



Approximating APS under Submodular and XOS Valuations with Binary Marginals

Pooja Kulkarni

University of Illinois at
Urbana-Champaign
Champaign, United States
poojark2@illinois.edu

Rucha Kulkarni

University of Illinois at
Urbana-Champaign
Champaign, United States
ruchark2@illinois.edu

Ruta Mehta

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

ABSTRACT

We study the problem of fairly dividing indivisible goods among a set of agents under the fairness notion of Any Price Share (APS). APS is known to dominate the widely studied Maximin share (MMS). Since an exact APS allocation may not exist, the focus has traditionally been on the computation of approximate APS allocations. [4] studied the problem under additive valuations, and asked (i) how large can the APS value be compared to the MMS value? and (ii) what guarantees can one achieve beyond additive functions. We partly answer these questions by considering valuations beyond additive, namely submodular and XOS functions, with binary marginals.

For the submodular functions with binary marginals, also known as matroid rank functions (MRFs), we show that APS is exactly equal to MMS. Consequently, following [5] we show that an exact APS allocation exists and can be computed efficiently while maximizing the social welfare. Complementing this result, we show that it is NP-hard to compute the APS value within a factor of $5/6$ for submodular valuations with three distinct marginals of $\{0, \frac{1}{2}, 1\}$.

We then consider binary XOS functions, which are immediate generalizations of binary submodular functions in the complement free hierarchy. In contrast to the MRFs setting, MMS and APS values are not equal under this case. Nevertheless, we can show that they are only a constant factor apart. In particular, we show that under binary XOS valuations, $MMS \leq APS \leq 2 \cdot MMS + 1$. Further, we show that this is almost the tightest bound we can get using MMS, by giving an instance where $APS \geq 2 \cdot MMS$. The upper bound on APS, combined with [17], implies a 0.1222-approximation for APS under binary XOS valuations. And the lower bound implies the non-existence of better than 0.5-APS even when agents have identical valuations, which is in sharp contrast to the guaranteed existence of exact MMS allocation when agent valuations are identical.

KEYWORDS

Fair Division; Approximation Algorithms; Matroid Structures; XOS Valuations



This work is licensed under a Creative Commons Attribution International 4.0 License.

Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), N. Alechina, V. Dignum, M. Dastani, J.S. Sichman (eds.), May 6–10, 2024, Auckland, New Zealand. © 2024 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

ACM Reference Format:

Pooja Kulkarni, Rucha Kulkarni, and Ruta Mehta. 2024. Approximating APS under Submodular and XOS Valuations with Binary Marginals. In *Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), Auckland, New Zealand, May 6 – 10, 2024, IFAAMAS, 9 pages.*

1 INTRODUCTION

Finding fair allocations of indivisible resources is a central problem within economics, game theory, social choice theory, and computer science. Given a set $[n]$ of agents and a set $[m]$ of indivisible items, the problem asks to partition the items among the agents in a *fair* manner. Preferences of each agent $i \in [n]$ for bundles of goods are represented by monotone valuation functions $v_i : 2^{[m]} \rightarrow \mathbb{R}_+$. A formal study of fair division began with the work of [21]. Since then, several notions of fairness like MMS, EF1 and Prop1 have been introduced and are well-studied (See [1, 2] for surveys on these).

Any Price Share (APS) is one of the more recently introduced [4] notions, and has already garnered significant interest (See Section 1.1 for a brief review). The APS value of an agent is defined as the maximum value she can obtain with a budget of $1/n$, given any vector of prices of goods that sums to 1. An allocation where every agent gets at least her APS value is called an *APS allocation*. An attractive feature of APS is that it is independent of the valuations of the other agents. This is the same with the well-studied notion of maximin share (MMS) [8, 13, 14, 17]. The APS value of any agent is known to dominate their MMS value [4]. As MMS allocations are known to not exist [18] even in the additive valuations case¹, the same holds for APS allocations. We, therefore, focus on the problem of finding approximate APS, or α -APS allocations, which give every agent a bundle of value at least α times their APS, for some $\alpha > 0$.

The problem of finding approximate APS allocations has been studied for the case of additive valuations [4] and more recently for submodular valuations [9]. Numerous real world applications of fair division, like public housing to ethnic minorities, assigning kindergarten slots, course seat assignments for classes with capacity constraints and where students can specify preferences for a fixed maximum number of classes, require the valuation functions to be beyond additive and capture a *diminishing marginal returns* property: this essentially means that the marginal value of a good over a set of goods diminishes over supersets of the set. Submodular functions capture this very natural property and are therefore

¹the value of a set equals the sum of values of goods in the set.

considered a fundamental class of valuations. Fractionally subadditive (XOS) functions are immediate generalizations of submodular functions in the complement-free hierarchy. We study APS under both these function classes, with the constraint that the marginal values are *binary*, meaning the marginal value of any good over any subset of goods is either zero or one. These classes have a rich structure and have been well-studied for other fairness notions and also in optimization theory (See Section 1.1 for a brief review). We partly resolve the following questions posed in [4] for submodular and XOS functions with binary marginal values.

Question 1: How far apart can the MMS and APS values of an agent be in any instance?

Question 2: What guarantees for APS can we ensure beyond additive valuations?

Submodular functions with binary marginal values are equivalently known as Matroid rank functions (MRFs), and are widely studied, for instance [8, 10, 19, 20]. We show the following surprising result. Although the APS value is known to be strictly higher than the MMS value even for general additive functions [4], we show that for any MRF, these two notions are equivalent. This immediately leads to polynomial time algorithms to obtain exact APS allocations using the algorithms that obtain exact MMS allocations [8]. In fact, the known algorithms also ensure economic efficiency, by giving APS allocations that simultaneously maximize social welfare (the total sum of values received by agents).

We then analyze the classic generalization of binary submodular functions, namely fractionally subadditive functions (XOS) with binary marginals. For this, we show that the APS value of an agent with such a valuation function is at most $2\mu + 1$, where μ is her MMS value. Using this fact together with 0.3666-MMS allocation computing algorithm by [16] we obtain an efficient algorithm for computing 0.1222-APS allocation. In contrast to the relation between MMS and APS for MRFs, we show that there exist instances with only two agents and identical valuations in this setting such that $APS \geq 2MMS$. As a consequence we get that even under identical valuations with two agents, better than 0.5-APS allocation may not exist. This is in sharp contrast to MMS where by definition an exact MMS allocation exists when agents have identical valuations. Finally, we show that if binary submodular functions are generalized to allow three distinct marginal values, in $\{0, 1/2, 1\}$, instead of the two values $\{0, 1\}$ in MRFs, then the problem of computing α -APS allocations, even among agents with identical valuation functions, for any factor α better than $5/6$, is NP-hard. Our results can be summarized as follows.

- APS = MMS for submodular functions with binary marginals. Exact APS values and allocations that give APS along with maximum social welfare can thus be efficiently computed.
- APS $\leq 2 \cdot MMS + 1$, for XOS functions with binary marginals. A 0.1222-APS allocation can thus be efficiently computed.
- There exist instances with identical binary, XOS valuations where 0.5-APS allocation does not exist.
- Submodular functions with ternary marginals: computing APS values approximately to a factor better than $5/6$ is NP-hard.

