as $|S|$ satisfies agent i's cardinality constraint.

Both MMS and APS are *share based* fairness notions, where each agent is entitled to a bundle worth their *fair share.* Under MMS, this *fair share* of an agent is defined as the maximum value she can guarantee herself under the classical *cut-and-choose* mechanism when she is the cutter; she partitions the item set into n bundles and gets to pick last. Therefore, clearly, she will partition such that the value of the minimum valued bundle is maximized. If $\Pi(M)$ denotes the set of all allocations of M among the n agents, and $(A_1 \dots, A_n)$ denotes any allocation into n bundles, then the MMS value of agent i is defined as,

$$\mathrm{MMS}_i = \max_{(A_1,\dots,A_n) \in \Pi(M)} \min_{j \in [n]} V_i(A_j)$$

An MMS allocation is one where every agent i gets a bundle worth at least MMS_i. [3] introduced a stronger notion called *Any Price Share (APS)* via a pricing mechanism.[1] That is, if denotes the simplex of all price vectors for the m goods where the sum of all prices is 1 and all prices are non-negative, hence $= \{(p_1 \dots, p_m) \mid \sum_{j=1}^{m} p_j = 1,\ p_j \geq 0\ \forall j\}$, then APS is given by,

$$\mathrm{APS}_i = \min_{(p_1,\dots,p_m) \in} \ \max_{S \subseteq M : \sum_{j \in S} p_j \leq 1/n} V_i(S)$$

[3] showed that $\mathrm{APS}_i \geq \mathrm{MMS}_i$ for all agent i as far as the valuation functions are monotone.

Allocations achieving MMS and APS shares may not exist even under additive valuations [21, 10]. Therefore, the focus has been on finding approximate solutions, where in an α-APS (MMS) allocation, every agent receives a bundle worth at least α times their APS (MMS) value. This problem has been studied extensively for additive valuations and when M contains only goods (see [2] for a survey and pointers), with much progress [21, 11, 16]. It is known that a $(\frac{3}{4} + \frac{3}{3836})$-MMS always exists and can be computed in polynomial time [1], while there are examples showing that $\frac{39}{40}$-MMS may not exist [10] even in the setting of three agents.

We consider agents with cardinality constraints. That is, each agent i has an additive valuation function V_i as well as a cardinality constraint of $k_i > 0$, with the understanding that if M is a set of goods, then agent i wants no more than k_i items, and if M is a set of chores then agent i needs to be allocated at least k_i items. Note that such a valuation function generalizes additive while is contained in submodular and supermodular functions for goods and chores, respectively. The latter are not as well-studied. The best bound known for goods under submodular functions are 10/27 and 1/3 for MMS and APS, respectively [22], while nothing is known for the chores under supermodular functions.

[15] studied MMS under cardinality constraints, but assumed *homogeneous* cardinalities, *i.e.,* $k_i = k$, $\forall i$. They gave algorithms to find 2/3-MMS and 3/2-MMS respectively for goods and chores.

[1] We note that APS was defined with respect to asymmetric agents, where agent i has budget/weight of $b_i > 0$ while $\sum_i b_i = 1$, in this paper we focus on symmetric-APS.

For more complex valuation function. where M is partitioned into sets $M_1, \ldots, M_d$ and each set has a separate cardinality constraint namely k^l for set M_l, then extended the goods algorithm to find 1/2-MMS allocations.

We extend the above results on two fronts. First, we allow agents to have heterogeneous cardinality constraints, and second, we extend the results to the stronger notion of APS.

(1) 1/2-APS for goods
(2) 2-APS for chores.
(3) 3/2-APS for chores when agents have homogeneous cardinality constraints.
(4) *Best of both worlds* guarantee: ex-ante proportional and ex-post 1/2-APS and 2-APS for goods and chores, respectively.

We note that all the above results extend to MMS since APS is a stronger notion [10]. On the technical side, we obtain the following new insights that may be of independent interest:

- *Sliding-window with Breaks.*
- *bag-filling until last shout* and *reverse-bag-filling*
- Analysis of the APS value for bounds and structure under heterogeneous cardinality constraints.

2 FURTHER RELATED WORK

The notion of MMS was introduced by [7]. Homogeneous cardinality constraints were implemented within heterogeneous categories by [15], giving a $\frac{1}{2}$-MMS allocation for goods, and a $\frac{2}{3}$-MMS allocation in the single-category instance, as well as a 2-MMS allocation for chores, and a $\frac{3}{2}$-MMS allocation in the single-category instance. [3] Introduced the notion of APS and proved that APS $\geq MMS$ in the goods case, and gave a $\frac{3}{5}-$APS allocation for additive valuations and arbitrary entitlements.

Beyond additive valuations, some well-known classes of valuation functions such as subadditive, fractionally sub-additive i.e. XOS, submodular, and their interesting special cases have been studied in the literature [5, 17, 23]. For submodular valuations, [13] gave an algorithm to find a $\frac{1}{3}$-MMS allocation based on a certain local search procedure, and [4] showed that a simple round-robin procedure can achieve a $\frac{1}{3}(1-1/e)$-MMS allocation. This was recently improved to $\frac{10}{27}$-MMS by [22], who also gave $\frac{1}{3}$-APS algorithm. For subadditive valuations, it is known that better than $\frac{1}{2}$-MMS allocations may not exist [13].

Yet another class of fairness notions are *envy-based.* The classical notion of envy-freeness (EF) dictates that no agent envies another agent's bundle over her own. However, an EF allocation may not exist for allocating indivisible items. There has been extensive work on relaxations of EF. The strongest among these is EF up to any item (EFX) [8]. The other popular notion is of EF up to one item (EF1). The existence of EFX is known only for special cases, e.g., [20, 9, 19, 14], and remains a major open problem [21]. While EF1 is well-known to exist in general [18], and can be combined with the efficiency guarantees.

3 PRELIMINARIES: APS AND MMS FOR GOODS

To start with, in this section, we define all the terms with respect to goods. We differ the definitions for chores to Section 5.1 to avoid confusion. An instance of the fair allocation problem under *heterogeneous cardinality constraints* is given by $I = \langle N, M, V, K \rangle$, where $N = \{1, 2, \ldots n\}$ is the set of agents, $M = \{1, 2, \ldots m\}$ is the set of goods, $V = \langle v_1, v_2, \ldots v_n \rangle$ is the collection of the agents' valuation functions $v_i : 2^M \to \mathbb{R}_+$, and $K = \langle k_1, k_2, \ldots k_n \rangle$ is the collection of cardinality thresholds. Further, we can assume instances are ordered, that is agents have the same ranking of goods, without loss of generality by [4, 6]. When referring to agent i for a single item j, we use the notation $v_i(j)$ to denote the agent's value of said item. Without loss of generality, we assume that the agents are arranged in non-decreasing order of thresholds. Hence, $k_1 \leq k_2 \cdots \leq k_n$. We assume that the valuation functions are additive i.e. $v_i(S) = \sum_{g \in S} v_i(g)$ for any subset $S \subseteq M$. An *allocation* $A = (A_1, A_2, \ldots A_n) \in \Pi_n(M)$ is an n-partition of M, with A_i being the bundle received by agent i.

Let $\pi_i^j(S)$ denote the set of the max $(|S|, j)$ most valuable goods in S for agent i. If the instance is ordered, then π_i^j is the same for every agent, and in that case, we may omit the subscript. For any agent i, we define the *final valuation function* $f_i : 2^M \to \mathbb{R}_+$ as follows:

$$f_i(S) := \sum_{g \in \pi_i^{k_i}(S)} v_i(g)$$

Under the final valuation function, agent i can have value for at most k_i items in a set. Thus, these final valuation functions have the cardinality constraints "built" into them. We use these functions to define the maximin share of each agent.

