$^1\!/\!_2$-Approximate MMS Allocation for Separable Piecewise Linear Concave Valuations

Anonymous submission

Abstract

We study fair distribution of a collection of m indivisible goods among a group of n agents, using the widely recognized fairness principles of Maximin Share (MMS) and Any Price Share (APS). These principles have undergone thorough investigation within the context of additive valuations. We explore these notions for valuations that extend beyond additivity.

First, we study approximate MMS under the separable (piecewise-linear) concave (SPLC) valuations, an important class generalizing additive, where the best known factor was $1/3$-MMS. We show that $^1\!/\!_2$-MMS allocation exists and can be computed in polynomial time, significantly improving the state-of-the-art. We note that SPLC valuations introduce an elevated level of intricacy in contrast to additive. For instance, the MMS value of an agent can be as high as her value for the entire set of items.[1] We use a relax-and-round paradigm that goes through competitive equilibrium and LP relaxation. Our result extends to give (symmetric) $^1\!/\!_2$-APS, a stronger guarantee than MMS.

APS is a stronger notion that generalizes MMS by allowing agents with arbitrary entitlements. We study the approximation of APS under submodular valuation functions. We design and analyze a simple greedy algorithm using concave extensions of submodular functions. We prove that the algorithm gives a $^1\!/\!_3$-APS allocation which matches the best-known factor by (Uziahu and Feige 2023). Concave extensions are hard to compute in polynomial time and are, therefore, generally not used in approximation algorithms. Our approach shows a way to utilize it within analysis (while bypassing its computation), and hence might be of independent interest.

1 Introduction

We consider the problem of fairly allocating a set $\mathcal{M}$ of m indivisible goods among a set $\mathcal{N}$ of n agents with heterogeneous preferences under the popular fairness notions of *Maximin share* (MMS) (Budish 2011) and *Any Price Share* (APS) (Babaioff, Ezra, and Feige 2021). These notions have been extensively studied for the setting where the agents have additive valuations (Barman and Krishna Murthy 2017; Ghodsi et al. 2018; Garg, McGlaughlin, and Taki 2018; Garg and Taki 2020; Akrami, Garg, and Taki 2023; Akrami and Garg 2023). This paper studies the problem beyond additive valuations, particularly for the classical separable-concave valuations (Vazirani and Yannakakis 2011; Chaudhury et al. 2022) and submodular valuations (Ghodsi et al. 2018; Barman and Krishnamurthy 2020; Uziahu and Feige 2023).

MMS and APS are *share based* fairness notions, where each agent is entitled to a bundle worth her *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, she partitions so that the value of the minimum valued bundle is maximized. Let $\Pi_{\mathcal{N}}(\mathcal{M})$ denote the set of all allocations of $\mathcal{M}$ among the n agents. If $(A_1,\ldots,A_n)$ denotes any allocation into n bundles, and $v_i : 2^{\mathcal{M}} \to \mathbb{R}_+$ denotes agent i's valuation function, then the MMS value of agent i is defined as,

$$\mathsf{MMS}_i = \max_{(A_1,\ldots,A_n)\in\Pi_{\mathcal{N}}(\mathcal{M})} \min_{j\in[n]} v_i(A_j)$$

An MMS allocation is one where every agent i gets a bundle worth at least MMS_i. MMS treats all agents equally. In some settings it is necessary to consider *weighted* agents, where the weight or entitlement of agent i is $b_i > 0$; the weights are normalized to satisfy $\sum_i b_i = 1$. It is not straight forward to define a weighted generalization of MMS. To address this, (Babaioff, Ezra, and Feige 2021) introduced the notion of APS. This fair share value of agent i is defined as the value she can ensure herself with a budget of b_i when the *prices* of the items are chosen adversarially, subject to a normalization constraint that the total sum of prices is 1. More formally, let $\mathcal{P} = \{(p_1\ldots,p_m) \mid \sum_{j=1}^m p_j = 1,\ p_j \geq 0\,\forall j\}$ denote the simplex of price vectors for the m goods. If the budget of agent i is b_i, then,

$$\mathsf{APS}_i = \min_{(p_1,\ldots,p_m)\in\mathcal{P}} \max_{S\subseteq\mathcal{M}:\sum_{j\in S} p_j \leq b_i} v_i(S)$$

We note that when $b_i = \frac{1}{n}$, $\mathsf{APS}_i \geq \mathsf{MMS}_i$ (Babaioff, Ezra, and Feige 2021). Thus, allocations that give guarantees with respect to APS_i at $b_i = \frac{1}{n}$ automatically provide same guarantees for MMS_i.

Allocations achieving MMS and APS shares may not exist even under additive valuations (Procaccia and Wang

[1] Also the equilibrium computation problem, which is polynomial-time for additive valuations, becomes intractable for SPLC (Vazirani and Yannakakis 2011).

2014; Feige, Sapir, and Tauber 2021). Therefore, the focus has been on finding approximate solutions, where in an α-MMS (APS) allocation, every agent receives a bundle worth at least α times their MMS (APS) value. This problem has been studied extensively for additive valuations (see (Amanatidis et al. 2022) for a survey and pointers) with much progress (Procaccia and Wang 2014; Garg, McGlaughlin, and Taki 2018; Kulkarni, Mehta, and Taki 2021). It is known that a $(\frac{3}{4} + \frac{3}{3836})$-$\mathsf{MMS}$ always exists and can be computed in polynomial time (Akrami and Garg 2023), while there are examples showing that $\frac{39}{40}$-MMS may not exist (Feige, Sapir, and Tauber 2021) even in the setting of three agents.