1.1 Further Related Work

APS. APS was introduced in [4], who also prove that APS dominates MMS for non-negative valuations. Further, they give 0.667-APS allocation for goods, and a 2-APS allocation for chores, under additive valuations. [15] study the connection of the known notion of Propx with APS for chores, giving a 2-APS allocation here. [11] compare APS with other *share based* notions for the case of agents with asymmetric entitlements. [12] study group fairness guarantees with APS, under additive valuations. [9] recently gave a $\frac{1}{3}$ -approximate algorithm for computing APS with submodular valuations for asymmetric agents i.e., when agents have different entitlements.

Matroid Rank Functions. Rank functions of matroid are one of the fundamental set functions and the optimization of these functions has been studied in detail, see [19]. [10] and [6] identify multiple domains where matroid rank functions show up naturally like fair allocation of public housing units. These functions have been studied in context of fair division, for other fairness and efficiency like Nash Social Welfare² [3] EF1³[4, 10], MMS [6] and combinations of these [22]. Notably, polynomial time algorithms that output the optimal allocations under all these fairness notions are known in the respective works.

Binary XOS valuations. Binary XOS valuations generalize matroid rank functions and have been studied in fair division context. [16] give an algorithm that gives a 0.3667-MMS. [7] give a 288-approximation algorithm for maximizing Nash social welfare under these valuations.

2 NOTATION AND PRELIMINARIES

Notation. $[k]$ denotes the set $\{1, 2, \dots, k-1, k\}$.

Model. We study the problem of fairly dividing a set of m indivisible goods, among n agents. Preferences of an agent $i \in [n]$ is defined by a valuation function $v_i : 2^{[m]} \rightarrow \mathbb{R}_{\geq 0}$ over the set of goods. We use $v(g|A_i \setminus \{g\})$ to denote the marginal increase in value when good g is added to the set A_i , i.e., $v(g|A_i \setminus \{g\}) = v(A_i \cup g) - v(A_i)$. We represent a fair division problem instance by $(([n], [m], (v_i)_{i \in [n]})$.

Allocations. An allocation, $\mathcal{A} := (A_1, \dots, A_n)$ is a partition of all the goods among the n agents, i.e. for all $i, j \in [n]$ with $i \neq j$, $A_i \cap A_j = \emptyset$ and $\cup_{i \in [n]} A_i = [m]$. We denote the set of all allocations by $\Pi_{[n]}([m])$. We also define a *partial allocation*, denoted by $\mathcal{P} = (P_1, \dots, P_n)$, as a partition of any subset of goods, that is, where $P_i \cap P_j = \emptyset$ for all $i \neq j$ and $\cup_{i \in [n]} P_i \subseteq [m]$. Finally, we use the notion of *non-wasteful allocations* also defined in [6]. These are allocations where the marginal utility of all the goods in every bundle is non-zero, that is, for such an allocation \mathcal{A} , $v(g|A_i \setminus \{g\}) > 0$ for all $i \in [n]$ and any $g \in A_i$. We now define the fairness notions we use in this work.

2.1 Fairness Notions

Any Price Share (APS). Let \mathcal{P} denote the simplex of price vectors over the set of goods $[m]$, formally, $\mathcal{P} = \{(p_1, \dots, p_m) \geq$

²Nash welfare is the geometric mean of agent's valuations

³An agent values her bundle more than other agent's bundle up to removal of one (some) good

$0 \mid \sum_i p_i = 1\}$. Informally, Any Price Share is the value that an agent can guarantee themselves at *any* price with the budget of $\frac{1}{n}$. Formally, for an instance $([n], [m], (v_i)_{i \in [n]})$, the APS value of agent i is defined as,

$$\text{APS}_i^n([m]) := \min_{p \in \mathcal{P}} \max_{S \subseteq [m], p(S) \leq \frac{1}{n}} v_i(S) \quad (1)$$

where $p(S)$ is the sum of prices of goods in S . We will refer $\text{APS}_i^{[n]}([m])$ by APS_i when the qualifiers n and m are clear.

An alternate definition without using prices is as follows.

Definition 2.1 (Any Price Share). *The APS value of an agent i for an instance $([n], [m], (v_i)_{i \in [n]})$ is the solution of the following program.*

$$\begin{aligned} \text{APS}_i &= \max z \\ \text{subject to: } & \sum_{T \subseteq [m]} \lambda_T = 1 \\ & \lambda_T = 0 \quad \forall T \text{ such that } v_i(T) < z \\ & \sum_{T \subseteq [m]: j \in T} \lambda_T \leq \frac{1}{n} \quad \forall j \in [m] \\ & \lambda_T \geq 0 \quad \forall T \subseteq [m] \end{aligned}$$

Essentially, an agent must decide the maximum value z that satisfies the following. They associate non-zero weights λ_T to all the sets $T \subseteq [m]$ such that any set with a value less than z has weight zero, the sum of the weights on all the sets is 1, and the total weight on any good, defined as the sum of weights of the sets containing the good, is at most $\frac{1}{n}$. This maximum value of z is their APS value. Both of these definitions and their equivalence is stated in [4].

Maximin Share (MMS). The Maximin share (MMS) value of an agent i for an instance $([n], [m], (v_i)_{i \in [n]})$ is defined as the minimum value they can guarantee while partitioning all the goods into n bundles, assuming they pick the worst bundle in any allocation. Formally,

$$\text{MMS}_i^n([m]) = \max_{(A_1, \dots, A_n) \in \Pi_{[n]}([m])} \min_{k \in [n]} v_i(A_k).$$

We refer to $\text{MMS}_i^n([m])$ by MMS_i when the qualifiers n and m are clear. We will refer to the allocation that defines the MMS value of any agent i , that is, $\text{argmax}_{\Pi_{[n]}([m])} \min_{k \in [n]} v_i(A_k)$, as the *MMS defining allocation* of agent i .

Note that both the APS and MMS values of an agent do not depend on the valuation functions of the other agents, and depend only on the number of agents in the fair allocation instance.

The following relation between the APS and MMS values of any agent is known.

Claim 2.1. [4] For any monotone valuation function v_i of agent i , we have $\text{APS}_i \geq \text{MMS}_i$.

At times we abuse notation, and refer as *the APS (or MMS) value of a function $v(\cdot)$* , which essentially is a value of an agent whose valuation function is $v(\cdot)$.

2.2 Valuation Functions

Binary marginals. For a valuation function $v(\cdot)$, the *marginal utility* of a good $g \in [m]$ over a set $S \subseteq [m]$, denoted by $v(g|S)$, is defined as the increase in the total value of the bundle $S \cup \{g\}$ over the set S , that is, $v(g|S) = v(S \cup \{g\}) - v(S)$. We consider valuation functions with *binary* marginals, defined as those for whom the marginal utility of any good over any set is either 0 or 1, that is, $v(g|S) \in \{0, 1\}$ for every $g \in [m]$ and $S \subseteq [m]$.

