Fair and Efficient Division of Indivisibles

Rucha Kulkarni

April 14, 2022

1 Introduction

Finding fair allocations is a fundamental problem in algorithmic game theory. With divisible resources, it is commonly phrased as the cake cutting problem and has been extensively studied; see [71] for a summary. Here, a division of a *cake* that gave one piece to each of n agents was termed fair if it ensured properties like (a) envy-freeness, meaning every agent values her own piece more than those allocated to other agents, and (b) proportionality, meaning every agent values her piece at least $1/n$ fraction of her total value for the cake.

When there are two agents, the simple cut and choose protocol is known to work since the biblical era, where one agent cuts the cake into two pieces and the other agent gets to choose first. Recent years have seen a surge of works on the fair division of indivisible items, like school/course seats, assets and liabilities, and computing resources on networks, due to their wide applications [73, 21, 78, 65, 36, 23, 45]. A simple example of allocating a single indivisible item among two agents shows that both envy-free and proportional allocations may not exist here.

This led to the design of measures of fairness for the indivisible item setting. Today, several measures are known and have been extensively studied (for instance, [66] introduced the Nash welfare, and [23] introduced the maximin share notion and some relaxations of envy-freeness). The problem of finding allocations that are fair and additionally *economically efficient* as well was then studied, either using measures that balanced both objectives, like the Nash welfare [See Section 1.1 for a review of existing literature], or by adding efficiency constraints like Pareto optimality (PO) along with known fairness measures. For instance, [17] study EF1, a common relaxation of envy-freeness called *envy-free up to (the removal of) one desirable item (from the envied bundle)* with PO for the additive valuations setting[1], [9, 81, 26, 72] study relaxations of envy-freeness with PO for various special cases of additive valuations, [9] study proportionality with PO. Additionally, [8] discuss EF1 solutions that satisfy the efficiency notions of utilitarian maximality and rank maximality.

The focus of this article is the problem of finding fair and efficient allocations of indivisible items using two objectives: the Nash welfare, and the Maximin share and Pareto optimality. To formally describe these measures, we introduce the following notation. Let $(\mathcal{N}, \mathcal{M}, \mathcal{V})$ denote an instance where $\mathcal{N}$ is a set of n agents, $\mathcal{M}$ is a set of m indivisible items, and $\mathcal{V}$ is the set of n monotonically non-decreasing valuation functions $\mathcal{V} = \{v_i \mid v_i : S \subseteq \mathcal{M} \to \mathbb{R}, i \in \mathcal{N}\}$. If all the items in $\mathcal{M}$ are non-negatively valued by each agent, the items are called goods and $\mathcal{M}$ is called a *goods manna*. If the items are all non-positively valued, $\mathcal{M}$ is called a *chores manna*, and every item is called a

[1] a valuation function is additive if the value of a set of items is the sum of values of the singleton sets containing the individual items

chore. More generally, if every item can be positively valued by some agents and negatively valued by some, $\mathcal{M}$ is called a *mixed manna*. When allocating a chores or a mixed manna, the problem usually requires us to allocate every chore to at least one agent. Finally let us denote by $\mathcal{A}$ and $\mathcal{A}_i$ respectively, an allocation of the items in $\mathcal{M}$ among agents in $\mathcal{N}$, and the set of items assigned to agent i in $\mathcal{A}$. Let us now define our objectives.

Nash welfare. A popular notion proven to strike a balance between fairness and efficiency [24] is the Nash welfare product, defined as,

$$\mathsf{NSW}(\mathcal{A}) = \left(\prod_{\mathsf{i}} \mathsf{v_i}(\mathcal{A}_\mathsf{i})\right)^{1/\mathsf{n}} \tag{1}$$

The *Nash welfare or* NSW *problem* is to find an allocation of maximum NSW value. This is typically studied for a goods manna.