Definition 3.1 (Maximin Share). Let $I = \langle N, M, V, K \rangle$ be an instance of the fair allocation problem for goods under heterogeneous cardinality constraints. For an agent i, let f_i be as defined above. The *maximin share* of i is defined as

$$\text{MMS}_i := \max_{A \in \Pi_n(M)} \min_{A_j \in A} f_i(A_j)$$

Definition 3.2 (Any Price Share). Let $I = \langle N, M, V, K, B \rangle$ be an instance of the fair allocation problem for goods under heterogeneous cardinality constraints, where $B = \langle b_1, b_2, \ldots b_n \rangle$ is the set of agents' entitlements with every $b_i \geq 0$, and $\sum_{i=1}^n b_i = 1$. For an agent i, let f_i be as defined above. The *any price share* of i is defined as

$$\text{APS}_i := \min_{(p_1, p_2, \ldots p_m) \in P} \max_{S \subseteq M} \left\{ f_i(S) \mid \sum_{j \in S} p_j \leq b_i \right\}$$

Where $P = \{(p_1, p_2, ..., p_m) \mid p_j \geq 0 \; \forall j \in M, \sum_{j \in M} p_j = 1\}$ is the set of item-price vectors that sum to 1.

Definition 3.3 (Any Price Share, dual definition). Let $I = \langle N, M, V, K, B \rangle$ be an instance of the fair allocation problem for goods under heterogeneous cardinality constraints, where $B = \langle b_1, b_2, \ldots b_n \rangle$ is the set of agents' entitlements with every $b_i \geq 0$, and $\sum_{i=1}^n b_i = 1$. The *Any Price Share* of i is defined as

$$\text{APS}_i := \max z$$

Where z is subject to the following constraints:

1. $\sum_{T \subseteq M} \lambda_T = 1$
2. $\lambda_T \geq 0 \; \forall T \subseteq M$
3. $\lambda_T = 0 \; \forall T \subseteq M \; s.t. \; f_i(T) < z$
4. $\sum_{T \subseteq M : j \in T} \lambda_T \leq b_i \; \forall j \in M$

As shown in [3] these two definitions are equivalent.

APS (MMS) Allocation. An allocation A is said to be an APS (MMS) allocation if for every agent i, $f_i(A_i) \geq APS_i$ ($f_i(A_i) \geq MMS_i$). It is an α-APS (α-MMS) allocation for an $\alpha \in [0,1]$ if for each $i \in N$, $f_i(A_i) \geq \alpha * APS_i$ ($f_i(A_i) \geq \alpha * MMS_i$). [3] also showed that an α-APS allocation A is also an α-MMS allocation. Therefore, from now on we will focus on finding α-APS allocation, and the results extend to α-MMS allocations.

4 APPROXIMATE APS FOR GOODS

We start with showing two important claims regarding APS that are crucial for our algorithm: (i) one-good reduction for APS (similar to MMS [12]), and (ii) for agent i, it suffices to focus on top nk_i items while computing APS the APS value (Definition 3.2.

Lemma 4.1 (One-Good Reduction). Let $I = \langle N, M, V, K, B\rangle$ be an ordered instance of fair division under ordered heterogeneous cardinality constraints, and with equal entitlements ($b_j = \frac{1}{n}\ \forall j \in M$). If a single item is given to an agent, and this item and agent are then removed from the instance, all remaining agents' APS value remain the same or are increased.

Proof. Let APS$_i$ be the optimal solution to the original instance of the LP, and consider the same solution used in the new instance that has an item and agent removed. We can prove the properties of the One-Good Reduction by showing that for any agent i, the optimal $z^* = \text{APS}_i$ is still a solution for the new instance, and thus the maximal z to the new instance must be greater or equal to this feasible value z^*. We will construct the new feasible solution from the old optimal solution as follows:

Given the entitlements are equal, $b_i = \frac{1}{n}\ \forall i$ in the original instance, and after the removal of item j^* and agent i^*, the entitlements will be $b'_i = \frac{1}{n-1}\ \forall i$. Additionally, in the new instance, $M' = M \setminus \{j^*\}$, so every $T' \subseteq M'$ is also a subset of M, and the remaining $T \subseteq M, T \nsubseteq M'$ are the exact sets $T' \cup \{j^*\}, \forall T' \subseteq M'$.

Let λ^*_T be the value of λ_T in the optimal solution for the original problem for each $T \subseteq M$. Let $z' = z^*$ and $\lambda'_T = \frac{\lambda^*_T}{S}$ where $S = \sum_{K \subseteq M : j^* \notin K} \lambda^*_K = \sum_{K \subseteq M'} \lambda^*_K$. Note that S is strictly positive (specifically greater than $\frac{n-1}{n}$ as proven later) so this fraction is valid.

The first constraint holds true, since

$$\sum_{T \subseteq M'} \lambda'_T = \sum_{T \subseteq M'} \frac{\lambda^*_T}{S} = \frac{\sum_{T \subseteq M'} \lambda^*_T}{S} = 1.$$

The second constraint holds true because the same constraint in the original LP implies that all $\lambda^*_T \geq 0$, and so $S = \sum_{K \subseteq M : j \notin K} \lambda^*_K \geq 0$, thus it follows that $\lambda'_T = \frac{\lambda^*_T}{S} \geq 0$.

The third constraint remains true because if $v_i(T') < z$ in the new LP, then in the original LP, $v_i(T) < z$ for $T = T'$, and by the third constraint in the original LP, this implies that $\lambda^*_T = 0$. Thus, $\lambda'_T = \frac{\lambda^*_T}{S} = 0$, satisfying the third constraint of the new LP.

The fourth constraint also holds. Using the original LP we find that

$$\sum_{T \subseteq M : j \in T} \lambda^*_T \leq \frac{1}{n} \forall j \in M \implies \sum_{T \subseteq M : j^* \in T} \lambda^*_T \leq \frac{1}{n} \forall j^* \in M$$

By the first constraint in the original LP,

$$1 = \sum_{T \subseteq M} \lambda^*_T = \sum_{T \subseteq M : j^* \in T} \lambda^*_T + \sum_{T \subseteq M : j^* \notin T} \lambda^*_T$$

so

$$S = \sum_{T \subseteq M : j^* \notin T} \lambda^*_T = 1 - \sum_{T \subseteq M : j^* \in T} \lambda^*_T \geq 1 - \frac{1}{n} = \frac{n-1}{n}.$$

Thus,

$$\sum_{T \subseteq M' : j \in T} \lambda'_T = \sum_{T \subseteq M' : j \in T} \frac{\lambda^*_T}{S} = \frac{\sum_{T \subseteq M' : j \in T} \lambda^*_T}{S} \leq \frac{1/n}{n - 1/n} = \frac{1}{n-1},$$

thus it holds that

$$\sum_{T : j \in T} \lambda'_T \leq \frac{1}{n-1} = b'_i \forall j \in M'.$$

Finally we can see that $(z', \lambda'_T\ \forall T \in M')$ is a feasible solution to the new LP, thus the optimal solution cannot have a smaller z value than the original z^*, as we are maximizing, so the APS cannot have decreased. □

Lemma 4.2. Let $I = \langle N, M, V, K, B\rangle$ be an ordered instance of fair division under ordered heterogeneous cardinality constraints, and with equal entitlements ($b_j = \frac{1}{n}\ \forall j \in M$). For any price vector $(p_1, p_2, \ldots, p_m)$ satisfying $p_j \geq 0\ \forall j \in M, p_j = 0\ \forall j \in M \setminus \Pi_i^{nk_i}(M)$, $\sum_{j \in M} p_j = 1$, the highest valued affordable bundle consists only of items from the top nk_i items.

$$\operatorname*{argmax}_{S \subseteq M} \left\{ f_i(S) \mid \sum_{j \in S} p_j \leq b_i \right\} \subseteq \Pi_i^{nk_i}(M)$$

The detailed proof of the above lemma may be found in Appendix A. The high-level proof sketch is as follows: To prove this claim, we will use induction on the number of agents, n. We assume towards contradiction that an instance with n agents requires the bundle selected for APS$_i$ to include an item that is not in the $\Pi_i^{nk_i}(M)$ items, and by using this bundle and the price vector used to find APS$_i$ we will construct an instance for the $n-1$ case, where we prove that it is required to also have an item not in the $\Pi_i^{nk_i}(M)$ items in the optimal bundle. As this contradicts our inductive hypothesis, we can conclude that the n case must hold.