Submodular functions. A function is called *submodular* if it satisfies the property of *diminishing marginal returns*, which specifies that the marginal utility of any good g over any subset of goods S must not be larger than its marginal utility over any subset of S . Formally, a function $v : 2^{[m]} \rightarrow \mathbb{R}_{\geq 0}$ is called submodular if and only if,

$$v(g|S) \leq v(g|S'), \quad \forall g \in [m], S' \subseteq S \subseteq [m].$$

XOS functions. A function $v : 2^{[m]} \rightarrow \mathbb{R}_{\geq 0}$ is called *additive*, if the value of a set of goods is equal to the sum of values of the goods in the set, that is, $v(S) = \sum_{g \in S} v(\{g\})$. A function $v : 2^{[m]} \rightarrow \mathbb{R}_{\geq 0}$ is said to be *XOS*, or *fractionally subadditive*, if and only if there exists a family of additive set functions \mathcal{F} , such that the value of each subset $S \subseteq [m]$ is the maximum function value of S from the functions in \mathcal{F} , that is, $v(S) = \max_{f \in \mathcal{F}} f(S)$. Note that the cardinality of the family \mathcal{F} can be exponentially high in m .

We focus on submodular and XOS functions with binary marginals. In case of the submodular functions, these are equivalent to what are called *matroid rank functions* (MRFs) (See Section 2.3 for a matroid based definition of these functions).

For the submodular functions, we also study the case of *ternary marginals*, where all the marginal values $v(g|S) \in \{0, 1/2, 1\}$ for all goods $g \in [m]$ and $S \subseteq [m]$.

2.3 Matroid Preliminaries

Matroid. A matroid, denoted by \mathcal{M} , is a tuple (E, \mathcal{I}) where E is a set of elements, called the *ground set*, and $\mathcal{I} \subseteq 2^E$ is a collection of subsets of E called the *independent sets* of the matroid, that satisfies the following properties.

- (1) If $S \in \mathcal{I}$ then $\bar{S} \in \mathcal{I}$ for all $\bar{S} \subseteq S$.
- (2) If $I, J \in \mathcal{I}$ and $|I| > |J|$ then there exists $i \in \{I \setminus J\}$ such that $J \cup \{i\} \in \mathcal{I}$.

Bases of a matroid. Any independent set of the largest cardinality, that is any set $B \in \text{argmax}_{I \in \mathcal{I}} |I|$, is called a *base* of the corresponding matroid.

Rank of a matroid. Every matroid $\mathcal{M} = (E, \mathcal{I})$ has what is called a *rank function* associated with it, that maps any subset S of the ground set E to a non-negative integer, equal to the size of the largest independent set that is a subset of S . We denote this function by $r_{\mathcal{M}} : 2^E \rightarrow \mathbb{Z}_{\geq 0}$. Formally,

$$r_{\mathcal{M}}(S) := \max_{I \subseteq S, I \in \mathcal{I}} |I| \quad (2)$$

The *rank* of a matroid \mathcal{M} is the value $r_{\mathcal{M}}(E)$.

Matroid rank functions and submodularity. It is well known that the rank function of a matroid $\mathcal{M} = (E, \mathcal{I})$ is equivalent to a

submodular function on a set of $|E|$ items with binary marginals [19]. That is, any submodular function on a set of m items with binary marginals corresponds to a matroid with a ground set of m elements, one corresponding to each item, referred as the *underlying matroid* hence forth. The submodular function's value for any subset of the items is equal to the value of the rank function of the underlying matroid for the set of elements corresponding to the items.

Matroid Union. The union function applied to a collection of matroids generates a matroid known as the *union matroid*. Let $\mathcal{M}^{\cup k} = (E^{\cup k}, \mathcal{I}^{\cup k})$ denote the union of a collection of matroids $\mathcal{M}_1 = (E_1, \mathcal{I}_1), \dots, \mathcal{M}_k = (E_k, \mathcal{I}_k)$. $\mathcal{M}^{\cup k}$ is defined as,

$$E^{\cup k} = \bigcup_{i \in [k]} E_i, \quad \mathcal{I}^{\cup k} = \{ \bigcup_{i \in [k]} I_i \mid I_i \in \mathcal{I}_i \}.$$

Essentially, for the union matroid, the ground set is the union of the ground sets of the matroids in the collection, and the independent sets are all possible sets formed by taking the union of one independent set from each underlying matroid.

Rank function of a Union matroid. The rank function of a union matroid $\mathcal{M}^{\cup k}$, denoted by $r_{\mathcal{M}^{\cup k}}(\cdot)$ or simply $r_{\mathcal{M}}(\cdot)$ when the underlying matroids are clear, has the following well known formula.

$$r_{\mathcal{M}}(S) = \min_{T \subseteq S} \left[|S \setminus T| + \sum_{i \in [k]} r_{\mathcal{M}_i}(T \cap E_i) \right] \quad (3)$$

Here for each $i \in [k]$, $r_{\mathcal{M}_i}(\cdot)$ is the rank function of the underlying matroid \mathcal{M}_i .

Union of copies of a matroid. Given a matroid $\mathcal{M} = (E, \mathcal{I})$, let \mathcal{M}^n denote the union of n copies of \mathcal{M} . Let $r(\cdot)$, $r(\mathcal{M})$ and $r^n(\mathcal{M}^n)$ respectively denote the rank function of \mathcal{M} , the rank of \mathcal{M} , and the rank of \mathcal{M}^n . The following properties relating these quantities are well known [19].

Lemma 2.1. $r^n(\mathcal{M}^n) = n \cdot r(\mathcal{M})$ if and only if for all subsets $T \subseteq E$,

$$|E \setminus T| \geq n \cdot [r(E) - r(T)]. \quad (4)$$

Lemma 2.2. If $r^n(\mathcal{M}^n) = n \cdot r(\mathcal{M})$, then \mathcal{M} has at least n disjoint bases.

3 SUBMODULAR VALUATIONS WITH BINARY MARGINALS (MATROID RANK FUNCTIONS)

In this section, we will prove Theorem 3.1. As a corollary, we get the computational result of Theorem 3.2.

Theorem 3.1. If the valuation function v_i of agent i is a submodular function with binary marginals, a.k.a. matroid rank function, then their APS and MMS values are equal, i.e., $\text{MMS}_i = \text{APS}_i$.

Proof Idea. Recall that such a v_i is a *matroid rank function*. To prove this theorem, we consider the underlying matroid of the valuation function v_i . We use the set based definition of APS_i , Definition 2.1, to show that equation (4) is true for \mathcal{M} . This is the most crucial and technically involved step in the proof. With this equation, it then follows from Lemma 2.1 that the ranks of \mathcal{M} and the union matroid of n copies of \mathcal{M} , say respectively $r(\mathcal{M})$ and $r^n(\mathcal{M}^n)$, satisfy $r^n(\mathcal{M}^n) = n \cdot r(\mathcal{M})$. Lemma 2.2 then implies that \mathcal{M} has at least n disjoint bases. These bases translate to bundles of the goods

in the fair allocation instance such that the value of i for each base is at least equal to $r(\mathcal{M})$. From the definition of MMS, MMS_i is thus at least $r(\mathcal{M})$. Finally we show $r(\mathcal{M})$ is equal to APS_i . Combining with Claim 2.1 proves the Theorem.

In the remaining section we discuss the proof in detail, using the above notations.