Additive valuations are inapplicable if agents have *decreasing marginal gains*, a crucial property in practice. This raises the need to go beyond additive. 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 (Barman and Verma 2020; Li and Vetta 2021; Viswanathan and Zick 2022). Here we consider separable (piecewise-linear) concave (SPLC) (Vazirani and Yannakakis 2011; Garg et al. 2012a; Magnanti and Stratila 2004; Chaudhury et al. 2022) and submodular valuations. SPLC valuations generalize additive valuations, and form a subclass of submodular valuations. Such functions are separable across different types of goods, and concave within each type capturing decreasing marginal gains. Submodular functions allow decreasing marginal gain across all goods. Formally, a real-valued function $f : 2^{\mathcal{M}} \to \mathbb{R}$ is submodular iff $f(A \cup \{e\}) - f(A) \geq f(B \cup \{e\}) - f(B)$ for all $A \subset B$ and $e \notin B$. For submodular valuations, (Ghodsi et al. 2018) gave an algorithm to find a $\frac{1}{3}$-MMS allocation based on a certain local search procedure, and (Barman and Krishna Murthy 2017) 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 (Uziahu and Feige 2023), who also gave $\frac{1}{3}$-APS algorithm. These results also apply to separable-concave functions and remain the best known.

In terms of lower bounds, for submodular valuations, even for special cases like assignment valuations and weighted matroid rank valuations it is known that better than $\frac{2}{3}$-MMS allocations may not exist (Barman and Krishnamurthy 2020; Kulkarni, Kulkarni, and Mehta 2023). For a more general class, namely fractionally subadditive valuations, it is known that better than $\frac{1}{2}$-MMS allocations may not exist (Ghodsi et al. 2018). Closing the gaps for these rich classes of valuations is of much interest. A natural question here is whether $\frac{1}{2}$-MMS allocations exist for submodular valuations and it is open even for SPLC valuations.

One of the difficulties in going beyond additive valuations is the lack of good upper bound on the MMS (APS) values of the agents. For example, if v_i is additive and $b_i = \frac{1}{n}$ then $\mathsf{MMS}_i \leq \mathsf{APS}_i \leq \frac{v_i(\mathcal{M})}{n}$, while if v_i is separable-concave or submodular then we can have $\mathsf{MMS}_i = v_i(\mathcal{M})$ (see Example A.1 in Appendix A).

1.1 Our Results

In this paper, we develop novel ways to upper bound the MMS and APS values for monotone valuation functions via *market equilibrium* and *concave extensions*. This is our conceptual contribution. We leverage this to obtain two results.

First, we design an LP-relaxation-based method to find a $1/2$-MMS allocation for SPLC valuations in polynomial time. Our result is in fact stronger; the algorithm outputs an allocation that gives each agent a value at least $1/2$-APS_i (for symmetric agents). SPLC valuations are a special case of submodular valuations, and for the latter, the best known result for MMS allocations is a very recent result (Uziahu and Feige 2023) that yields a $\frac{10}{27}$-MMS allocation. Thus, for SPLC valuations we obtain an improved approximation.

Second, we show that a simple greedy algorithm achieves $\frac{1}{3}$-APS allocation for submodular valuations. This result was independently obtained in a recent work by (Uziahu and Feige 2023). Their proof technique is conceptually different and uses the bidding game methodology of (Babaioff, Ezra, and Feige 2021). We make use of an argument via the concave extension approach. We believe that this may be of independent interest and may offer helpful insights.

In Appendix A, we also point out an example (Example A.2) to show that a greedy-like approach cannot yield a $1/2$-MMS allocation for SPLC valuations. This points to the importance of the LP-based approach that we utilize.

2 Preliminaries

We use $[k]$ to denote the set $\{1, 2, \cdots, k-1, k\}$.

2.1 Fair division model

We study the problem of fairly dividing a set of m indivisible goods $\mathcal{M} = [m]$ among a set of n agents $\mathcal{N} = [n]$ who can have asymmetric entitlements. The entitlement or weight of agent i is denoted by b_i, and the weights are normalized so that their sum is 1, that is, $\sum_{i \in \mathcal{N}} b_i = 1$. The preferences of an agent $i \in \mathcal{N}$ are defined by a valuation function $v_i : 2^{\mathcal{M}} \to \mathbb{R}_{\geq 0}$ over the set of goods. We represent a fair division problem instance by $(\mathcal{N}, \mathcal{M}, (v_i)_{i \in \mathcal{N}}, (b_i)_{i \in \mathcal{N}})$. When all agents have the same weight, we denote the instance by $(\mathcal{N}, \mathcal{M}, (v_i)_{i \in \mathcal{N}})$ and call it the symmetric fair division instance. An allocation $A := (A_1, \ldots, A_n)$ 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 $\cup_{i \in [n]} A_i = [m]$. Note that empty parts are allowed. We denote the set of all allocations by $\Pi_{[n]}([m])$. We will also sometimes use fractional allocations of goods. We denote these allocations by $\mathbf{x}$ and the allocation of a particular agent, $i \in \mathcal{N}$ by $\mathbf{x}_i$. Formally, $\mathbf{x} = (x_{ij})_{i \in \mathcal{N}, j \in \mathcal{M}}$ such that $\sum_{i \in \mathcal{N}} x_{ij} \leq 1$ for all $j \in \mathcal{M}$ and $\mathbf{x}_i = (x_{ij})_{j \in \mathcal{M}}$ where $(x_{ij})_{i \in \mathcal{N}, j \in \mathcal{M}}$ is a fractional allocation.

Throughout this paper, we assume that the valuation functions are monotone and non-negative. Further, Section 4 assumes that valuations functions are SPLC and Section 5 assumes that the valuation functions are submodular. We define these classes of functions next.

SPLC **valuations.** Separable-concave, a.k.a. separable piecewise-linear concave (SPLC), valuations is a well-studied class that subsumes additive valuations. Under SPLC valuations, we have t types of goods, with each good $j \in [t]$ having k_j copies. For an SPLC valuation function $f(\cdot)$, a value f_{jk} is associated with k^{th} copy of good j. The functions are concave so that for all $j \in [t]$, we have $f_{j1} \geq f_{j2} \geq \ldots \geq f_{jk_j}$. Finally, the valuations are additive across different goods. Formally, we let $\mathcal{M}$ be the set of all goods (that includes all copies of each type of good). For all $j \in [t]$, we denote by $\mathcal{M}_j \subseteq \mathcal{M}$ the subset of goods that are copies of good j. The value for any set $S \subseteq \mathcal{M}$ is given by $f(S) = \sum_{j\in[t]} \sum_{k\leq|S\cap\mathcal{M}_j|} f_{jk}$.