Lemma 4.3. Let $I = \langle N, M, V, K, B\rangle$ be an ordered instance of fair division under ordered heterogeneous cardinality constraints, and with equal entitlements ($b_j = \frac{1}{n}\ \forall j \in M$). For any price vector $(p_1, p_2, \ldots, p_m)$ satisfying $p_j \geq 0\ \forall j \in M, \sum_{j \in M} p_j = 1$, the highest valued affordable bundle consists only of items from the top nk_i items.

$$\operatorname*{argmax}_{S \subseteq M} \left\{ f_i(S) \mid \sum_{j \in S} p_j \leq b_i \right\} \subseteq \Pi_i^{nk_i}(M)$$

Proof. By lemma 4.2, we know this to be true if only the $\Pi_i^{nk_i}(M)$ items have associated price, while the rest cost 0. Consider price vector $(p_1, p_2, \ldots, p_m)$ that does not have this property. Such a price vector must thus have a lower price for at least one of the items in $\Pi_i^{nk_i}(M)$, making it more affordable.

Thus, either the maximum value affordable bundle in this case is the same as in the lemma 4.2 case, which would only contain of items in the $\pi_i^{nk_i}(M)$ items, or the bundle can now include an additional or a better item from the $\pi_i^{nk_i}(M)$ items. Since the pricing is chosen to minimize the value of the maximum affordable bundle, this pricing would not be chosen over one that assigns higher prices to the items in the $\pi_i^{nk_i}(M)$. Either way, the adversarial pricing chosen will result in a bundle of items from the $\pi_i^{nk_i}(M)$ items. ◻

Lemma 4.4. Let $I = \langle N, M, V, K, B \rangle$ be an ordered instance of fair division under ordered heterogeneous cardinality constraints, and with equal entitlements ($b_j = \frac{1}{n}\ \forall j \in M$). For any agent i, the APS value of the agent is at most their proportional share.

$$\mathrm{APS}_i \le \frac{1}{n} v_i(\pi_i^{nk_i}(M))$$

Proof. Consider the price vector $p_j = \dfrac{v_{ij}}{v_i(\pi_i^{nk_i}(M))}$ for all items j in the top nk_i items, and $p_j = 0$ for all other j. This satisfies price constraints, $p_j \ge 0$, and

$$\sum_{j \in M} p_j = \sum_{j \in \pi_i^{nk_i}(M)} \frac{v_{ij}}{v_i(\pi_i^{nk_i}(M))} = \frac{\sum_{j \in \pi_i^{nk_i}(M)} v_{ij}}{v_i(\pi_i^{nk_i}(M))} = 1$$

Any $S \subseteq M$ that satisfies $\sum_{j \in S} p_j \le b_i$, or

$$\sum_{j \in S \cap \pi_i^{nk_i}(M)} \frac{v_{ij}}{v_i(\pi_i^{nk_i}(M))} \le \frac{1}{n}$$

implies that $v_i(S \cap \pi_i^{nk_i}(M)) \le \frac{1}{n} v_i(\pi_i^{nk_i}(M))$. By lemma 4.2, it is true that using this price vector, the optimal affordable bundle contains items only from $\pi_i^{nk_i}(M)$. Thus, for

$$S^* = \operatorname*{argmax}_{S \subseteq M} \left\{ v_i(S) \mid \sum_{j \in S} p_j \le b_i \right\}, S^* \subseteq \pi_i^{nk_i}(M)$$

so $S^* \cap \pi_i^{nk_i}(M) = S^*$ so it is true that $v_i(S^*) \le \frac{1}{n} v_i(\pi_i^{nk_i}(M))$, or,

$$\max_{S \subseteq M} \left\{ v_i(S) \mid \sum_{j \in S} p_j \le b_i \right\} \le \frac{1}{n} v_i(\pi_i^{nk_i}(M))$$

Since APS takes the minimum over all price vectors, the APS value of agent i must be at most the proportional share. ◻

From now on we can assume valuations are scaled such that $v_i(\pi_i^{nk_i}(M)) = n$ and thus $\mathrm{MMS}_i \le 1$ and $\mathrm{APS}_i \le 1$ by lemma 4.4, and item values less than $\frac{1}{2}$ have been allocated through the One-Good Reduction.

4.1 (1/2)-APS for Goods: via Round-Robin

The Round-Robin algorithm consists of agents in a pre-determined order taking turns picking an item until all items are gone.

Algorithm 1 1/2-APS Under Heterogeneous Cardinalities, Round Robin

Input: Scaled, Ordered Instance

$A \leftarrow (\emptyset, \emptyset \dots \emptyset)$ ▷ *Empty Allocation*
while $M \neq \emptyset$ **do**
  for $i = 1$ to n **do**
    Let $c = \operatorname{argmax}_{j \in M} v_i(j)$
    $A_i \leftarrow A_i \cup \{c\}$
    $M \leftarrow M \setminus \{c\}$
    if $M = \emptyset$ **then**
      break

Theorem 4.1. *Algorithm 1* returns a 1/2-MMS and 1/2-APS allocation under heterogeneous cardinality constraints for goods.

Proof. Every agent will receive $\frac{|M|}{n}$ items, and those who have a cardinality bound of $k_i < \frac{|M|}{n}$ will use the final valuation function f_i to only give value to their top k_i items, and ignore the remaining ones.

Now, for every agent i consider the value they are given (or value "taken"): items $i, n+i, 2n+i, \dots$ and the value they have missed out on (the value the other agents have taken, or value "lost"): all other items in the top nk_i items. Each item that comes before i, which there are less than n of, is upper bounded by $\frac{1}{2}$. Every item that comes after $jn+i$ for $j = 0, 1, \dots, k_i$ is upper bounded by $v_i(jn+i)$, because items are sorted by value in decreasing order. There are n-1 items strictly between $jn+i$ and $(j+1)n+i$, so for each item $jn+i$ given to agent i, they missed out on $n-1$ items of value at most $jn+i$.

Thus the total value "taken" by i is $v_i(i) + v_i(n+i) + v_i(2n+i) + \dots = \sum_{j=0}^{k_i-1} v_i(jn+i)$ And the total value "lost" by i is at most

$$\frac{1}{2}n + (n-1)v_i(i) + (n-1)v_i(n+i) + \dots = \frac{n}{2} + (n-1)\sum_{j=0}^{k_i-1} v_i(jn+i)$$

since there are at most n items before i that are upper bounded by $\frac{1}{2}$ and $n-1$ items that are upper bounded by $jn+i$ for every item $i, n+1, 2n+i, \dots, k_i n + i$.

The total value of the top nk_i items is the value of the items i received plus the value of the items all other agents have received from that range, or value "taken" + value "lost"

$$\le \sum_{j=0}^{k_i-1} v_i(jn+i) + \frac{n}{2} + (n-1)\sum_{j=0}^{k_i-1} v_i(jn+i) = \frac{n}{2} + n\sum_{j=0}^{k_i-1} v_i(jn+i),$$

and since the top nk_i items are valued at n, we see that

$$n \le \frac{n}{2} + n\sum_{j=0}^{k_i-1} v_i(jn+i) \implies \frac{1}{2} \le \sum_{j=0}^{k_i-1} v_i(jn+i)$$

so the value "taken" by i is at least $\frac{1}{2}$.

Thus every agent i takes value at least $\frac{1}{2}$, and since MMS and APS are each at most 1, a bundle of value $\frac{1}{2}$ is at least $\frac{1}{2}$ − MMS and at least $\frac{1}{2}$ − APS.