A key notion towards establishing equation (4) is *capping the valuation function* of i . Using $v_i(\cdot)$, we define a new function $\widehat{v}_i(\cdot)$ as,

$$\widehat{v}_i(S) = \min\{v_i(S), \text{APS}_i\} \quad (5)$$

We first claim that capping v_i maintains the matroid rank property.

Lemma 3.1. If v_i is an MRF then \widehat{v}_i as defined in Equation 5 is also an MRF.

PROOF. We will show that $\widehat{v}_i(\cdot)$ is a submodular function with binary marginals, hence equivalently is an MRF.

Consider any good $g \in [m]$ and set $S \subseteq [m]$. We have,

$$\begin{aligned} \widehat{v}_i(g|S) &= \widehat{v}_i(g \cup S) - \widehat{v}_i(S) \\ &= \min\{v_i(g \cup S), \text{APS}_i\} - \min\{v_i(S), \text{APS}_i\}. \end{aligned}$$

Now if $\min\{v_i(S), \text{APS}_i\} = \text{APS}_i$, then by the monotonicity of $v_i(\cdot)$, $\min\{v_i(g \cup S), \text{APS}_i\} = \text{APS}_i$, implying their difference is zero, and $\widehat{v}_i(g|S) = 0$.

Otherwise, if $v_i(S) < \text{APS}_i$, then $v_i(g \cup S)$ is at most 1 more than $v_i(S)$, as $v_i(\cdot)$ has binary marginals. Therefore, $\min\{v_i(g \cup S), \text{APS}_i\}$ is also at most 1 more than $v_i(S) = \min\{v_i(S), \text{APS}_i\}$, their difference is at most one, hence $\widehat{v}_i(g|S) \leq 1$. This shows that $\widehat{v}_i(\cdot)$ has binary marginals.

It is left to show that $\widehat{v}_i(\cdot)$ is submodular. We consider any set S' that is a superset of S , and show $\widehat{v}_i(g|S) \geq \widehat{v}_i(g|S')$.

If $g \in S$, then both of these values are zero. Also when $\widehat{v}_i(g|S) = 1$, then as $\widehat{v}_i(\cdot)$ has binary marginals, the inequality follows easily. Finally, suppose $g \notin S$, and $\widehat{v}_i(g|S) = 0$. Similarly as $\widehat{v}_i(g|S)$, we have,

$$\widehat{v}_i(g|S') = \min\{v_i(g \cup S'), \text{APS}_i\} - \min\{v_i(S'), \text{APS}_i\}$$

Here if $v_i(S) \geq \text{APS}_i$, then from the monotonicity of v_i , all the terms $v_i(S' \cup \{g\}), v_i(S'), v_i(S \cup \{g\}) \geq \text{APS}_i$, and both the marginal utilities $\widehat{v}_i(g|S)$ and $\widehat{v}_i(g|S')$ are zero.

Otherwise, when $v_i(S) < \text{APS}_i$, then as $\widehat{v}_i(g|S) = 0$, we have $\min\{v_i(g \cup S), \text{APS}_i\} - v_i(S) = 0$. Again as $v_i(S) < \text{APS}_i$, $\min\{v_i(g \cup S), \text{APS}_i\} = v_i(g \cup S)$. Therefore, $\widehat{v}_i(g|S) = v_i(g|S)$, and $v_i(g|S)$ is also zero. As $v_i(\cdot)$ is submodular, $v_i(g|S) \geq v_i(g|S')$. As $v_i(\cdot)$ has binary marginals, $v_i(g|S') = 0$. We have,

$$\begin{aligned} v_i(g|S') &= v_i(g \cup S') - v_i(S') \\ &\geq \min\{v_i(g \cup S'), \text{APS}_i\} - \min\{v_i(S'), \text{APS}_i\} \\ &= \widehat{v}_i(g|S'). \end{aligned}$$

Thus $\widehat{v}_i(g|S') = 0$, and therefore $\widehat{v}_i(g|S') = \widehat{v}_i(g|S)$. \square

Next, we relate the APS values of i under \widehat{v}_i to their corresponding values under v_i . Let APS value of i under \widehat{v}_i be $\overline{\text{APS}}_i$.

Claim 3.1. $\overline{\text{APS}}_i = \text{APS}_i$.

PROOF. First note that since \widehat{v}_i is capped at APS_i , the value of any set, in particular the best bundle she can afford at any price vector, cannot be greater than APS_i . Thus, from the price based definition of APS shown in equation (1), $\overline{\text{APS}}_i \leq \text{APS}_i$.

To see the other direction, consider Definition 2.1 of APS. Since the APS value of the function v_i is APS_i , there exist some k sets $\mathcal{S} = \{S_1, \dots, S_k\}$ each of value at least APS_i under v_i , and corresponding weights $\Lambda = \{\lambda_1, \dots, \lambda_k\}$ that satisfy the constraints in Definition 2.1. Now, even under \widehat{v}_i , these sets have value at least APS_i , in fact, exactly APS_i . We show that the sets \mathcal{S} and their weights Λ form a feasible solution to the program of Definition 2.1 for $z = \text{APS}_i$, even under \widehat{v}_i .

The only constraints that depend on the valuation function are, $\lambda_T = 0$ for all T where $\widehat{v}_i(T) \leq z$. As we have fixed $z = \text{APS}_i$, and as $\widehat{v}_i(T) \leq v_i(T)$ for any set T , these constraints hold. The remaining constraints hold trivially. Therefore, $\overline{\text{APS}}_i \geq z = \text{APS}_i$. \square

Analogously, we relate the MMS values of i under \widehat{v}_i to their corresponding values under v_i . Let MMS value of i under \widehat{v}_i be $\overline{\text{MMS}}_i$.

Claim 3.2. $\text{MMS}_i \geq \overline{\text{MMS}}_i$.

PROOF. Consider any MMS defining allocation A^π under \widehat{v}_i . The minimum valued bundle in this allocation has value $\overline{\text{MMS}}_i$ according to \widehat{v}_i . By definition of \widehat{v}_i , $v_i(S) \geq \widehat{v}_i(S)$ for every set $S \subseteq [m]$. Thus, the same allocation has a value at least $\overline{\text{MMS}}_i$ even under v_i . From the definition of MMS, MMS_i is at least equal to the minimum bundle's value under v_i from the allocation A^π , hence at least $\overline{\text{MMS}}_i$. \square

Let $\mathcal{M}_{\widehat{v}}$ denote the underlying matroid of the function \widehat{v}_i , and let $\mathcal{M}_{\widehat{v}}^n$ be the union matroid of n copies of $\mathcal{M}_{\widehat{v}}$.

Lemma 3.2. $r(\mathcal{M}_{\widehat{v}}) = \text{APS}_i$.

PROOF. As \widehat{v}_i caps the valuations at APS_i , no independent set can have size more than APS_i , hence the rank of $\mathcal{M}_{\widehat{v}}$ is at most APS_i . At the same time, from Claim 3.1, $\overline{\text{APS}}_i = \text{APS}_i$. The set based Definition 2.1 of APS then implies that there exist some sets of value at least APS_i under \widehat{v}_i . This implies that the rank $r_{\mathcal{M}_{\widehat{v}}}([m]) \geq \text{APS}_i$. Together, we get $r_{\mathcal{M}_{\widehat{v}}}([m]) = \text{APS}_i$. \square

We now prove the key lemma towards proving Theorem 3.1.