Throughout this paper, when we refer to fair division problems in the specific context of SPLC valuations, we denote the instance by $(\mathcal{N}, t, (k_j)_{j\in[t]}, (v_i)_{i\in\mathcal{N}})$.

Submodular Valuations. Submodular valuations are a popular class of valuations in the complement-free hierarchy. The valuations are characterized by the property of decreasing marginal utility. In particular, a valuation function $f(\cdot)$ is said to be submodular if and only if, for all goods $g \in \mathcal{M}$ and any subsets $S \subset Q \subseteq \mathcal{M}$, $f(g \mid S) \geq f(g \mid Q)$, where $f(g \mid S)$ denotes the marginal utility of good g on set S, i.e. $f(g \mid S) := f(g \cup S) - f(S)$.

2.2 Fairness Notions

Maximin Share (MMS) Maximin Share (MMS) is defined for symmetric agents, that is for all agents $i \in [n]$, $b_i = 1/n$. Consider a symmetric fair division instance $(\mathcal{N}, \mathcal{M}, (v_i)_{i\in\mathcal{N}})$. The MMS value of an agent $i \in \mathcal{N}$ is defined as,

$$\mathsf{MMS}_i^n([m]) := \max_{(A_1,\ldots,A_n)\in\Pi_{[n]}([m])} \min_{k\in[n]} v_i(A_k) \quad (1)$$

We refer to $\mathsf{MMS}_i^n([m])$ by MMS_i when the qualifiers n and m are clear from the context. The following claim about MMS, called single good reduction is well known and we will use it in our analysis.

Claim 2.1. *Given a fair division instance $(\mathcal{N}, \mathcal{M}, (v_i)_{i\in\mathcal{N}})$, the* MMS *value of an agent is retained if we remove any single agent and any single good. That is,* $\mathsf{MMS}_i^n([m]) \geq \mathsf{MMS}_i^{n-1}([m \setminus \{g\}])$ *for all* $g \in [m]$

Any Price Share (APS) Let $\mathcal{P}$ denote the simplex of price vectors over the set of goods $\mathcal{M} = [m]$, formally, $\mathcal{P} = \{(p_1, \ldots, p_m) \geq 0 \mid \sum_i p_i = 1\}$. For an instance $(\mathcal{N}, \mathcal{M}, (v_i)_{i\in\mathcal{N}}, (b_i)_{i\in\mathcal{N}})$, the APS value of agent i is defined as,

$$\mathsf{APS}_i^n([m]) := \min_{p\in\mathcal{P}} \max_{S\subseteq[m], p(S)\leq b_i} v_i(S) \quad (2)$$

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

An alternate definition without prices is as follows.

Definition 2.1 (Any Price Share)**.** *The* APS *value of an agent i for an instance $(\mathcal{N}, \mathcal{M}, (v_i)_{i\in\mathcal{N}}, (b_i)_{i\in\mathcal{N}})$ is the solution of the following program.*

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

These definitions and their equivalence is stated in (Babaioff, Ezra, and Feige 2021). We also note the following claim that was proved in (Babaioff, Ezra, and Feige 2021)

Claim 2.2. *For any symmetric fair division instance,* $(\mathcal{N}, \mathcal{M}, (v_i)_{i\in\mathcal{N}})$, $\mathsf{APS}_i^n([m]) \geq \mathsf{MMS}_i^n([m])$ *for all agents* $i \in [n]$.

Similar to Claim 2.1, single good reduction also holds for APS when agents are symmetric. We give a proof of the following claim in Appendix B.

Claim 2.3. *Given a symmetric fair division instance* $(\mathcal{N}, \mathcal{M}, (v_i)_{i\in\mathcal{N}})$, *the* APS *value of an agent is retained if we remove any single agent and any single good. That is,* $\mathsf{APS}_i^n([m]) \geq \mathsf{APS}_i^{n-1}([m \setminus \{g\}])$ *for all* $g \in [m]$

α-Approximate Allocations. Given a symmetric fair division instance, $(\mathcal{N}, \mathcal{M}, (v_i)_{i\in\mathcal{N}})$, we say that an allocation $A = (A_1, \ldots, A_n)$ is α-MMS if for all $i \in [n]$, $v_i(A_i) \geq \alpha\mathsf{MMS}_i$. Similarly, for an instance $(\mathcal{N}, \mathcal{M}, (v_i)_{i\in\mathcal{N}}, (b_i)_{i\in\mathcal{N}})$, we say an allocation is α-APS if for all agents $i \in \mathcal{N}$, $v_i(A_i) \geq \alpha\mathsf{APS}_i$. The notion of α-MMS or α-APS allocations also applies to fractional allocations.

2.3 Concave Extension

The valuation functions of agents are discrete set functions. In Section 3 we discuss new upper bounds on MMS and APS values of agents. These upper bounds require us to consider fractional allocations. Therefore, we must extend the input valuation function to include values for fractional sets. We assume $f : 2^V \to \mathbb{R}_+$ is a non-negative monotone real-valued set function over a finite ground set V. A natural way to extend f to a continuous concave function over the hypercube $[0,1]^V$ is called the concave closure or concave extension, and is denoted by f^+. It is defined as follows.