To study the computational tractability of this problem, consider an instance of the Partition problem where given is a set of m elements, and each element j has a positive integer weight w_j. We reduce to an instance of the fair division problem as follows. Let there be two agents, a set $\mathcal{M}$ of m goods, one corresponding to each element of Partition, and an identical valuation function of both agents for the goods defined as $v(S) = \sum_{j \in S} w_j$, for any subset $S \subseteq \mathcal{M}$. It can be proved that the solution of the Partition problem corresponds to an allocation with the maximum NSW product among all allocations.

This reduction shows that the NSW problem is NP-hard, even for the special case of two agents and identical additive valuation functions. The problem is also known to be APX-hard [57]. This led to relaxing the problem to approximate versions, defined as follows.

The α-NSW problem. Given an instance $(\mathcal{N}, \mathcal{M}, \mathcal{V})$ over a goods manna, and a scalar $\alpha > 1$, find an allocation with NSW product at least $1/\alpha$ times the NSW value of the maximal NSW product allocation.

Extensive work has led to a rich theory on this problem [See Section 1.1]. Almost all of this theory studies special cases of the problem like assuming that the valuation functions of all agents are additive(-like). These works fail to capture several real world applications like school seat assignments, spectrum allocation, air traffic management, allocating computing resources on a network, splitting assets and liabilities in partnership dissolution, and office tasks. These problems require one or more of the following generalizations.

Submodular valuation functions. A function is called submodular if it is monotone non-decreasing, and every good has non-increasing marginal utility over larger sets of goods. That is, v is a submodular function over the set of goods M iff $v(\{g\} \cup A) - v(A) \geq v(\{g\} \cup B) - v(B)$ for all goods $g \in M$ and sets $A \subseteq B \subseteq M$.

Asymmetric weights to agents. Suppose every agent i in the fair division problem instance is now associated with a scalar value w_i, representing its weight. Intuitively, a fair division must now distribute the goods in proportion to the agent's weight. The NSW notion can be extended to this setting, by studying the weighted geometric mean of valuations,

$$\mathsf{NSW}(\mathcal{A}) = \left(\prod_{\mathsf{i}} \mathsf{v_i}(\mathcal{A}_\mathsf{i})^{\mathsf{w_i}}\right)^{1/\sum_\mathsf{i} \mathsf{w_i}}. \tag{2}$$

This product is called the *asymmetric* Nash welfare product of an allocation, and first proposed by [48, 50]. The asymmetric NSW problem is therefore to find allocations that (approximately) maximizes the asymmetric NSW value.

The NSW problem with asymmetric agents and submodular valuation functions is far more general than the settings studied prior to our work. While numerous techniques had been introduced to solve several important special cases, all of these exploited the symmetry of agents and the characteristics of additive-like valuation functions. For instance, the notion of a maximum bang-per-buck (MBB) item is critically used in most of these approaches. There is no such equivalent notion for the submodular case. Also, many approaches define novel markets by relaxing the indivisibility condition on the goods, and design novel rounding techniques. In this rounding step, some agents can lose a lot of their value, but it is shown that combined with the other agents, the NSW product does not reduce too much. In the asymmetric case, this may not work, as these few agents may have high weights and reduce the asymmetric NSW product by a lot. As a consequence, no approximation algorithm with a factor independent of the number of items m [67] was known either for the asymmetric agents with additive valuations or the symmetric agents beyond additive(-like) valuations cases. As typically the number of items m is far higher than the number of agents n, the natural question here, also raised in [31, 17], is,

Is there an efficient algorithm to find an α-NSW allocation where the agents are asymmetric, and/or have submodular valuation functions, for any α that is independent of the number of items?

We settle this question in our first work [41]. Our main contribution is two approximation algorithms, SMatch for the asymmetric agents with additive valuations setting, and RepReMatch for asymmetric agents with submodular valuations. We show that these algorithms achieve approximation factors of $O(n)$ for the additive case and $O(n \log n)$ for the submodular case, where n is the number of agents. SMatch outputs an allocation that is also EF1. Both algorithms are simple to understand and involve non-trivial modifications of a greedy repeated matchings approach. Allocations of high valued items are done separately by un-matching certain items and re-matching them, by processes that are different in both algorithms. We discuss the details of SMatch in Section 2.