Lemma 3.3. For any subset $T \subseteq [m]$,

$$|[m] \setminus T| \geq n \cdot [\text{APS}_i - r_{\mathcal{M}_{\widehat{v}}}(T)] \quad (6)$$

PROOF. From Claim 3.1, the APS value under $\widehat{v}_i(\cdot)$ is APS_i . Definition 2.1 of APS shows that there exists an optimal feasible solution to the program. Let the sets and their associated weights in the solution be $\mathcal{S} = \{S_1, \dots, S_k\}$ and $\Lambda = \{\lambda_1, \dots, \lambda_k\}$ respectively. We have $\widehat{v}_i(S_j) \geq \text{APS}_i$ for all $j \in [k]$, and the total weight on any particular good g , that is, $\sum_{j: g \in S_j} \lambda_j$, is at most $\frac{1}{n}$. As $r_{\mathcal{M}_{\widehat{v}}}(S_j) = \text{APS}_i$, as $\widehat{v}_i(S_j) = \text{APS}_i$ for all $j \in [k]$. That is, the sets S_j are bases of the matroid $\mathcal{M}_{\widehat{v}}$.

Consider any set $T \subseteq [m]$. Let the rank of T be t_r . As $r(\mathcal{M}_{\widehat{v}}) = \text{APS}_i$ from Lemma 3.2, $t_r \leq \text{APS}_i$. Therefore, using Property 2 of

Matroid definition 2.3, we can move $\text{APS}_i - t_r$ elements from each S_j to T . Let S'_j be any set of $\text{APS}_i - t_r$ elements that can be added to T , with a marginal utility of one for each element.

The total weight of all the distinct elements in the sets S'_j can be expressed in two ways as,

$$\sum_{g \in \cup_j S'_j} \sum_{j: g \in S'_j} w(g, S'_j) = \sum_{j \in [k]} \sum_{g \in S'_j} w(S'_j),$$

where $w(S'_j)$ is the weight of the set S'_j , and $w(g, S'_j)$ is the weight on good g due to it belonging in S'_j , in the solution (\mathcal{S}, Λ) to the program defining APS.

As the total weight on each good is at most $1/n$, the left expression can be evaluated as,

$$\sum_{g \in \cup_j S'_j} \sum_{j: g \in S'_j} w(g, S'_j) \leq \sum_{g \in \cup_j S'_j} \frac{1}{n} = |\cup_{g \in [k]} S'_j| \cdot \frac{1}{n}.$$

As $w(S'_j)$ is equal to the weight of the set S_j , the right expression can be evaluated as,

$$\sum_{j \in [k]} \sum_{g \in S'_j} w(S'_j) = \sum_{j \in [k]} |S'_j| \lambda_j.$$

Equating the two, we get,

$$\sum_{j \in [r]} \lambda_j |S'_j| \leq \frac{1}{n} \cdot |\cup_{j \in [r]} S'_j|.$$

As $|S'_j| = \text{APS}_i - t_r$, and $\sum_{j \in [r]} \lambda_j = 1$,

$$|\cup_{j \in [r]} S'_j| \geq n \cdot (\text{APS}_i - t_r) \quad (7)$$

Finally, as all the sets S'_j $j \in [k]$, add elements to T that are not already present in T , we have that $\cup_{j \in [k]} S'_j \subseteq [m] \setminus T$. Thus, $|\cup_{j \in [k]} S'_j| \geq |[m] \setminus T|$. Substituting in equation (7),

$$|[m] \setminus T| \geq n \cdot (\text{APS}_i - t_r) = n \cdot [\text{APS}_i - r_{\mathcal{M}_{\widehat{v}}}(T)]. \quad \square$$

Theorem 3.1 follows as a combination of all the lemmas.

PROOF OF THEOREM 3.1. First, by substituting Lemma 3.2 in equation (3.3), and combining with Lemma 2.1 we immediately get the following relation.

$$r(\mathcal{M}_{\widehat{v}}^n) = n \cdot r(\mathcal{M}_{\widehat{v}}). \quad (8)$$

Combining this with Lemma 2.2, we get that $\mathcal{M}_{\widehat{v}}$ has at least n disjoint bases. This means we can create a partition of $[m]$ where each part has value $r(\mathcal{M}_{\widehat{v}})$, which is equal to APS_i from Lemma 3.2. Thus, $\overline{\text{MMS}}_i \geq \text{APS}_i$. Along with Lemma 3.2 we then have, $\text{MMS}_i \geq \text{APS}_i$. Finally, we know from Claim 2.1 $\text{APS}_i \geq \text{MMS}_i$. Therefore $\text{APS}_i = \text{MMS}_i$. \square

Finally, we prove as a corollary of Theorem 3.1, the following computational result.

Theorem 3.2. Given a fair allocation instance $([n], [m], (v_i)_{i \in [n]})$ where every agent's valuation function is an MRF, there is an efficient (polynomial time) algorithm to compute allocation that simultaneously guarantees every agent a bundle of value at least APS_i and has maximum social welfare among all allocations.

PROOF. We know from [8] that an allocation that gives every agent a bundle of value at least MMS_i and maximizes social welfare exists and can be computed efficiently. Theorem 3.1 implies the same allocation guarantees an APS or higher valued bundle to each agent. \square

4 XOS VALUATIONS WITH BINARY MARGINALS

In this section, we consider APS approximations when agents have XOS valuations with binary marginals. Theorems 4.1, 4.2 and 4.3 are the main results of this section.

Theorem 4.1. *If the valuation function of an agent i in an instance $([n], [m], (v_i)_{i \in [n]})$ is an XOS function with binary marginals, then their APS and MMS values satisfy $APS_i \leq 2 \cdot MMS_i + 1$.*

Proof Idea. Recall the notions of partial and wasteful allocations from Section 2. The crucial step in the proof is Algorithm 1, which takes as input a fair allocation instance $([n], [m], (v_i)_{i \in [n]})$, and yields a non-wasteful, partial allocation where each of the allocated bundles has value at most $MMS_i + 1$, and the set of unallocated goods has a value of at most MMS_i . Using this allocation, we fix prices on the goods such that for agent i the highest value of any affordable bundle of goods, that is one with total price at most $1/n$, is at most $2MMS_i + 1$. The price based definition shown in equation (1) of APS, then implies the theorem.

Let us now discuss the details of the proof. Hence forth, we call an allocation *balanced* for an agent, if the difference in the values of the smallest and largest bundles according to the agent's valuation function is at most 1.

Algorithm 1: Non-wasteful balanced MMS allocation

Input : $([n], [m], v_i(\cdot))$ where v_i is a binary XOS valuation function

Output: A non-wasteful, balanced MMS allocation according to v_i , where the leftover goods also have value at most $MMS_i + 1$.