◻

4.2 (1/2)-APS for Goods: via Bag-Filling

In this algorithm, each agent will only have "view" of their top nk_i items, so they will never be given the opportunity to value a bundle that is not in this range. In addition, any bag taken will have at most k_i items from the top nk_i items for every i, and we can ensure that the range of view of each agent decreases by k_i every iteration, so the range always equals the number of remaining agents times k_i.

The bag-filling algorithm begins with n many empty bags labeled B_1 through B_n. We then iterate from $j = n$ to 1 to fill each bag. In each iteration, the bag B_j will receive the k_{max} least valuable remaining goods. If an agent values this bag at least 1/2, the bag and the agent can be resolved. Otherwise, we enter a while loop that shifts the window of included items in the bag by swapping the worst item in the bag with the worst item not yet considered. This will continue until a barrier of nk_i is reached for some i, in which case $|B_j| - k_i$ items will stay in place immediately before the barrier, while only k_i of the items will continue shifting. This while loop terminates when some agent values the bag at least 1/2, which we prove is guaranteed, the bag will be given to such an agent, and the agent and items will be removed from the instance, and the next iteration will begin filling the next bag. At the end of the algorithm if there are any remaining items they can be distributed arbitrarily, as doing so can only increase the agents' assigned values.

Let $\mathrm{g_{min}}(S)$ denote the least valuable good in S (same for all agents since we assume the instance is ordered) for any $S \subseteq M$.

Algorithm 2 (1/2)-APS Under Heterogeneous Cardinalities, Bag-Filling

Input: Scaled, Ordered instance with $v_i(j) < 1/2 \; \forall i \in N \; \forall j \in M$

1: $A = (\emptyset, \emptyset, \dots, \emptyset)$
2: $B_1 = \{\}, B_2 = \{\} \dots B_n = \{\}$
3: **for** $j = n$ to 1 **do**
4: Add the k_{max} least valuable items to B_j
5: **while** $f_i(B'_j) < 1/2 \; \forall i \in N$ **do**
6: UPDATEBAG(I, B'_j, j)
7: Find smallest $i \in N$ with $f_i(B_j) \geq 1/2$
8: $A_i = \pi^{k_i}(B_j)$, $M = M \setminus \pi^{k_i}(B_j)$, $N = N \setminus \{i\}$, $K = K \setminus \{k_i\}$

If there are remaining items, distribute them arbitrarily.

1: **function** UPDATEBAG(I, B_j, j)
2: Let $i := \min\{t \in N : (\pi^{jk_t}(M)) \cap B_j \neq \phi\}$
3: **if** $|(\pi^{jk_i}(M)) \cap B_j| < k_i$ **then**
4: Let $i' := \min\{t \in N : |(\pi^{jk_t}(M)) \cap B_j| = k_t\}$
5: Swap $\mathrm{g_{min}}((\pi^{jk_i}(M)) \setminus B_j)$ with $\mathrm{g_{min}}((\pi^{jk_{i'}}(M)) \cap B_j)$
6: **else**
7: Swap $\mathrm{g_{min}}((\pi^{jk_i}(M)) \setminus B_j)$ with $\mathrm{g_{min}}((\pi^{jk_i}(M)) \cap B_j)$

Theorem 4.2. *Algorithm 2 returns a 1/2-APS allocation under heterogeneous cardinality constraints for goods.*

Proof. To prove that this algorithm finds a $\frac{1}{2}$−MMS and $\frac{1}{2}$−APS allocation of goods to agents in polynomial time we must prove that (i) in every iteration a bag is assigned of value at least $\frac{1}{2}$ and (ii) the algorithm runs in polynomial time.

To prove (i), we need to ensure that at each iteration j of the for-loop, the while loop in lines 5-7 eventually terminates, or more specifically, UPDATEBAG() eventually produces a bag B_j such that some agent i' values it at least 1/2. To prove this, we can show that throughout the while-loop, the invariant $v_i(\pi^{jk_i}(M)) \geq j$ holds for every agent i. This will prove that when $j = 1$, $v_i(\pi^{k_i}(M)) \geq 1$, so for any remaining agent i, there is a bag of size at most k_i and value at least 1, and UPDATEBAG() will find such a bag by shifting the window in range jk_i up to the top k_i items, through the swapping in line 7.

We know this claim holds for $j = n$, since we re-scaled the values such that $v_i(\pi^{nk_i}(M)) = n$

Now assume that at iteration j, $v_i(\pi^{jk_i}(M)) \geq j$ for all agents i. We want to show that for the following iteration, $v_i(\pi^{(j-1)k_i}(M \setminus B_j)) \geq j - 1$ for all remaining agents i.

Case 1: $B_j \cap \pi^{jk_i}(M) = \phi$

In this case, if the last k_i items are valued less than 1, then certainly the remaining top $(j-1)k + i$ will be valued greater than $j - 1$. If the items are valued greater than 1, then since every other of the $j - 1$ chunks of size k_i has a better value, they will also have value greater than 1 and will add up to have value greater than $j - 1$.

Case 2: $B_j \cap \pi^{jk_i}(M) \neq \phi$

By the else of the conditional on line 4 of UPDATEBAG(), there will never be a bag of size greater than k_i taken from the top jk_i items.

If exactly k_i items are taken, then

$$v_i(\pi^{(j-1)k_i}(M \setminus B_j)) = v_i(\pi^{jk_i}(M) \setminus B_j) = v_i(\pi^{jk_i}(M)) - v_i(B_j)$$

by additivity of valuation. Since at iteration j the while loop terminates the first time an agent (or agents) values the bag $\geq 1/2$, then in the iteration of the while loop one previous to termination, the bag currently is valued $< 1/2$ for all agents, so $v_i(B_j) < 1/2$. During the last while loop iteration, UPDATEBAG() only adds or swaps one good, which must be valued less than 1/2, so the bags value cannot increase by 1/2 or more. Thus, after this iteration, $v_i(B_j) < 1/2 + 1/2 = 1$. Thus, $v_i(\pi^{jk_i}(M)) - v_i(B_j) \geq j + 1$.

If the assignment in iteration j has size $k < k_i$ taken from the top jk_i items, then either agent i has taken the bag or another agent has taken a smaller portion of the original B_j. In this case there are two possibilities:

If B_j has less than k_i items of the top jk_i items in it then by the conditional of UPDATEBAG() in line 3, the bag must still be at the bottom k items in the range. Thus it will again be true that the last k_i items of $\pi^{jk_i}(M)$ will be removed to make $\pi^{(j-1)k_i}(M \setminus B_j)$, and we have shown in **Case 1** that this will result in $v_i(\pi^{(j-1)k_i}(M \setminus B_j)) \geq j - 1$.

In any other case, B_j has exactly k_i items of the top jk_i items in it. We know that B_j is valued at most 1 by agent i, so it must be true that the value of the k items taken and the $k_i - k$ items in the top jk_i items that were in B_j, but were not taken, must be less than 1. In this case, the $k_i - k$ items in the jk_i range cannot be worse than the least valuable $k_i - k$ items in that range. Since $\pi^{(j-1)k_i}(M \setminus B_j))$ will be the same as $\pi^{jk_i}(M)$ only missing the k

items in B_j and the $k_i - k$ least valuable items in the jk_i range, this removed value will be less or equal to the value that agent i had for B_j, which was less than 1. Thus since less than 1 value is removed, it is true that $v_i(\pi^{(j-1)k_i}(M \setminus B_j)) \geq j-1$.

To prove that (ii) holds, we can simply observe that the algorithm consists of one for loop which iterates exactly n times, and inside it a while loop that iterates at most $|M|$ times, since there are at most $|M|$ items that can be swapped or added. Thus, the algorithm terminates in $O(mn)$ time. □

5 APS FOR CHORES

We start with defining relevant terms for chores and proving useful bounds on the APS value.