Table 1 summarizes the approximation guarantees of the algorithms RepReMatch and SMatch under popular valuation functions [69]. All the best known results prior to our work are also stated here for reference.

Valuations	**Symmetric Agents**		**Asymmetric Agents**	
	Hardness	**Algorithm**	**Hardness**	**Algorithm**
Additive	1.069 [39]	1.45 [17]	1.069 [39]	$O(n)$ ✓
Budget-additive	—"—	1.45 [30]	—"—	—"—
SPLC	—"—	—"—	—"—	$O(n \log n)$ ✓
OXS Gross substitutes	—"—	$O(n \log n)$ ✓	—"—	—"—
Submodular	1.5819 ✓	—"—	1.5819 ✓	—"—
XOS Subadditive	—"—	$O(m)$ [68]	—"—	$O(m)$ [68]

Table 1: Summary of results. Every entry has the best known approximation guarantee for the setting. A checkmark in front of the result shows our contribution.

To complement these results, we also provide a 1.5819-factor hardness of approximation result for the submodular NSW problem. This hardness also applies to the case when the number of agents is constant. This shows that the submodular problem is strictly harder than the settings studied so far, for which 1.45 factor approximation algorithms are known.

For the special case of the submodular NSW problem where the number of agents is constant, we describe another algorithm with a *matching* 1.5819 *approximation factor*, hence resolving this case. Finally, a 1.45-factor guarantee can be shown for the further special case of restricted additive valuations, by showing that the allocation returned by the algorithm in this case is PO. This matches the current best known approximation factor for this case.

We now introduce the second objective of focus of this article and summarize our work here.

Maximin share and Pareto optimality. A well-studied notion of fairness is the maximin share (MMS)[23]. The MMS value of an agent is the maximum value they can receive if they were to distribute the items of a mixed manna into n sets and choose the least valued set. That is, when $\mathcal{P}$ is the set of all allocations of $\mathcal{M}$ into n sets,

$$\mathsf{MMS_i} = \max_{\mathcal{A}\in\mathcal{P}} \min_{\mathsf{j}\in[\mathsf{n}]} \mathsf{v_i}(\mathcal{A}_\mathsf{j}) \tag{3}$$

The MMS *problem* is to find an MMS allocation, defined as one where every agent receives a set of items worth at least their MMS value.

The reduction from Partition discussed previously also shows that the MMS problem is NP-hard, even with two agents and identical valuation functions. Additionally, for the case of additive valuation functions that are non-identical, [70] showed that an MMS allocation may not even exist. Hence, the following approximation problem is studied.

The α-MMS problem. Given an instance $(\mathcal{N}, \mathcal{M}, \mathcal{V})$ and an $\alpha > 0$, find an α-MMS allocation, defined as one where every agent i receives a set of items that they value at least α times their MMS value.

This problem has been extensively studied for the goods or chores manna settings. We initiate the study of this problem for two generalizations.

Mixed manna. As discussed previously, a mixed manna includes both goods that are freely disposable and chores that have to be assigned.

Economic efficiency with α-MMS: Consider the following example. Suppose there are two agents denoted as $1, 2$, and two items, say u, v. Agent 1 values item u at 1000, and v at 0, while agent 2 has the same but swapped values, that is value 0 for item u and 1000 for v. The MMS value of each agent is 0 in this case, and the allocation assigning the lesser valued item to each agent is a 1-MMS allocation. Clearly a more desirable allocation for both agents would give each their more liked good. This instance shows that while the MMS objective guarantees fairness, it does not ensure economic efficiency.

Towards resolving this, we add the long established notion of economic efficiency, Pareto optimality (PO). An allocation is called PO if there is no other allocation where every agent receives an equal or higher valued set of items, and at least one agent gets a higher valued set. That is, $\mathcal{A}$ is called PO if there is no other allocation $\mathcal{A}'$ such that $v_i(\mathcal{A}'_i) \geq v_i(\mathcal{A}_i)$ for all $i \in [n]$, and there exists some $i \in [n]$ with $v_i(\mathcal{A}'_i) > v_i(\mathcal{A}_i)$.