- 1 Initialize $\mathcal{A} = (A_1, A_2, \dots, A_n)$ to be any MMS-defining allocation for agent i
- 2 Initialize $\mathcal{R} \leftarrow \emptyset$
- 3 **for** $j \in [n]$ **do**
- 4 **if** $v_i(A_j) \geq MMS_i + 1$ **then**
- 5 Let $G \in \{A \subseteq A_j | v_i(A) = |A| = MMS_i\}$
- 6 Set $A_j \leftarrow G$
- 7 $\mathcal{R} \leftarrow \mathcal{R} \cup \{A_j \setminus G\}$
- 8 $\mathcal{R} \leftarrow [m] \setminus \bigcup_{j \in [n]} A_j$
- 9 **while** $v_i(\mathcal{R}) \geq MMS_i + 1$ **do**
- 10 Let $A' \subseteq \mathcal{R}$ with $v_i(A') = |A'| = MMS_i + 1$
- 11 Let $k \leftarrow \arg \min_{j \in [n]} v_i(A_j)$
- 12 $\mathcal{R} \leftarrow \mathcal{R} \cup A_k$
- 13 $\mathcal{A}_k \leftarrow A'$
- 14 **return** \mathcal{A}

Algorithm. The algorithm starts by computing any MMS-defining allocation \mathcal{A} for agent i , and performs two phases. First, while any bundle A_j 's value in \mathcal{A} is more than $MMS_i + 1$, it considers any subset G of A_j that has both size and value exactly MMS_i , leaves G with agent j and removes the remaining goods. All goods removed in this way are added to a bundle called \mathcal{R} . In the second phase, while the value of \mathcal{R} is higher than $MMS_i + 1$, it considers any subset A' of \mathcal{R} of both size and value exactly $MMS_i + 1$. The algorithm takes away the bundle of the smallest valued agent, adds this to \mathcal{R} , and gives A' to this agent instead.

We use two results that will be useful in establishing that this algorithm converges in the special kind of allocation desired. Claim 4.1 is a technical property of XOS valuations, followed by Lemma 4.1 which shows a key property of Algorithm 1.

Claim 4.1. *Given an XOS valuation function $v : 2^{[m]} \rightarrow \mathbb{R}_{\geq 0}$ with binary marginals, and any set $S \subseteq [m]$, we can find a subset $S' \subseteq S$ such that $v(S') = v(S) = |S'|$.*

PROOF. Since v is an XOS function, there is a family \mathcal{F} of additive functions such that for all $S \subseteq [m]$, there is an additive function $f \in \mathcal{F}$ with $v(S) = f(S)$. Given the set S consider any such function f . As f is additive, $f(S) = \sum_{g \in S} f(g)$. Further, as v has binary marginals, $f(g) \in \{0, 1\}$ for all g . We define the set $S' := \{g \in S | f(g) = 1\}$. Thus we have, $v(S) = f(S) = f(S') = |S'|$.

Finally, as $f(S') = |S'|$, $v(S') \geq |S'|$, but as v has binary marginals, $v(S') \leq |S'|$, thus $v(S') = |S'|$. \square

Remark 4.1. *We note here that while Claim 4.1 seems obvious, it is not true for binary subadditive valuations which are the immediate generalisation of binary XOS valuations. To see this, consider a function on $m = 3$ goods where the entire set of goods is valued at 2 and any strict subset of the three goods has a value 1. One can verify that this function is subadditive and that Claim 4.1 does not hold for this function.*

We next prove the following lemma.

Lemma 4.1. *Algorithm 1 terminates, and the output allocation \mathcal{A} is a non-wasteful, balanced, MMS allocation according to v_i . Furthermore, $v_i(\mathcal{R}) \leq MMS_i$.*

PROOF. Consider the first For loop (Steps 3 to 7). This loop accesses every bundle A_j at most once, and if its value is more than $MMS_i + 1$, finds a subset $G \subseteq A_j$ of value MMS_i . From Claim 4.1, such a set G always exists. Since the algorithm is not required to execute in polynomial time, let us assume that the algorithm finds the set by enumeration. Thus, at the end of this loop, we have $v_i(A_j) = MMS_i$ for each $j \in [n]$.

Let us now look at the While loop (Steps 9 to 13). If this loop has n or more iterations, then after n iterations it replaces more than n bundles, and finds a new (partial) allocation where every bundle has value equal to $MMS_i + 1$. From the definition of MMS, this means $MMS_i > MMS_i + 1$, a contradiction. Therefore, this loop is executed at most $n - 1$ times and in the end, we have $v_i(A_j) \leq MMS_i + 1$ for all j . After the loop terminates, the condition for staying in the loop is falsified, hence $v_i(\mathcal{R}) \leq MMS_i$.

Therefore, in at most $O(n)$ iterations of the For and While loops, the algorithm terminates, and yields an allocation where $\text{MMS}_i \leq v_i(A_j) \leq \text{MMS}_i + 1$ for all $j \in [n]$, and $v_i(\mathcal{R}) \leq \text{MMS}_i$, that is, a non-wasteful, balanced and MMS allocation. \square

PROOF OF THEOREM 4.1. Consider the partial allocation \mathcal{A} and the set of remaining goods \mathcal{R} obtained at the end of Algorithm 1. We define a price vector $\mathbf{p} = (p_j)_{j \in [m]}$ for the goods based on \mathcal{A} as follows.

$$p_j = \begin{cases} \frac{1}{n \cdot |A_k|}, & \forall j \in A_k, \forall A_k \in \mathcal{A} \\ 0, & \forall j \in \mathcal{R}. \end{cases}$$

Let us see the maximum value that an agent with a budget $1/n$ can afford with this price vector. First, they can get all of \mathcal{R} for free. From Lemma 4.1, \mathcal{R} has value at most MMS_i for i . Further, each bundle A_k has value under v_i at most $\text{MMS}_i + 1$. Therefore, each good in \mathcal{A} has price at least $1/(n \cdot (\text{MMS}_i + 1))$. Even if agent i picks all the lowest priced goods and gets a marginal increment of one for each of them, at a budget of $\frac{1}{n}$, they can receive a value of at most $\text{MMS}_i + 1$ from \mathcal{A} . By subadditivity of v_i , their total value from \mathcal{A} and \mathcal{R} together is at most $2 \cdot \text{MMS}_i + 1$ value. From the price based definition of APS, $\text{APS}_i \leq 2 \cdot \text{MMS}_i + 1$. \square

Next, we prove that Theorem 4.1 is almost the tightest relation between MMS and APS values for this setting.

Theorem 4.2. *There exists a fair allocation instance with three agents and six goods, where all the agents have an identical XOS valuation function with binary marginals, and their APS and MMS values satisfy $\text{APS} \geq 2 \cdot \text{MMS}$.*

PROOF. The instance $([n], [m], (v_i)_{i \in [n]})$ is as follows. There are three agents and six goods, i.e. $n = 3$ and $m = 6$. Let the goods be denoted by g_i , $i \in [6]$. The identical XOS valuation function of all the agents has two additive functions in the family \mathcal{F} , say f_1 and f_2 .

The first three goods have value 1 under f_1 and the remaining have value 0, that is, $f_1(g_i) = 1$, for $i \in [3]$, and $f_1(g_i) = 0$ for $i \in [6] \setminus [3]$. Under f_2 , the opposite is the case, i.e., $f_2(g_i) = 0$, for $i \in [3]$, and $f_2(g_i) = 1$ for $i \in [6] \setminus [3]$.