Definition 2.2 (Concave extension of f)**.**

$$f^+(x) = \max_{(\alpha_S)_{S\subseteq 2^{\mathcal{M}}}} \{\sum_{S\subseteq V} f(S)\alpha_S \mid \sum_S \alpha_S = 1 \text{ and } \sum_{S\ni i} \alpha_S = x_i \quad \forall i \in V \text{ and } \alpha_S \geq 0 \quad \forall S \subseteq V\}. \quad (3)$$

Remark 2.1. *For submodular valuations, concave extensions* NP*-hard to evaluate. Therefore, another extension* *called the Multilinear Extension is more widely used in Fair Division literature with these functions. For those familiar with this theory, we show in Appendix B that under Multilinear Extension with submodular valuations, there are instances where no fractional* 1-MMS *allocation exists. This is the main reason we work with concave extension here.*

2.4 Market (Competitive) Equilibrium

The theory of market equilibrium typically considers fractional allocations of goods. Therefore, in this part we will assume that agents have continuous valuation function. Consider an instance $(\mathcal{N}, \mathcal{M}, \widehat{v}_i(\cdot), (b_i)_{i\in\mathcal{N}})$ where, for $m = |\mathcal{M}|$, $\widehat{v}_i : \mathbb{R}^m \to \mathbb{R}_+$ is a non-negative non-decreasing continuous valuation function of agent i. A market equilibrium constitutes prices $\mathbf{p} = (p_j)_{j\in\mathcal{M}}$ of goods and an allocation $\mathbf{x} = (x_{ij})_{i\in\mathcal{N}, j\in\mathcal{M}}$, where x_{ij} is the amount of good j allocated to agent i, such that the following conditions are satisfied (Arrow and Debreu 1954)

1. Each agent $i \in \mathcal{N}$ is allocated her *optimal bundle*, i.e., $(x_{i1}, \ldots, x_{im}) \in \text{argmax}_{y\in\mathbb{R}^m_+ : \Sigma_{j\in\mathcal{M}} : y_j p_j \le b_i} \widehat{v}_i(y)$.
2. Market clears: all goods $j \in \mathcal{M}$ are completely sold, i.e. $\sum_{i\in\mathcal{N}} x_{ij} = 1$, or if $\sum_{i\in N} x_{ij} < 1$ then $p_j = 0$.

3 Upper Bounds via Market Equilibrium and Concave Extension

In this section we give new upper bounds that we can use to approximate APS and MMS. These upper bounds are on APS value of agents and by Claim 2.2 they also upper bound the MMS value. Section 3.1 finds an upper bound using Market Equilibrium and section 3.2 finds an upper bound using concave extensions of functions. They are based on primal and dual definitions of APS respectively. The proofs of these upper bounds appear in Appendix C.

3.1 Market Equilibrium Based Bound

Consider a fair division instance, $(\mathcal{N}, \mathcal{M}, (v_i)_{i\in\mathcal{N}}, (b_i)_{i\in\mathcal{N}})$. Let $\widehat{v}_i(\cdot)$ be any continuous extension of $v_i(\cdot)$ *under which market equilibrium exists*. Then we can prove the following lemma.

Lemma 3.1. *For any fair division instance, $(\mathcal{N}, \mathcal{M}, (v_i)_{i\in\mathcal{N}}, (b_i)_{i\in\mathcal{N}})$, let $(\mathbf{x}^*, \mathbf{p}^*)$ denote a market equilibrium with continuous extension $\widehat{v}_i(\cdot)$. Then, for all $i \in \mathcal{N}$, $\mathsf{APS}_i \le \widehat{v}_i(\mathbf{x}^*_i)$.*

Remark 3.1. *For any monotone, non-negative set function, $v(\cdot)$, its concave extension $v^+(\cdot)$ will be non-decreasing, non-negative, continuous, and concave. As long as agents together are non-satiated for the available supply of goods,* i.e., *have non-zero marginal up to consuming all the goods, a market (competitive) equilibrium is known to exist (Arrow and Debreu 1954). This will give us an upper bound to work with.*

3.2 Concave Extension Based Bound

For a given set function $f : 2^V \to \mathbb{R}_+$ and a real value $\gamma \ge 0$ we define the truncation of f to γ, denoted by $f_{\downarrow\gamma}$, as follows: $f_{\downarrow\gamma}(A) = \min\{f(A), \gamma\}$ for each $A \subseteq V$. It is well-known and easy to verify that truncation of a monotone submodular function yields another monotone submodular function. We make a connection between APS value and the concave closure via the following lemma.

Lemma 3.2. *Consider a fair division instance, $(\mathcal{N}, \mathcal{M}, (v_i)_{i\in\mathcal{N}}, (b_i)_{i\in\mathcal{N}})$. For all agents $i \in \mathcal{N}$, $\mathsf{APS}_i = \sup\{z : v^+_{i\downarrow z}(b_i, b_i, \ldots, b_i) = z\}$.*

Remark 3.2. *Note that the truncation is important in the indivisible setting, for otherwise the relaxation is too weak. We also see that APS_i can be computed if one can evaluate the concave closure of the truncation of v_i*

4 1/2-MMS for SPLC valuations

In this section, we give a polynomial time algorithm for computing an allocation that gives each agent a bundle they value at least half as much as their MMS value. As mentioned in the Introduction, all our results in this section hold for symmetric APS also. The section is organized as follows. In Section 4.1 we give a linear relaxation for SPLC valuations and show that under this relaxation, there exists a fractional allocation that gives each agent their MMS value. In Section 4.2 we give a linear program and show that if the program is feasible, we can, in polynomial time find an integral solution where each agent loses at most one good. Then in Section 4.3 we use the results in Sections 4.1 and 4.2 to prove existence of $\frac{1}{2}$-MMS allocation under SPLC valuations. Finally in Section 4.4, we address the computational aspects and give a polynomial time algorithm to compute the $\frac{1}{2}$-MMS allocation. All missing proofs of this Section are in Appendix D.

4.1 Fractional 1-MMS allocation exists

Recall that under SPLC valuations, we have t goods and for each good, $j \in [t]$ there are k_j copies. Every agent $i \in [n]$ has a value v_{ijk} associated with k^{th} copy of j^{th} good. For an SPLC valuation $v_i(\cdot)$, we denote its linear extension by $v_i^L(\cdot)$ and define it as follows.