5.1 Definitions, and Bounds on APS

A chores instance is given by (N, M, D, K) where M is a set of chores, and $D = (d_1, \ldots, d_n)$ are the dis-utility functions of the agents. Each agent i derives *dis*-utility from chores, where $d_i(j) \geq 0$ is the dis-utility from chore j and $d_i(S) = \sum_{j \in S} d_i(S)$ is the dis-utility for subset $S \subseteq M$. They wish to minimize their dis-utility subject to *earning* constraint. Accordingly, the definition of APS extends to chores by interpreting the price vector as rewards that agents earn for completing chores. We interpret the entitlement as the earning requirement of the agent. Let $R = \{(r_1, r_2, \ldots r_m) \mid r_j \geq 0\, \forall j \in M, \sum_{j \in M} r_j = 1\}$ be the set of feasible rewards vectors. The AnyPrice share for chores can be defined as follows:

Definition 5.1 (AnyPrice Share for Chores)**.** The AnyPrice Share (APS) value of an agent i with entitlement b_i, denoted as $\text{APS}_i(b_i)$ is defined as following

$$\text{APS}_i := \max_{(r_1, r_2, \ldots r_m) \in R} \min_{S \subseteq M} \left\{ d_i(S) \mid \sum_{j \in S} r_j \geq b_i \right\}$$

In the case of goods, cardinality constraints modeled the "diminishing returns" of the agents as they received more goods. However, in the case of chores, an additional chore is more burdensome to an agent if they already have a lot of chores to do. To model this situation, the cardinality constraint lower bounds the number of chores an agent must receive. Note that when such cardinality constraints are imposed $|M| \geq \sum_{i \in N} k_i$ to ensure at least one feasible allocation exists.

Definition 5.2 (AnyPrice Share Under Heterogeneous Cardinality Constraint)**.** The APS value of an agent i with entitlement b_i and cardinality constraint k_i is defined as

$$APS_i := \max_{(r_1, r_2, \ldots r_m) \in R} \min_{S \subseteq M} \left\{ d_i(S) \mid \sum_{j \in S} r_j \geq b_i, |S| \geq k_i \right\}$$

While the notion of APS is defined for arbitrary entitlements, in the following sections, we only consider the setting in which all agents have equal entitlements (i.e. $b_i = \frac{1}{n}$) since there are several nice properties when we restrict to this case. We prove the properties pertinent to our algorithms in the remainder of this section.

MMS values. MMS value of an agent i is defined symmetrically to the goods case: $MMS_i = \min_{A \in \Pi_n(i, M)} \max_{A_j \in A} d_i(A_j)$, where $\Pi_n(i, M)$ is the set of all the partitions of set M in n bundles such that each bundle has size at least k_i.

APS (MMS) Allocation. An allocation A is said to be an APS (MMS) allocation if for every agent i, $|A_i| \geq k_i$ and $d_i(A_i) \leq APS_i$ ($d_i(A_i) \leq MMS_i$). It is an α-APS (α-MMS) allocation for an $\alpha \geq 1$ if for each $i \in N$, $|A_i| \geq k_i$ and $d_i(A_i) \leq \alpha * APS_i$ ($d_i(A_i) \leq \alpha * MMS_i$). As for the goods case [3], it follows that an α-APS allocation A is also an α-MMS allocation. Therefore, we will focus on finding α-APS allocation for chores, and the results extend to α-MMS allocations.

Similar to the goods' case, the valuation functions of the agents can be scaled so that $d_i(M) = n$ for all agent i and we can reduce an arbitrary instance to an ordered instance. Hence, we assume without loss of generality that the chores are ordered from the highest disutility to the lowest, $d_i(j) \geq d_i(k)\ \forall i \in N,\ \forall j, k \in M$ such that $j < k$.

Lemma 5.1. For any agent i, the APS value of the agent is at least their proportional share.

$$\text{APS}_i \geq \frac{1}{n} d_i(M)$$

Proof. Consider the reward vector given by $r_j = \dfrac{d_i(j)}{d_i(M)}$. Clearly, this reward vector is feasible. For any $S \subseteq M$ that satisfies the reward constraint, $\sum_{j \in S} r_j = \sum_{j \in S} \dfrac{d_i(j)}{d_i(M)} \geq \dfrac{1}{n}$. Hence, $d_i(S) \geq \dfrac{1}{n} d_i(M)$. Thus, $\min_{S \subseteq M} \left\{ d_i(S) \mid \sum_{j \in S} r_j \geq b_i \right\} \geq \dfrac{1}{n} d_i(M)$. Since APS takes the maximum over all reward vectors, the APS value of agent i must be at least the proportional share. □

Note that lemma 5.1 and scaling the valuations functions so that $d_i(M) = n$ imply that $\text{APS}_i \geq 1$ for all agents $i \in N$.

Lemma 5.2. The APS value of any agent $i \in N$ is greater than the disutility of any single chore $j \in M$.

$$\text{APS}_i \geq d_i(j)\ \forall i \in N, j \in M$$

In the case of cardinality constraints, $\text{APS}_i \geq d_i(j) + d_i(\pi^{k_i - 1}(M))$

Proof. We set the reward for chore j to 1 and 0 for any other chore. To satisfy the reward constraint, the agent must have chore j in their APS bundle. Hence, $\text{APS}_i \geq d_i(j)$. When a cardinality constraint is imposed, the APS bundle must contain k_i chores. The constraint can be satisfied by picking the $k_i - 1$ chores in M with the least disutility. □

Lemma 5.3. Let $S_l = \{ln - l + 1, ln - l \ldots ln + 1\}$ for any l such that $ln + 1 \leq |M|$

$$\text{APS}_i \geq d_i(S_l)$$

Proof. We assign a reward of $\frac{1}{ln+1}$ to the chores $1, 2, \ldots ln + 1$ and 0 to the rest. Any bundle that satisfies the reward constraint must contain at least $l + 1$ chores. Since S_l contains the $l + 1$ chores with the least disutility and non-zero reward, all bundles satisfying the reward constraint must have a higher disutility than S_l. Hence, the APS value for agent i must be greater than the disutility of S_l □

As a direct consequence of lemma 5.2 and 5.3, we have that $\text{APS}_i \geq d_i(1) \geq d_i(2) \cdots \geq d_i(n)$ and $\text{APS}_i \geq d_i(n) + d_i(n+1) \geq 2d_i(n+1)$. Hence, $\frac{\text{APS}_i}{2} \geq d_i(n+1) \geq d_i(n+2) \cdots \geq d_i(m)$.

Note that all proofs presented in this section are also valid for the APS under heterogeneous cardinality constraints.

5.2 2-APS for Chores

We present a simple round-robin-style algorithm that ensures that every agent receives at most twice their APS value under the assumption that $|M| \geq nk_{max}$ where $k_{max} = \max_{i \in N} k_i$.

Algorithm 3 2-APS Under Heterogeneous Cardinalities

Input: Scaled, Ordered Instance

$A \longleftarrow (\emptyset, \emptyset \ldots \emptyset)$ ▷ *Empty Allocation*
while $M \neq \emptyset$ **do**
  for $i = 1$ to n **do**
    Let $c = \text{argmax}_{j \in M}\, d_i(j)$
    $A_i \longleftarrow A_i \cup \{c\}$
    $M \longleftarrow M \setminus \{c\}$
    if $M = \emptyset$ **then**
      break

Theorem 5.1. *Algorithm 3* returns a 2-APS allocation under heterogeneous cardinality constraints for chores.

Proof. Since we assume that $|M| \geq nk_{max}$, there are at least k_{max} rounds of allocation. Every agent receives at least k_{max} chores and their cardinality constraint is satisfied. It remains to show that each agent values their bundle at most twice their APS value.