As the agents are identical, they have the same APS and MMS values. Now, as there are three agents, if MMS was more than one, each agent must get at least two goods. Note that under v , an agent can get a value of two if they receive two goods from $\{g_1, g_2, g_3\}$ or two goods from $\{g_4, g_5, g_6\}$. But to create three bundles of size two each, at least one bundle would have one good from each set $\{g_1, g_2, g_3\}$ and $\{g_4, g_5, g_6\}$. This bundle however would have a value of 1 under v . Thus, the MMS cannot be more than 1.

On the other hand, consider the set of 6 sets $\{g_i, g_{i+1}\}$ for $i \in [3]$ and $i \in \{4, 5, 6\}$. Assign a weight of $1/6$ to each of these sets. The total weight assigned is 1. Also, each good belongs in exactly two sets, hence the total weight on any single good is $1/3$. Each set has value 2. Therefore, this is a feasible solution to the Linear program in the APS definition 2.1 for $z = 2$. Thus, $\text{APS} \geq 2$. Therefore, in this instance, $\text{APS} \geq 2 \cdot \text{MMS}$. \square

Theorems 4.1 and 4.2, along with the known computational results for MMS, yield the following results for APS.

Theorem 4.3. *Given a fair allocation instance $([n], [m], (v_i)_{i \in [n]})$ where every agent has a binary XOS valuation function,*

- (1) *A 0.1222-APS allocation, meaning one that gives every agent a bundle of value at least 0.1222 times their APS, can be computed in polynomial time.*
- (2) *Even when agents have identical valuations, no better than 0.5-APS allocation may exist.*

To prove this theorem, we first prove the following lemmas to separate the agents with $\text{APS} = 0$ and $\text{MMS} = 0$.

Lemma 4.2. *Given an instance $([n], [m], (v_i)_{i \in [n]})$ with XOS binary marginal valuations, one can check in polynomial time if any agent has $\text{MMS} = 0$.*

PROOF. To check if $\text{MMS}_i = 0$ for some agent i , we form a complete weighted bipartite graph where one side has n vertices, and the goods correspond to vertices on the other side. The weight of each edge is the value of the agent i for the good adjacent to the edge. We compute one maximum weight matching of the goods. If each of the n vertices in the left part gets assigned a good of value 1, then one can form an allocation with the matched goods in separate bundles. The remaining goods can be allocated arbitrarily. By subadditivity, each bundle has value at least 1, hence $\text{MMS}_i \geq 0$. \square

Lemma 4.3. *For an agent with a binary XOS valuation function in a fair allocation instance, $\text{APS} = 0$ if and only if $\text{MMS} = 0$.*

PROOF. If $\text{MMS}_i > 0$, then from Claim 2.1, $\text{APS}_i > 0$. Otherwise when $\text{MMS}_i = 0$, the allocation returned by Algorithm 1 has the following properties according to Lemma 4.1. The value of $\mathcal{R} = 0$, and the value of each bundle in \mathcal{A} is either 0 or 1. Further, there are at most $(n - 1)$ bundles with value 1, else $\text{MMS}_i \geq 1$. Also, each of the 1 valued bundles have exactly 1 good in them, and the 0 valued bundles have no goods in them. We assign prices to the goods as follows. Assign a price of $1/(n - 1)$ to each good in the highest valued $n - 1$ bundles of \mathcal{A} , assign a price of 0 to all the remaining goods (in \mathcal{R}).

At a budget of $1/n$, the agent cannot afford any 1 valued bundle. As the remaining goods together have a value 0, $\text{APS} = 0$. \square

PROOF OF THEOREM 4.3. From Lemmas 4.2 and 4.3, we remove all the agents with $\text{APS} = 0$ by giving them no goods in polynomial time. For the remaining agents, we know $\text{APS} \geq \text{MMS} \geq 1$. Combined with $\text{APS} \leq 2 \cdot \text{MMS} + 1$ from Theorem 4.1, we have $\text{APS} \leq 3 \cdot \text{MMS}$ for all the remaining agents.

[17] show that there exists an efficient algorithm to compute a 0.3666-MMS allocation, that is, an allocation where every agent receives a bundle of value at least 0.3666 times their MMS value. As $\text{APS} \leq 3 \cdot \text{MMS}$, this implies that an allocation that gives every agent a bundle of value at least 0.1222 times their APS can be computed in polynomial time.

Finally, Theorem 4.2 shows an instance where the agents have identical valuation functions, and their APS is at least twice their MMS. By definition of MMS, no allocation can have the smallest

bundle's value more than MMS. Therefore, in every allocation, at least the smallest bundle's agent receives a bundle of value at most half their APS, and a better than 0.5-APS allocation does not exist. \square

5 SUBMODULAR VALUATIONS WITH TERNARY MARGINALS

In this section, we show the following hardness result.

Theorem 5.1. *In an instance $([n], [m], (v_i)_{i \in [n]})$, it is NP-hard to compute the MMS value of an agent approximately up to a factor better than $5/6$, when the agent has a submodular valuation function, even when all the marginal utilities are in $\{0, 1/2, 1\}$.*

An immediate corollary, using Claim 2.1, is that computing the APS value approximately up to a factor better than $5/6$ is also NP-hard for such an agent. The proof of the theorem has three parts. First, we show a reduction from the known NP-complete problem 3-D-MATCHING to a fair allocation instance with agents with identical valuations. We then show that this valuation function in the reduced instance is submodular with all the marginal values in $\{0, 1/2, 1\}$. Finally, we show the correctness of the reduction, establishing the factor of the hardness of approximation.

Reduction. The 3-D-MATCHING problem is as follows. Given are three disjoint sets X, Y, Z , having m elements each, and a set \mathcal{T} of triples (a, b, c) , where $a \in X, b \in Y$ and $c \in Z$. The problem is to answer if there is a subset of m triples in \mathcal{T} called a 3-D-MATCHING of X, Y, Z , that cover all of X, Y and Z , meaning for all $s \in X \cup Y \cup Z$, $s \in 3\text{-D-MATCHING}$.

Given an instance of 3-D-MATCHING, we form a fair allocation instance as follows. There are m agents, and a set \mathcal{M} of $3m$ goods, one good corresponding to each element of X, Y and Z . All the agents have the following identical valuation function v for the goods. For any subset \mathcal{S} of \mathcal{M} , $v(\mathcal{S})$ is defined as follows.

$$v(\mathcal{S}) = \begin{cases} 1, & \text{if } |\mathcal{S}| = 1 \\ 2, & \text{if } |\mathcal{S}| = 2 \\ 2.5, & \text{if } |\mathcal{S}| = 3 \text{ and } \mathcal{S} \notin \mathcal{T} \\ 3, & \text{if } |\mathcal{S}| = 3 \text{ and } \mathcal{S} \in \mathcal{T} \\ 3, & \text{if } |\mathcal{S}| \geq 4 \end{cases}$$

Function v is submodular with ternary marginals. First, let us compute the marginal utility of a good g over sets of different sizes. From the definition of $v(\mathcal{S})$, one can verify that,

$$v(g|\mathcal{S}) = \begin{cases} 1, & \text{if } |\mathcal{S}| \leq 1 \\ 0.5, & \text{if } |\mathcal{S}| = 2 \text{ and } \mathcal{S} \cup \{g\} \notin \mathcal{T} \\ 1, & \text{if } |\mathcal{S}| = 2 \text{ and } \mathcal{S} \cup \{g\} \in \mathcal{T} \\ 0.5, & \text{if } |\mathcal{S}| = 3 \text{ and } \mathcal{S} \notin \mathcal{T} \\ 0, & \text{if } |\mathcal{S}| = 3 \text{ and } \mathcal{S} \in \mathcal{T} \\ 0, & \text{if } |\mathcal{S}| \geq 4. \end{cases}$$

Therefore, $v(g|\mathcal{S}) \in \{0, 1/2, 1\}$ for all g and \mathcal{S} .