Definition 4.1 (Linear Extension for SPLC function)**.** *Let variables x_{ijk} denote the fraction of k^{th} copy of good j that agent i receives. We use $\mathbf{x}_i = (x_{ijk})_{j\in[t],k\in[k_j]}$ to denote the vector of fractional allocation received by agent i. The value of the set is then defined as*

$$v_i^L(\mathbf{x}_i) := \sum_{j\in[t]} \sum_{k\in k_j} v_{ijk} x_{ijk} \tag{4}$$

It is known from (Vazirani and Yannakakis 2011) that a market equilibrium exists under this linear relaxation. Combining this with Lemma 3.1, we immediately get the following lemma.

Lemma 4.1. *Given an* SPLC *fair division instance $(\mathcal{N}, t, (k_j)_{j\in[t]}, (v_i)_{i\in\mathcal{N}})$, there exists a fractional allocation $\mathbf{x} = (x_{ijk})_{i\in[n],j\in[t],k\in[k_j]}$ such that $v_i^L(\mathbf{x}_i) \ge \mathsf{MMS}_i$.*

The above lemma proves existence of 1-MMS fractional allocations under linear relaxation. However, market equilibrium under this relaxation is PPAD-hard to compute (Garg et al. 2012b). In the next section we give a linear program that uses Lemma 4.1 to compute a (fractional) 1-MMS allocation in polynomial time.

4.2 Linear Program to Compute a Fractional 1-MMS allocation

Given an SPLC fair division instance, $(\mathcal{N}, t, (k_j)_{j\in[t]}, (v_i)_{i\in\mathcal{N}})$, consider the following Linear Feasibility Program parameterized by $(\mu_i)_{i\in\mathcal{N}}$.

$$
\begin{aligned}
\textstyle\sum_j \sum_k v_{ijk} x_{ijk} &\geq \mu_i && \text{for all } i \in \mathcal{N} \\
\textstyle\sum_i \sum_k x_{ijk} &\leq k_j && \text{for all } j \in [t] \qquad (5)\\
0 \leq x_{ijk} &\leq 1 && \text{for all } i, j, k
\end{aligned}
$$

We say that $(\mu_i)_{i\in\mathcal{N}}$ are *feasible* if there exists a feasible solution to (5) for the given $(\mu_i)_{i\in\mathcal{N}}$.

The rest of this section is dedicated to proving the following lemma which says that if LP (5) is feasible, we can get an integral allocation where each agent gets a value of at least $\mu_i - \mathsf{Max}_i$ where Max_i is the maximum value that agent i has for any good, i.e. $\mathsf{Max}_i = \max_{j,k} v_{ijk}$.

Lemma 4.2. *Given $(\mu_i)_{i\in\mathcal{N}}$ that are feasible, in polynomial time we can find an integral allocation such that for all $i \in \mathcal{N}$, agent i gets a set A_i with $v_i(A_i) \geq \mu_i - \max_{j,k} v_{ijk}$.*

Remark 4.1. *The rest of this section proves Lemma 4.2. Readers not wanting to go into technical details of this proof can move to Section 4.3 to see how to use it for proving the existence of $\frac{1}{2}$-MMS allocations.*

To prove Lemma 4.2, we need to start with a fractional solution that satisfies some properties enabling us to round it. Towards this, we consider the fractional solution that maximizes the following optimization program.

$$
\begin{aligned}
&\max \textstyle\sum_{i,j,k} v_{ijk} x_{ijk} \\
\textstyle\sum_j \sum_k v_{ijk} x_{ijk} &\geq \mu_i && \text{for all } i \in \mathcal{N} \\
\textstyle\sum_i \sum_k x_{ijk} &\leq k_j && \text{for all } j \in [t] \qquad (6)\\
x_{ijk} &\leq 1 && \text{for all } i, j, k \\
x_{ijk} &\geq 0 && \text{for all } i, j, k
\end{aligned}
$$

Denote the optimal of this program with $\mathbf{x}^* = (x^*_{ijk})_{i\in\mathcal{N}, j\in[t], k\in[k_j]}$. We start with the following simple claim.

Claim 4.1. *$\mathbf{x}^*$ can be modified so that for all $i \in \mathcal{N}, j \in [t]$, x^*_{ijk} is fractional for exactly one k.*

Therefore, for each agent and each good, we have only one copy fractionally allocated[2]. This lets us define the following graph of fractional allocations.

Definition 4.2 (Fractional Allocation Graph)**.** *This is a bipartite graph with agents on one side and one single copy of each good on the other side. We draw edge between agent i and good j if there is a copy of good j that agent i gets fractionally. The weight of this edge is the fractional amount of the good assigned to the agent.*

The optimal solution obtained might be such that the Fractional Allocation Graph has cycles in it. In Appendix D.1, we prove the following Lemma that says even if the graph has cycles, they can be eliminated without reducing the value received by any agent.

Lemma 4.3. *Given any fractional optimal solution to* LP (6) *$\mathbf{x}^*$, we can get a fractional solution, $\bar{\mathbf{x}}$ such that the Fractional Allocation Graph corresponding to $\bar{\mathbf{x}}$ has no cycles and $v_i^L(\mathbf{x}_i^*) = v_i^L(\bar{\mathbf{x}}_i)$ for all $i \in \mathcal{N}$.*

Algorithm 1 describes the rounding algorithm that lets us prove Lemma 4.2.

Algorithm 1: Rounding the Fractional Optimal of LP (6)

Input : $\mathbf{x}^*$ that is an optimal solution to LP (6).
Output: Integral Allocation $A = (A_1, \ldots, A_n)$ where each agent $i \in \mathcal{N}$ receives a value of at least $\mu_i - \max_{j,k} v_{ijk}$.

1 Use Lemma 4.3 to convert $\mathbf{x}^*$ to $\bar{\mathbf{x}}$ such that the Fractional Allocation Graph corresponding to $\bar{\mathbf{x}}$ has no cycles and value of each agent is same.
2 Create Fractional Allocation Graph corresponding to $\bar{\mathbf{x}}$.
3 In the forest obtained, root every tree by an arbitrary agent.
4 For each tree, assign all the children goods to the parent agent.