Consider $A_1 = \{1, n{+}1, \ldots (\lfloor \frac{m}{n} \rfloor - 1)n{+}1\}$ and $A_n = \{n, 2n, \ldots \lfloor \frac{m}{n} \rfloor n\}$. Since every agent picks their worst available chore and agent n has to pick last in every round, A_n contains the best chore (i.e. the chore with the lowest disutility) from every round. Thus, A_n is the best bundle amongst $A_1, A_2 \ldots A_n$ for all agents. By similar reasoning, A_1 is the worst. Since $A_1, A_2, \ldots A_n$ partition M, for any agent i, $nd_i(A_n) \leq d_i(A_1 \cup A_2 \cdots \cup A_n) = d_i(M) = n$. Hence, $d_i(A_n) \leq 1$. Since,

$$\begin{aligned} &d_i(n+1) \leq d_i(n), \\ &d_i(2n+1) \leq d_i(2n), \\ &\vdots \\ &d_i\left(\left(\left\lfloor \frac{m}{n} \right\rfloor - 1\right) n + 1\right) \leq d_i\left(\left(\left\lfloor \frac{m}{n} \right\rfloor - 1\right) n\right) \end{aligned}$$

implies $d_i(A_1 \setminus \{1\}) \leq d_i(A_n)$. Therefore,

$$\begin{aligned} d_i(A_i) &\leq d_i(A_n \cup \{1\}) \\ &\leq 1 + \text{APS}_i \\ &\leq 2\,\text{APS}_i\,. \end{aligned}$$

Hence, each agent values the bundle received by them at most $2\,\text{APS}_i$. □

5.3 (3/2)-APS for Chores Under Homogeneous Cardinality Constraints

Under the assumption that all agents have the same cardinality i.e. ($k_i = k$) and $d_{i1} - d_{in} \leq \frac{1}{2}$ (after ordering and scaling), we can provide a better approximation of the APS value to the agents.

To do so, we use a bag-filling style algorithm. We preallocate the n worst chores in separate bags. In each iteration, we consider a window of the $k-1$ worst chores that have not been assigned to a bag and add it to the current bag. If the value of the current bag is less than 1 for some agent, we expand this window until it has a value of at least 1 for every agent or the number of chores outside the window becomes too few and assign the bag to the agent who values it the least. If the value of the current bag is more than 3/2 for every agent, we slide the window until the value becomes less than 3/2 for some agent, at which point we assign it to the agent.

In the following algorithm, for any set of chores $S \subseteq M$ let $\pi^k(S), \mathcal{W}^k(S)$ denote the k chores with the lowest disutility in S and the highest disutility respectively. We omit the superscript when referring to the lowest or highest disutility chore in S.

Algorithm 4 (3/2)-APS Under Homogeneous Cardinalities

Input: Scaled, Ordered Instance with $k_i = k$, $|M| = nk$, $d_{i1} - d_{in} \leq \frac{1}{2}\ \ \forall i \in N$

1: $B_1 = \{1\}, B_2 = \{2\} \ldots B_n = \{n\}$
2: $A \longleftarrow (\emptyset, \emptyset \ldots \emptyset)$
3: **for** $r = 1$ to $n-1$ **do**
4:   $B_r \longleftarrow B_r \cup \mathcal{W}^{k-1}(M \setminus (B_r \cup B_{r+1} \cup \ldots B_n))$
5:   **if** $d_i(B_r) < 1$ for some agent $i \in N$ **then**
6:     **while** $d_i(B_r) < 1$ for some $i \in N$ and $|M \setminus B_r| > (n-r)k$ **do**
7:       Add $\mathcal{W}((M \setminus (B_r \cup B_{r+1} \cup \ldots B_n))$ to B_r
8:   **else**
9:     **while** $d_i(B_r) > 3/2\ \forall i \in N$ **do**
10:       Swap $c := \mathcal{W}(B_r \setminus \{r\})$ with $c' := \mathcal{W}(\{j \in M \setminus (B_r \cup B_{r+1} \cup \ldots B_n) : c < j\})$
11:   Find $i \in N$ s.t $d_i(B_r) \leq 3/2$
12:   $A_i \longleftarrow B_r$
13:   $M \longleftarrow M \setminus B_r$
14:   $N \longleftarrow N \setminus \{i\}$
15: Let $i' \in N$
16: $A_{i'} \longleftarrow M$
17: **return** A

Theorem 5.2. *Algorithm 4* terminates and returns a (3/2)-APS allocation under homogenous cardinality constraints for chores.

Proof. We first observe that at the start of any iteration r, $|M| \geq (n-r+1)k$. When $r=1$, this claim is trivially true. Let M_r and N_r denote the set of unallocated chores and remaining agents at the start of iteration r respectively. For any iteration $r > 1$, the size of the bag allocated is at most $|M_r| - (n-r)k$, which implies $|M_{r+1}| \geq (n-r)k$. Hence, the claim holds inductively.

Another useful observation is that for any bag B_r generated by the *while* loop in lines 9–10, $d_i(B_r) \geq 1$ for any agent $i \in N_{r'}\ \ \forall r' > r$. Assume to the contrary that there exists some agent $j \in N'_r$ for $r' > r$ such that $d_j(B_r) < 1$. Let $c' \in B_r$ and $c \in M_{r+1}$ be the last chores swapped by the *while* loop. Agent j's value for B_r before the swap must be less than $3/2$ since $d_j(c) - d_j(c') \leq 1/2$ by a corollary of lemma 5.2. Thus, the *while* loop would have terminated before swapping c and c' which is a contradiction. A direct consequence of this observation is that if for some agent $i \in N_{r'}$, $d_i(B_r) < 1$ for some $r < r'$ implies B_r was produced through the *while* loop in lines $6 - 7$.

To show that every agent receives at least k chores, we note that B_r contains k items after line 4. Throughout the *for* loop, we either add or swap chores B_r. Thus, if iteration r terminates, then the bag produced has at least k items. If all the iterations terminate, then by the first observation above $|M_n| \geq k$.

We argue that every iteration terminates. During iteration r, the algorithm either enters the *while* loop in lines $6 - 7$ or the *while* loop in $9 - 10$. The first *while* loop is guaranteed to terminate since $|B_r|$ can only grow until it reaches $|M| - (n - r)k$.

If we enter the second *while* loop, then $d_i(B_r) > 1$ for every agent $i \in N_r$. We claim that this implies $d_i(M_r) \leq n - r + 1$ for every $i \in N_r$. For the sake of contradiction, assume $d_i(M_r) > n - r + 1$. This is only possible if for some $r' < r$, $d_i(B_{r'}) < 1$. As argued above, this implies $B_{r'}$ was produced by the first *while* loop. Since $d_i(r') > d_i(r)$ and $B_{r'}$ contained the worst $|B'_r| - 1$ goods in $M \setminus (B_{r'} \dots B_n)$ implies $d_i(B_r) < d_i(B_{r'}) < 1$. This is a contradiction.

Using the facts that $|M_r| \geq (n - r + 1)k$ and $d_i(M_r) \leq n - r + 1$, we argue that the second *while* loop must terminate. Consider the set $S_r := \{r\} \cup \Pi^{k-1}(M_r)$. There are at least $(n - r + 1)$ disjoint sets of size k in $M_r \setminus \{r, r+1, \dots n\}$ with value at least as high as $\Pi^{k-1}(M_r)$. For any agent i,

$$d_i(M_r) = d_i(M_r \setminus \{r, r+1, \dots n\}) + \sum_{c=r}^{n} d_i(c) + \sum_{c=r}^{n} d_i(c) - d_i(r)$$

If $d_i(S_r) \geq 3/2$ for some agent $i \in N_r$ then,

$$d_i(M_r \setminus \{r, r+1, \dots n\}) + \sum_{c=r}^{n} d_i(c) \geq \frac{3}{2}(n - r + 1)$$

By our assumption, $d_i(1) - d_i(n) < 1/2 \implies$

$$\sum_{c=r}^{n} d_i(1) - d_i(r) \leq \frac{1}{2}(n - r + 1)$$

$$d_i(M_r) > \frac{3}{2}(n - r + 1) - \frac{1}{2}(n - r + 1) = n - r + 1$$

But this is a contradiction. Hence S_r must have a value of at most $3/2$ for every agent in N_r. Thus the second *while* loop is guaranteed to terminate when $B_r = S_r$.