Also $v(g|\mathcal{S}) \leq v(g|\mathcal{S}')$, for any two sets $\mathcal{S}, \mathcal{S}'$ with $|\mathcal{S}| \geq |\mathcal{S}'|$, hence also when $\mathcal{S}' \subseteq \mathcal{S}$. This establishes submodularity.

Correctness. Finally, we prove that the MMS value of any agent is 3 if and only if the 3-D-MATCHING instance has a solution, and is at most 2.5 otherwise.

Suppose a 3-D-MATCHING exists. Then one can divide the 3 goods from each triple in the solution to every agent. Each agent receives a bundle of value 3. As the highest value under v of any set of goods is 3, $\text{MMS} \leq 3$. Therefore, in this case, $\text{MMS} = 3$.

Alternatively, suppose the MMS value of the reduced instance is 3. Then every agent must receive exactly 3 goods. Otherwise, some agent will receive at most 2 goods, and have value at most 2. A bundle of 3 goods has value 3 only when the corresponding elements form a triple in the 3-D-MATCHING instance. Also, the bundles of goods in the MMS-defining allocation are disjoint, hence the triples allocated to all the agents are disjoint. Therefore, the allocation consists of goods corresponding to m disjoint triples, that cover all the elements, hence form a solution of 3-D-MATCHING. Now, if there was an algorithm that computed the MMS value within a factor better than $5/6$ for such instances with submodular functions and ternary marginals, then given the reduced instance, the algorithm would output a value higher than 2.5 if and only if a 3-D-MATCHING existed. This proves Theorem 5.1.

6 CONCLUSION

We analyzed the fairness notion of APS for indivisible goods under submodular and XOS functions with binary marginals, a rich and expressive class of valuation functions [6, 7, 10, 16]. Under binary submodular valuations, we give a rather surprising result that APS = MMS. This is not true for even additive valuations (with non-binary marginals.) On the other hand for fractionally subadditive functions with binary marginals, we show a gap of 2 between APS and MMS and show that is almost tight.

It would be interesting to study the relations between APS and MMS for other valuation functions, and in particular for binary subadditive valuations. Subadditive functions generalizes both fractionally subadditive and submodular functions, and remains relatively less explored. We note that, most of Section 4 can be extended to work for binary subadditive functions, except Claim 4.1. A generalization of this claim for subadditive valuations will be helpful in determining the gap between MMS and APS under these valuations.

Finally, a study of APS for the case when agents are heterogeneous, i.e. each agent has a weight or endowment, under valuations beyond-additive with binary marginals is the next natural question.

ACKNOWLEDGMENTS

Ruta Mehta is supported by NSF CAREER Award CCF 1750436 and NSF grant CCF 2334461.

REFERENCES

[1] Georgios Amanatidis, Georgios Birmpas, Aris Filos-Ratsikas, and Alexandros A Voudouris. 2022. Fair division of indivisible goods: A survey. *arXiv preprint arXiv:2202.07551* (2022).

[2] Haris Aziz, Bo Li, Herve Moulin, and Xiaowei Wu. 2022. Algorithmic fair allocation of indivisible items: A survey and new questions. *arXiv preprint arXiv:2202.08713* (2022).

[3] Moshe Babaioff, Tomer Ezra, and Uriel Feige. 2021. Fair and truthful mechanisms for dichotomous valuations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 5119–5126.

[4] Moshe Babaioff, Tomer Ezra, and Uriel Feige. 2021. Fair-Share Allocations for Agents with Arbitrary Entitlements. In *Proceedings of the 22nd ACM Conference on Economics and Computation*. 127–127.

[5] Siddharth Barman, Umang Bhaskar, Anand Krishna, and Ranjani G Sundaram. 2020. Tight Approximation Algorithms for p-Mean Welfare Under Subadditive Valuations. *arXiv preprint arXiv:2005.07370* (2020).

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

[7] Siddharth Barman and Paritosh Verma. 2021. Approximating Nash social welfare under binary XOS and binary subadditive valuations. In *International Conference on Web and Internet Economics*. Springer, 373–390.

[8] Siddharth Barman and Paritosh Verma. 2021. Existence and Computation of Maximin Fair Allocations Under Matroid-Rank Valuations. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*. 169–177.

[9] Gilad Ben Uziahu and Uriel Feige. 2023. On Fair Allocation of Indivisible Goods to Submodular Agents. *arXiv e-prints* (2023), arXiv–2303.

[10] Nawal Benabbou, Mithun Chakraborty, Ayumi Igarashi, and Yair Zick. 2021. Finding fair and efficient allocations for matroid rank valuations. *ACM Transactions on Economics and Computation* 9, 4 (2021), 1–41.

[11] Mithun Chakraborty, Erel Segal-Halevi, and Warut Suksompong. 2022. Weighted fairness notions for indivisible items revisited. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 4949–4956.

[12] Uriel Feige and Yehonatan Tahan. 2022. On allocations that give intersecting groups their fair share. *arXiv preprint arXiv:2204.06820* (2022).

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

[14] Mohammad Ghodsi, Mohammadtaghi Hajiaghayi, Masoud Seddighin, Saeed Seddighin, and Hadi Yami. 2018. Fair Allocation of Indivisible Goods: Improvements and Generalizations. In *Proceedings of the 2018 ACM Conference on Economics and Computation* (Ithaca, NY, USA) (EC '18). <https://doi.org/10.1145/3219166.3219238>

[15] Bo Li, Yingkai Li, and Xiaowei Wu. 2021. Almost proportional allocations for indivisible chores. *arXiv preprint arXiv:2103.11849* (2021).

[16] Zhentao Li and Adrian Vetta. 2018. The Fair Division of Hereditary Set Systems. In *International Conference on Web and Internet Economics*. Springer, 297–311.

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

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

[19] Alexander Schrijver et al. 2003. *Combinatorial optimization: polyhedra and efficiency*. Vol. 24. Springer.

[20] Akiyoshi Shioura. 2012. Matroid rank functions and discrete concavity. *Japan journal of industrial and applied mathematics* 29, 3 (2012), 535–546.

[21] Hugo Steinhaus. 1948. The problem of fair division. *Econometrica* 16 (1948), 101–104.

[22] Vignesh Viswanathan and Yair Zick. 2022. Yankee Swap: a Fast and Simple Fair Allocation Mechanism for Matroid Rank Valuations. *arXiv preprint arXiv:2206.08495* (2022).