Proof of Lemma 4.2. Consider the rounding procedure described in Algorithm 1. From Lemma 4.3, we get that the agents have lost no value up to step 2. After that in Step 4, every agent loses at most one good – the parent good in the tree. This proves the lemma's claim on value of agent. To see that the algorithm runs in polynomial time, note that solving a Linear Program can be done in polynomial time using any of the well known algorithms. We detail in Appendix D that cycle cancellation removes a cycle by deleting one edge. Since there are at most $n \cdot t$ edges in the graph, this also runs in polynomial time. Therefore, overall the subroutine runs in polynomial time. □

[2]Note that in the fractional allocation, there might be more than one copy of a good that is fractionally allocated among the agents, however each agent is only allocated one copy fractionally.

4.3 Existence of $\frac{1}{2}$-MMS Allocations

Algorithm 2 gives an algorithm that proves the existence of $\frac{1}{2}$-MMS allocations. It does so by assuming we can compute MMS values for all agents. While these are NP-hard to compute making the algorithm non-polynomial time, it exhibits all other important ideas in getting a polynomial time $\frac{1}{2}$-MMS allocation.

Algorithm 2: Existence of $\frac{1}{2}$-MMS Algorithm for SPLC valuations

---

Input : $(\mathcal{N}, t, (k_j)_{j\in[t]}, (v_i)_{i\in\mathcal{N}})$
Output: Allocation A where for every agent $v_i(A_i) \geq \mathsf{MMS}_i/2$

1 Define $\mu_i := \mathsf{MMS}_i$ for all $i \in \mathcal{N}$.
2 Initialize set of *active* agents $\mathcal{A} = \mathcal{N}$ and set of *active* goods $\mathcal{G} = \mathcal{M}$.
3 **while** *there exists* $i \in \mathcal{N}$ *and* $j \in [t]$ *such that* $v_{ij1} \geq \frac{\mu_i}{2}$ **do**
4 $\quad A_i \leftarrow \{j\}$
5 $\quad \mathcal{A} \leftarrow \mathcal{A} \setminus \{i\}$
6 $\quad$ Remove one copy of j from $\mathcal{G}$
7 Solve the Linear Program (6) for the reduced instance $(\mathcal{A}, \mathcal{G}, (v_i)_{i\in\mathcal{N}})$ and obtain fractional allocation $\mathbf{x}$.
8 Use Lemma 4.2 to get an integral allocation and assign appropriate bundles to agents in $\mathcal{A}$.
9 Return $A = (A_i)_{i\in\mathcal{N}}$.

Theorem 4.1. *Given an* SPLC *fair division instance,* $(\mathcal{N}, t, (k_j)_{j\in[t]}, (v_i)_{i\in\mathcal{N}})$*, Algorithm 4 gives each agent a bundle* A_i *such that* $v_i(A_i) \geq \frac{1}{2}\mathsf{MMS}_i$.

Proof. Note that for any agent who gets allocated an item in Steps 3 to 6, they get a value of $\frac{\mathsf{MMS}_i}{2}$ by definition of $\mu_i = \mathsf{MMS}_i$. Recall from Claim 2.1 that removal of one good and one agent retains the MMS value of the agent. Therefore, for all remaining agents, at the end of while loop from Step 3 to 6, the MMS value of active agents is retained in the reduced instance. Further the reduced instance satisfies the property that for each agent $i \in \mathcal{N}$, $\max_{j,k} v_{ijk} \leq \frac{\mathsf{MMS}_i}{2}$. Finally, using Lemma 4.1, note that $\mu_i := \mathsf{MMS}_i$ for all $i \in \mathcal{A}$ (the set of Active agents left after single good reduction), form *feasible* $(\mu_i)_{i\in\mathcal{A}}$. Therefore, we can use Lemma 4.2 to get an integral allocation where each agent, $i \in \mathcal{A}$ gets a value of $v_i(A_i) \geq \mathsf{MMS}_i - \max_{j,k} v_{ijk} \geq \frac{\mathsf{MMS}_i}{2}$. This completes the proof. $\square$

4.4 Algorithm : Computational Aspects

We now address the final computational aspect of finding MMS_i values here. Since they are NP-hard to compute, we use μ_i values as follows:

$$\mu_i = \sum_{j\in[t]} \left(\sum_{k \leq \frac{k_j}{n}} v_{ijk} + v_{ij\lceil k_j/n \rceil} \left(\frac{k_j}{n} - \left\lfloor \frac{k_j}{n} \right\rfloor \right) \right) \quad (7)$$

In Appendix, section D.2, we give the complete algorithm that shows how to use these μ_i to get a $\frac{1}{2}$-MMS allocation. We only state the main theorem here.

Theorem 4.2. *There exists an algorithm which in polynomial time outputs an allocation that gives* $\frac{1}{2}$-MMS *to each agent.*

Remark 4.2. *The values μ_i defined in Equation 7 are upper bounds on* APS *also. From Claim 2.3, single good reduction also holds for* APS *with symmetric agents. These two things together give us that all our results of this section also go through for* APS *approximation for symmetric agents.*

5 Approximate APS allocations with Submodular valuations

We now move to the more general case of submodular valuations and agents with non-symmetric entitlements. In this section, given a fair division instance, $(\mathcal{N}, \mathcal{M}, (v_i)_{i\in\mathcal{N}}, (b_i)_{i\in\mathcal{N}})$, we show that a greedy algorithm gives us an allocation $A = (A_1, \ldots, A_n)$ such that $v_i(A_i) \geq \frac{1}{3}\mathsf{APS}_i$ for all $i \in \mathcal{N}$. While this algorithm is simpler than the Algorithm for SPLC valuations seen in previous section, we give an example in Appendix A (example A.2) to show that a natural modification fails to give a $\frac{1}{2}$ approximation for even SPLC valuations. The section is organized as follows. In Section 5.1, we outline some basic properties for APS. Section 5.2 describes the details of the greedy algorithm assuming we can compute the APS values for agents. Section 5.3 analyses the algorithm and shows how to bypass the computational aspect of computing APS values.