It only remains to be shown that every agent receives a bundle of value at most $3/2$. It is clear that the bag received by an agent through the *for* loop has a value of at most $3/2$ for the agent. Let i' be the last agent to receive an allocation. If in every iteration, $d_{i'}(B_r) > 1$, then $d_{i'}(M_n) \leq (n - n + 1) \leq 1$. Otherwise, for some $r' < n$, $d_{i'}(B_{r'}) < 1$. This implies that $d_{i'}(\{r'\} \cup \mathcal{W}^{k-1}(M_{r'} \setminus (B_{r'} \cup \dots B_n))) < 1$. There is a bijection f from M_n to $\{r'\} \cup \mathcal{W}^{k-1}(M_{r'} \setminus (B_{r'} \cup \dots B_n))$ such that $d_{i'}(f(c)) < d_{i'}(c)$ for every $c \in M_n$. Hence, $d_{i'}(M_n) < 1$. Thus, the bundle received by every agent has a value of at most $3/2$ for them. □

6 BEST-OF-BOTH-WORLDS (BOBW) GUARANTEES

In this section, we extend our algorithms to *best of both worlds* setting in which we want to provide agents with both ex-ante and ex-post guarantees over randomized allocations. In the *best of both worlds under cardinality constraints* setting, the guarantees must be with respect to the final valuation functions f_i's in the goods' case. In the case of chores, the ex-post allocation must satisfy each agent's cardinality constraints.

BoBW for Goods. We present a round-robin-based algorithm to generate a distribution (with linear size support) over allocations that guarantees every agent their proportional share ex-ante and $(1/2) -$ APS ex-post. We invoke round-robin as a subroutine to generate allocations. In addition to passing N and M as arguments to round-robin, we also pass a tuple $(i_1, i_2, \dots i_n)$ where $i_k \in N \; \forall k$ to specify the order in which the agents pick goods. As before, we assume that the instance is ordered and the valuation functions are scaled so that $v_i(\Pi^{k_i n}(M)) = n$ for every agent. We also assume that $|M| > nk_{max}$ where $k_{max} = \max K$ without loss of generality since we can always pad the instance with zero-valued goods.

Algorithm 5 Ex-ante Prop and Ex-post (1/2)-APS under Cardinality Constraints

Input: Scaled, Ordered Instance with $|M| > nk_{max}$
  $r \leftarrow (1, 2, \dots n)$ ▷ *Initial Ordering of Agents*
  $D \leftarrow \emptyset$ ▷ *Empty Distribution*
  for $k = 1$ to n **do**
    $A_k \leftarrow$ Round-Robin(N, M, r)
    Insert $(1/n, A_k)$ into D
    Cyclically permute r
  return D

Theorem 6.1. Given an instance (N, M, V, K) with goods and heterogeneous cardinality constraints, the distribution returned by *Algorithm* 5 guarantees every agent their proportional share ex-ante and $(1/2) -$ APS ex-post.

Proof. Let A_{ki} refer to the bundle received by agent i in allocation A_k. Since we go through all cyclic permutations of $(1, 2 \dots n)$ and the goods are ordered, every good $j \in M$ belongs to A_{ki} for exactly one value of k. Hence, in expectation, agent i receives a $1/n$ fraction of every good. Clearly, this fractional allocation is proportional. As argued in section 3, every allocation in D is $(1/2) -$ APS for all agents. Hence, the ex-post guarantee is also satisfied. □

BoBW for Chores. Since this round-robin algorithm makes no assumption about the ordering of the agents, we note that it can be extended to the *best of both worlds* guarantee for chores under the heterogeneous cardinality constraints, in the same way as the goods' case. In particular, we get the following:

Theorem 6.2. For a chores instance (N, M, D, K) with heterogeneous cardinality constraints, there exists an algorithm that guarantees every agent their proportional share ex-ante and 2-APS ex-post.

REFERENCES

[1] Hannaneh Akrami and Jugal Garg. *Breaking the* 3/4 *Barrier for Approximate Maximin Share.* 2023. arXiv: 2307.07304 [cs.GT].

[2] Georgios Amanatidis et al. "Fair Division of Indivisible Goods: A Survey". In: *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022.* Ed. by Luc De Raedt. ijcai.org, 2022, pp. 5385–5393. doi: 10.24963/ijcai.2022/756. url: https://doi.org/10.24963/ijcai.2022/756.

[3] Moshe Babaioff, Tomer Ezra, and Uriel Feige. "Fair-Share Allocations for Agents with Arbitrary Entitlements". In: *EC '21: The 22nd ACM Conference on Economics and Computation, Budapest, Hungary, July 18-23, 2021.* Ed. by Péter Biró, Shuchi Chawla, and Federico Echenique. ACM, 2021, p. 127. doi: 10.1145/3465456.3467559. url: https://doi.org/10.1145/3465456.3467559.

[4] Siddharth Barman and Sanath Kumar Krishna Murthy. "Approximation Algorithms for Maximin Fair Division". In: *Proceedings of the 2017 ACM Conference on Economics and Computation.* 2017, pp. 647–664.

[5] Siddharth Barman and Paritosh Verma. "Existence and computation of maximin fair allocations under matroid-rank valuations". In: *arXiv preprint arXiv:2012.12710* (2020).

[6] Sylvain Bouveret and Michel Lemaître. *Efficiency and Sequenceability in Fair Division of Indivisible Goods with Additive Preferences.* 2016. arXiv: 1604.01734 [cs.GT].

[7] Eric Budish. "The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes". In: *Journal of Political Economy* 119.6 (2011), pp. 1061–1103.

[8] Ioannis Caragiannis et al. "The Unreasonable Fairness of Maximum Nash Welfare". In: *Proceedings of the 2016 ACM Conference on Economics and Computation, EC '16, Maastricht, The Netherlands, July 24-28, 2016.* ACM, 2016, pp. 305–322. doi: 10.1145/2940716.2940726. url: https://doi.org/10.1145/2940716.2940726.

[9] Bhaskar Ray Chaudhury, Jugal Garg, and Kurt Mehlhorn. "EFX Exists for Three Agents". In: *21st.* ACM, 2020, pp. 1–19.

[10] Uriel Feige, Ariel Sapir, and Laliv Tauber. "A Tight Negative Example for MMS Fair Allocations". In: *Web and Internet Economics - 17th International Conference, WINE 2021, Potsdam, Germany, December 14-17, 2021, Proceedings.* Ed. by Michal Feldman, Hu Fu, and Inbal Talgam-Cohen. Vol. 13112. Lecture Notes in Computer Science. Springer, 2021, pp. 355–372. doi: 10.1007/978-3-030-94676-0_20. url: https://doi.org/10.1007/978-3-030-94676-0%5C_20.

[11] 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.

[12] Jugal Garg, Peter McGlaughlin, and Setareh Taki. "Approximating Maximin Share Allocations". In: *2nd Symposium on Simplicity in Algorithms (SOSA 2019).* Ed. by Jeremy T. Fineman and Michael Mitzenmacher. Vol. 69. OpenAccess Series in Informatics (OASIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2018, 20:1–20:11. isbn: 978-3-95977-099-6. doi: 10.4230/OASIcs.SOSA.2019.20. url: http://drops.dagstuhl.de/opus/volltexte/2018/10046.

[13] Mohammad Ghodsi et al. "Fair allocation of indivisible goods: Improvements and generalizations". In: *Proceedings of the 2018 ACM Conference on Economics and Computation.* 2018, pp. 539–556.

[14] Pranay Gorantla, Kunal Marwaha, and Santhoshini Velusamy. "Fair allocation of a multiset of indivisible items". In: *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023.* Ed. by Nikhil Bansal and Viswanath Nagarajan. SIAM, 2023, pp. 304–331. doi: 10.1137/1.9781611977554.ch13. url: https://doi.org/10.1137/1.9781611977554.ch13.