Our second line of research [53] initiates work on the α-MMS problem for a mixed manna. We also give the first algorithms for the α-MMS + PO problem, which was open even for the special case of symmetric agents with additive valuations[2] and a goods manna.

For the problem of computing α-MMS allocations for a mixed manna, we first show that for *any* fixed $\alpha \in (0, 1]$, an α-MMS allocation may not always exist; in contrast, non-existence with a goods manna is known only for $\alpha = 39/40$ ([38]). This rules out the possibility of algorithms

[2]open for additive and non-identical valuations, as with identical additive valuations, every allocation is PO

that guarantee to output α-MMS allocations for a fixed α, thus fundamentally changing the kind of questions that can be asked in this setting. The natural most general problem that can be asked is the following.

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

This *exact* problem is intractable: In the case of identical agents, an $(\alpha = 1)$-MMS allocation exists by definition. However, finding one is known to be NP-hard for a goods manna. Also, checking if a given instance admits an MMS allocation is known to be in $\mathsf{NP}^{\mathsf{NP}}$, but not known to be in NP ([20]). On the positive side, a polynomial-time approximation scheme (PTAS) is known for this case due to [79]; given a *constant* $\epsilon \in (0, 1]$, the algorithm finds a $(1 - \epsilon)$-MMS allocation in polynomial time. No such result is known when the agents are not identical. Guaranteeing PO in addition adds to the complexity, since even checking if a given allocation is PO is coNP-hard even with two agents ([5]). In light of these results, we then ask,

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

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

Theorem 1.1. *For any instance $(n, \mathcal{M}, v)$ with identical agents and $v(\mathcal{M}) > 0$ such that exactly one of the following two holds: (a) either $n = 2$ or (b) $|v(\mathcal{M})| \geq \tau \cdot \min\{v^+, v^-\}$ for a constant τ, finding an α-MMS allocation of $(n, \mathcal{M}, v)$ for any $\alpha \in (0, 1]$ is NP-hard.*

To prove the above theorem, we design two reductions from the well-known NP-hard problem Partition to the problem of finding an α-MMS allocation of an instance $(n, \mathcal{M}, v)$ for any $\alpha \in (0, 1]$,. Note that as the agents are identical, an α-MMS allocation always exists for $\alpha = 1$. This hardness is striking because it shows inapproximability within *any* non-trivial factor when either (i) or (ii) is not satisfied. This also indicates that the two conditions are unavoidable without adding some other condition.

Then for instances satisfying the conditions (i) and (ii), we design a PTAS (as asked in the above question). Our algorithm, in principle, gives a little more than a PTAS. It runs in time $2^{O(1/\min\{\epsilon^2, \gamma^2\})} poly(\mathrm{m})$ for given ϵ, γ, thus gives polynomial run-time for ϵ, γ as small as $O(1/\sqrt{\log m})$, where $m = |\mathcal{M}|$.

α-MMS + PO *for goods (chores) manna.* As a corollary, we obtain a PTAS for finding α-MMS + PO allocations of a goods manna and a chores manna when the number of agents is a constant. This improves the previous results for these settings in two aspects: (i) provides the best possible approximation factor; factors better than the general case known for good manna are $4/5$ for $n = 4$ by [45], $8/9$ for $n = 3$ by [46], and 1 for $n = 2$ by [20], and (ii) provides an additional (approximate) PO guarantee.

One key challenge in designing the PTAS is handling items of high value for any agent. In the goods or chores mannas, these items can be greedily assigned, for example as singleton bundles. But in a mixed manna, *high valued* goods (chores) may have to be bundled with specific sets of chores (goods) or low valued items to form lesser valued bundles. To resolve this, we design an LP and a rounding technique to allocate low valued goods optimally.