5.1 Properties of APS

We will use the following properties of APS in our analysis. The proofs are simple and we omit them here.

Claim 5.1. *For any* $z \leq APS_i$, $v^{+}_{i\downarrow z}(b_i, b_i, \ldots, b_i) = z$. *Moreover, if* $\alpha_S, S \subseteq [m]$ *is a feasible solution to the LP defining the value* $v^{+}_{i\downarrow z}(b_i, b_i, \ldots, b_i)$ *then* $\alpha_S > 0$ *implies that* $v_{i\downarrow z}(S) = z$.

Claim 5.2. *[Scale-freeness] If a valuation function* $v : 2^{\mathcal{M}} \to \mathbb{R}_{\geq 0}$ *is multiplied by any scalar value* α*, the* APS *value of an agent with the scaled function is* α *times the* APS *value of an agent with the original function and same entitlement.*

Claim 5.3. *[Capping retains* APS*] If an agent with some entitlement* b *and valuation function* $v : 2^{\mathcal{M}} \to \mathbb{R}_{\geq 0}$ *has* APS *value* γ*, then the* APS *value of an agent with the same entitlement and the capped function* $v'(\mathcal{S}) := \min\{\mathsf{APS}, v(\mathcal{S})\}$ $\forall Set \subseteq \mathcal{M}$ *is also* γ.

5.2 Algorithm.

The main subroutine of our algorithm is Algorithm 3, which takes as input a fair division instance $(\mathcal{N}, \mathcal{M}, (v_i)_{i\in\mathcal{N}}, (b_i)_{i\in\mathcal{N}})$ along with a vector of numbers $(\beta_i)_{i\in\mathcal{N}}$. It outputs an allocation with the following property: each agent i receives a bundle of value at least $\frac{1}{3}\beta_i$ if

$\beta_i \le \mathsf{APS}_i$; note that the guarantee holds for agent i even if the β values for other agents are larger than their APS values. In particular, if we have the APS values of all agents, the algorithm is efficient and gives each agent $\frac{1}{3}$ fraction of her APS. Although computing the APS value of an agent is NP-hard, we show in Appendix E that we can bypass this issue and use Algorithm 3 to derive a polynomial time algorithm for a $\frac{1}{3}$-approximate APS allocation.

Algorithm 3 works as follows. Given an instance of the fair division problem, and guesses for the APS values of all agents (β_i for agent i), it performs two pre-processing steps. The intuition is the following. Suppose we knew the exact APS_i values for each i. Since our algorithm is greedy, we must normalise valuations. APS values are scale invariant (Claim 5.2) therefore, we scale v_i such that $\mathsf{APS}_i = nb_i$ implying $\sum_i \mathsf{APS}_i = n$. More over, if we knew APS_i then truncating v_i to APS_i is convenient and does not affect the value (Claim 5.3). In our algorithm we do have β_i values rather than APS_i values. Nevertheless we will proceed as if these values are correct estimates. After scaling and truncating we have for each agent i a valuation function $\hat{v}_i$. We have the property that $\hat{v}_i$ is truncated at nb_i.

The key part of the algorithm is the following greedy strategy. The algorithm allocates goods in multiple rounds. Each round greedily chooses an agent and a good that maximizes the objective $\min\{2nb_i/3, v_i(j \mid A_i)\}$ over all agents and goods. Recall that $v_i(j \mid A_i)$ is the marginal value of the (unallocated) good j to agent i's current bundle A_i. The selected good is allocated to the selected agent. As soon as an agent receives a bundle of value at least $nb_i/3$, it is satisfied and removed from consideration in future rounds.

It is easy to see that the algorithm will terminate in at most m rounds. We will show that at termination the following property is true: for each agent i such that $\beta_i \le \mathsf{APS}_i$, $v_i(A_i) \ge \beta_i/3$ where A_i is the allocation to i. This will be used to obtain a polynomial time algorithm for a $\frac{1}{3}$-APS allocation (see Appendix E).

5.3 Analysis

The following is the main theorem of our section.

Theorem 5.1. *If $\beta_i \le \mathsf{APS}_i$ for an agent $i \in \mathcal{A}$, then Algorithm 3 terminates with an allocation such that $v_i(A_i) \ge \frac{1}{3}\beta_i$.*

We prove Theorem 5.1 by contradiction. Fix an agent i and suppose $\beta_i \le \mathsf{APS}_i$ and $\hat{v}_i(A_i) < nb_i/3$. The algorithm removes i from consideration during the algorithm if at time t, $\hat{v}_i(A_i^t) \ge nb_i/3$, thus i must have stayed active until termination and the algorithm allocated all goods to agents. We compute the total sum of marginals that agent i sees for all the goods (allocated to her and other agents). We compute this value in two different ways and obtain a contradiction. The details of proof and of computational aspects of the algorithm are deferred to Appendix E.

Algorithm 3: Greedy Procedure for APS with Submodular Valuations

---

Input : $(\mathcal{N}, \mathcal{M}, (v_i)_{i\in\mathcal{N}}, (b_i)_{i\in\mathcal{N}})$, vector $(\beta_i)_{i\in\mathcal{N}}$
Output: Either (i) allocation $\mathcal{A}$ where for every agent $v_i(A_i) \ge \beta_i/3$, or (ii) some agent $i : \beta_i > \mathsf{APS}_i$