[15] Halvard Hummel and Magnus Lie Hetland. "Guaranteeing Half-Maximin Shares Under Cardinality Constraints". In: *21st International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2022, Auckland, New Zealand, May 9-13, 2022.* Ed. by Piotr Faliszewski et al. International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS), 2022, pp. 1633–1635. doi: 10.5555/3535850.3536059. url: https://www.ifaamas.org/Proceedings/aamas2022/pdfs/p1633.pdf.

[16] Rucha Kulkarni, Ruta Mehta, and Setareh Taki. "Indivisible mixed manna: On the computability of mms+ po allocations". In: *Proceedings of the 22nd ACM Conference on Economics and Computation.* 2021, pp. 683–684.

[17] Zhentao Li and Adrian Vetta. "The fair division of hereditary set systems". In: *ACM Transactions on Economics and Computation (TEAC)* 9.2 (2021), pp. 1–19.

[18] Richard J. Lipton et al. "On approximately fair allocations of indivisible goods". In: *5th.* 2004, pp. 125–131.

[19] Vasilis Livanos, Ruta Mehta, and Aniket Murhekar. "(Almost) Envy-Free, Proportional and Efficient Allocations of an Indivisible Mixed Manna". In: *21st International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2022, Auckland, New Zealand, May 9-13, 2022.* Ed. by Piotr Faliszewski et al. International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS), 2022, pp. 1678–1680. doi: 10.5555/3535850.3536074. url: https://www.ifaamas.org/Proceedings/aamas2022/pdfs/p1678.pdf.

[20] Benjamin Plaut and Tim Roughgarden. "Almost Envy-Freeness with General Valuations". In: *SIAM J. Discret. Math.* 34.2 (2020), pp. 1039–1068.

[21] Ariel D. Procaccia and Junxing Wang. "Fair enough: guaranteeing approximate maximin shares". In: *ACM Conference on Economics and Computation, EC '14, Stanford , CA, USA, June 8-12, 2014.* Ed. by Moshe Babaioff, Vincent Conitzer, and David A. Easley. ACM, 2014, pp. 675–692. doi: 10.1145/2600057.2602835. url: https://doi.org/10.1145/2600057.2602835.

[22] Gilad Ben Uziahu and Uriel Feige. "On fair allocation of indivisible goods to submodular agents". In: *arXiv preprint arXiv:2303.12444* (2023).

[23] Vignesh Viswanathan and Yair Zick. "Yankee swap: a fast and simple fair allocation mechanism for matroid rank valuations". In: *arXiv preprint arXiv:2206.08495* (2022).

A PROOF OF LEMMA 4.2

Proof. To prove this claim we will use induction on the number of agents, n.

When $n = 1$, we can see that $b_i = \frac{1}{n} = 1$, and since $\sum_{j \in M} p_j = 1$, when $S = \pi_i^{k_i}(M)$, $\sum_{j \in S} p_j \leq 1 = b_i$, so $f_i(S) = v_i(\pi_i^{k_i}(M))$. Since S is a bundle of the k_i best items, this is certainly the optimal bundle, and indeed consists of only of items in the $\pi_i^{nk_i}(M)$ items.

Assuming the claim holds for all $1 \leq n' < n$, we now prove that the claim also holds for n.

Assume towards contradiction that this is false, and so the optimal bundle of affordable items, S^*, consist of at least one item not in the top nk_i items. Note that we assume $|S^*| \leq k_i$, as we can only value the top k_i items of any bundle $S \subseteq M$, and so excluding all items past the top k_i items does not change objective value. Let $S_1^* = \pi_i^{nk_i}(M) \cap S*$ and let $S_2^* = S^* \setminus \pi_i^{nk_i}(M)$, so we see that $S^* = S_1^* \cup S_2^*$ and $S_1^* \cap S_2^* = \emptyset$.

Let $(p_1, p_2, \ldots, p_m)$ be an adversarial price vector that was used to minimize the value of the maximum bundle S^*. Construct a price vector for an instance of size $n-1$ as follows: Let the items $M' = M \setminus B$ where B is a set of k_i items in the top nk_i items of M whose prices sum to at least $\frac{1}{n}$. Such a set B is guaranteed to exist because if all n-partitions $(X_1, X_2, \ldots, X_n)$ of $\pi_i^{nk_i}(M)$ with $|X_1| = |X_2| = \cdots = |X_n| = k_i$ had pricing such that $\sum_{j \in X_i} p_j < \frac{1}{n}$, then it would be true that $\sum_{j \in \pi_i^{nk_i}(M)} p_j < 1$, however since

$$p_j = 0 \forall j \in M \setminus \pi_i^{nk_i}(M), \sum_{j \in M} p_j = \sum_{j \in \pi_i^{nk_i}(M)} p_j = 1,$$

which is a contradiction. The remaining prices for items in M' will be less than $\frac{n-1}{n}$ since $\sum_{j \in \pi_i^{nk_i}(M)} p_j = 1$ and $\sum_{j \in B} p_j \geq \frac{1}{n}$ implies

$$\sum_{j \in \pi_i^{nk_i}(M) \setminus B} p_j = \sum_{j \in \pi_i^{nk_i}(M')} p_j \leq 1 - \frac{1}{n} = \frac{n-1}{n}.$$

Let the new pricing $p_j' = p_j \frac{n}{n-1}$ for all $j \in M'$, so $\sum_{j \in M'} p_j \leq 1$.

Since S^* contained items past the top nk_i items, $|M| > nk_i$, and so $|M'| = |M| - k_i > (n-1)k_i$, and since $|S^*| \leq k_i$, and $n \geq 2$, $(n-1) \geq 1$, so $|M'| > (n-1)k_i \geq |S^*|$. Thus, it must be true that there is an item in M' that is not in S^*. If the only such items were not in the top $(n-1)k_i$ items, then it would have to be true that S^* is exactly the top $(n-1)k_i$ items of M', in which case S^* would have had to be in the top nk_i items of M which we know to be false, so there must be an item in $\pi_i^{(n-1)k_i}(M')$ that is not in S^*. Choose one such item, $q \in \pi_i^{(n-1)k_i}(M') \setminus S^*$, and set $p_q' = p_q' + 1 - \sum_{j \in M'} p_j$. Now, we have $\sum_{j \in M} p_j' = 1$.

For all $j \in M' \setminus \pi_i^{(n-1)k_i}(M')$, it must also be true that $j \in M \setminus \pi_i^{nk_i}(M)$ since M' is exactly M with k_i top items removed. Thus, it must be true that $p_j = 0$, and so $p_j' = 0 \frac{n}{n-1} = 0$ because the only new price otherwise adjusted is q, which is not in $M' \setminus \pi_i^{(n-1)k_i}(M')$. Thus we have a price assignment $(p_1', p_2', \ldots, p_{m'}')$ such that

$$p_j' \geq 0 \forall j \in M', p_j' = 0 \forall j \in M' \setminus \pi_i^{(n-1)k_i}(M'), \sum_{j \in M'} p_j' = 1.$$

Any bundle of items $S \subseteq M' \setminus \{q\}$ such that $\sum_{j \in S} p_j' \leq \frac{1}{n-1}$ satisfies $\sum_{j \in S} p_j \frac{n}{n-1} \leq \frac{1}{n-1}$ or $\sum_{j \in S} p_j \leq \frac{1}{n}$. A bundle S that contains q may no longer affordable. Thus, a bundle that is affordable in the $n-1$ case was also affordable in the n case. Since the valuation of the bundles remains the same, and every bundle $S \subseteq M'$ is also a subset of M, the maximally valued affordable bundle in the n case (which is still contained in M') must also be the maximally valued affordable bundle in the $n-1$ case. Finally we can see that this implies the solution to the $n-1$ case contains items from outside the top $(n-1)k_i$ items, which contradicts our inductive hypothesis. Thus, our initial assumption must be false, and the claim must also hold for n.

□