Normalization: $v_i' :=$ normalized v_i so that $\beta_i = nb_i$ that is, for all sets $S \subseteq \mathcal{M}$, $v_i'(S) = v_i(S) \cdot \frac{nb_i}{\beta_i}$
Truncation: $\hat{v}_i := v_{i\downarrow nb_i}$
Initialize $A^\pi \leftarrow (\emptyset)_{i\in\mathcal{N}}, \mathcal{M}^r \leftarrow \mathcal{M}, \mathcal{N}^r \leftarrow \mathcal{N}$ // $\mathcal{N}^r$ is list of active agents and $\mathcal{M}^r$ is unallocated items
while $\mathcal{M}^r \neq \emptyset$ *and* $\mathcal{N}^r \neq \emptyset$ **do**
  Let $S = \{(i,j) \mid i \in \mathcal{N}^r, j \in \mathcal{M}^r\}$ // all remaining agent-good pairs
  $(i^*, j^*) \in \operatorname{argmax}_{(i,j)\in S} \min\{\frac{2}{3}nb_i, \hat{v}_i(j \mid A_i)\}$ // greedily choose the agent with the highest marginal capped at $2nb_i/3$
  $A_{i^*} \leftarrow A_{i^*} \cup \{j^*\}, \mathcal{M}^r \leftarrow \mathcal{M}^r \setminus \{j^*\}$ // allocate the chosen good to the agent
  if $\hat{v}_{i*}(A_{i^*}) \ge nb_{i^*}/3$ **then**
    $\mathcal{N}^r \leftarrow \mathcal{N}^r \setminus \{i^*\}$ // remove agent if they received at least $nb_i/3$
if $\exists i \in [n], v_i(A_i) < \beta_i/3$ **then**
  return one such i
return A^π

References

Ageev, A. A.; and Sviridenko, M. I. 2004. Pipage rounding: A new method of constructing algorithms with proven performance guarantee. *Journal of Combinatorial Optimization*, 8: 307–328.

Akrami, H.; and Garg, J. 2023. Breaking the $3/4$ Barrier for Approximate Maximin Share. arXiv:2307.07304.

Akrami, H.; Garg, J.; and Taki, S. 2023. Improving Approximation Guarantees for Maximin Share. arXiv:2307.12916.

Amanatidis, G.; Aziz, H.; Birmpas, G.; Filos-Ratsikas, A.; Li, B.; Moulin, H.; Voudouris, A. A.; and Wu, X. 2022. Fair division of indivisible goods: A survey. *arXiv preprint arXiv:2208.08782*.

Arrow, K.; and Debreu, G. 1954. Existence of an Equilibrium for a Competitive Economy. *Econometrica*, 22: 265–290.

Babaioff, M.; Ezra, T.; and Feige, U. 2021. Fair-Share Allocations for Agents with Arbitrary Entitlements. In *Proceedings of the 22nd ACM Conference on Economics and Computation*, 127–127.

Barman, S.; and Krishna Murthy, S. K. 2017. Approximation Algorithms for Maximin Fair Division. In *Proceedings of the 2017 ACM Conference on Economics and Computation*, 647–664.

Barman, S.; and Krishnamurthy, S. K. 2020. Approximation algorithms for maximin fair division. *ACM Transactions on Economics and Computation (TEAC)*, 8(1): 1–28.

Barman, S.; and Verma, P. 2020. Existence and computation of maximin fair allocations under matroid-rank valuations. *arXiv preprint arXiv:2012.12710*.

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

Chaudhury, B. R.; Cheung, Y. K.; Garg, J.; Garg, N.; Hoefer, M.; and Mehlhorn, K. 2022. Fair division of indivisible goods for a class of concave valuations. *Journal of Artificial Intelligence Research*, 74: 111–142.

Feige, U.; Sapir, A.; and Tauber, L. 2021. A tight negative example for MMS fair allocations. In *International Conference on Web and Internet Economics*, 355–372. Springer.

Garg, J.; McGlaughlin, P.; and Taki, S. 2018. Approximating Maximin Share Allocations. In *2nd Symposium on Simplicity in Algorithms (SOSA 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

Garg, J.; Mehta, R.; Sohoni, M.; and Vazirani, V. V. 2012a. A complementary pivot algorithm for markets under separable, piecewise-linear concave utilities. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, 1003–1016.

Garg, J.; Mehta, R.; Sohoni, M.; and Vazirani, V. V. 2012b. A complementary pivot algorithm for markets under separable, piecewise-linear concave utilities. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, 1003–1016.

Garg, J.; and Taki, S. 2020. An Improved Approximation Algorithm for Maximin Shares. In *Proceedings of the 21st ACM Conference on Economics and Computation*, EC '20, 379–380. ISBN 9781450379755.

Ghodsi, M.; HajiAghayi, M.; Seddighin, M.; Seddighin, S.; and Yami, H. 2018. Fair allocation of indivisible goods: Improvements and generalizations. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, 539–556.

Kulkarni, P.; Kulkarni, R.; and Mehta, R. 2023. Maximin Share Allocations for Assignment Valuations. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, 2875–2876.

Kulkarni, R.; Mehta, R.; and Taki, S. 2021. Indivisible mixed manna: On the computability of mms+ po allocations. In *Proceedings of the 22nd ACM Conference on Economics and Computation*, 683–684.

Li, Z.; and Vetta, A. 2021. The fair division of hereditary set systems. *ACM Transactions on Economics and Computation (TEAC)*, 9(2): 1–19.

Magnanti, T. L.; and Stratila, D. 2004. Separable concave optimization approximately equals piecewise linear optimization. In *International Conference on Integer Programming and Combinatorial Optimization*, 234–243. Springer.

Procaccia, A. D.; and Wang, J. 2014. Fair enough: Guaranteeing approximate maximin shares. In *Proceedings of the fifteenth ACM conference on Economics and computation*, 675–692. ACM.

Uziahu, G. B.; and Feige, U. 2023. On fair allocation of indivisible goods to submodular agents. *arXiv preprint arXiv:2303.12444*.

Vazirani, V. V.; and Yannakakis, M. 2011. Market equilibrium under separable, piecewise-linear, concave utilities. *Journal of the ACM (JACM)*, 58(3): 1–25.

Viswanathan, V.; and Zick, Y. 2022. Yankee swap: a fast and simple fair allocation mechanism for matroid rank valuations. *arXiv preprint arXiv:2206.08495*.