A second difficulty is that both the MMS values of the agents, and the α for which α-MMS allocation exist, are not known. In fact, computing the exact MMS values is NP-hard (even with a goods manna). As the first key step for our algorithm we design a PTAS that returns $(1-\epsilon)$ approximate MMS values of agents, or equivalently a $(1-\epsilon)$-MMS allocation with identical agents. Our hardness result rules out even approximating the MMS values within any non-trivial factor in polynomial time if either condition is not satisfied. Hence our PTAS is for instances satisfying both. We need to tackle the cases with MMS ≥ 0 and MMS < 0 separately; the sign of the MMS value can be easily determined by observing that MMS is non-negative if and only if the sum of values of all items together is non-negative. Formally, we show the following result.

Theorem 1.2. *Given an instance $(n, \mathcal{M}, v)$ and a constant $\epsilon > 0$, if (i) n is a constant and (ii) for each $i \in [n]$, $|v_i(\mathcal{M})| \geq \tau \cdot \min\{v_i^+, v_i^-\}$, where $\tau > 0$ is a constant. Then, there is a* PTAS *that computes a $(1-\epsilon)$-*MMS *allocation.*

Finally, consider the task of ensuring the PO guarantee. Since certifying a PO allocation is a coNP-hard problem ([5]), known works (e.g., [17, 63, 42]) maintain a PO allocation with market equilibrium as a certificate. We develop a novel LP rounding approach to ensure PO with α-MMS. These could be intuitive if the PO guarantee was not required, and the only purpose was to optimally allocate the low valued items. The rounding now makes use of the popular notion of an *envy-graph* and the properties of the MMS.

The remainder of this article is organized as follows. We start with a review of the literature prior and following our work on the α-NSW and α-MMS problems. Sections 2 and 3 give an overview of our work on respectively the α-NSW and α-MMS(+PO) problems. We end with Section 4 that discusses further work that can be pursued based on our results and techniques, and some concluding remarks.

1.1 Related work

Nash welfare. The NSW was independently discovered by three different communities as a solution of the bargaining problem in classic game theory [66], a well-studied notion of proportional fairness in networking [51], and coincides with the celebrated notion of competitive equilibrium with equal incomes (CEEI) in economics [76]. [48, 50] proposed the notion of asymmetric NSW, which has also been extensively studied, and used in many different applications, e.g., bargaining theory [56, 25, 75], water allocation [47, 34, 33, 35], climate agreements [80].

The NSW problem is NP-hard even for two agents with identical additive valuations, and APX-hard in general [57]. A series of remarkable works [32, 31, 3, 4, 17, 39, 30] provide good approximation guarantees for special sub-classes of this problem where the agents are symmetric and have additive(-like) valuation functions. Slight generalizations of additive valuations have also been studied, namely budget-additive functions [39], separable piecewise linear concave (SPLC) functions [4], and their combination [30].

The NSW objective is also a central notion in fair division. For the case of symmetric agents with additive valuations, Caragiannis et al. [24] present a compelling argument in favor of the 'unreasonable' fairness of maximum NSW by showing that such an allocation has outstanding properties, namely, it is EF1 (a popular fairness property of envy-freeness up to one item) as well as Pareto optimal (PO). Even though computing a maximum NSW allocation is hard, its approximation recovers most of the fairness and efficiency guarantees; see e.g., [17, 30, 63]. The NSW objective also provides an interesting trade-off between the two extremal objectives of social welfare (i.e., sum of valuations) and max-min fairness, and in contrast to both it is invariant to individual scaling of each agent's valuations (see [64] for additional characteristics).

We initiated the study of the asymmetric NSW problem and the problem where agents have submodular valuations [41]. Following our work, [40] and [59] introduced and studied interesting special cases of submodular functions. [58] then showed a constant factor approximation algorithm for the general submodular case. [27], among results for other fairness notions, gave a linear factor algorithm for the subadditive valuations setting, a far more general case than submodular valuations. [12] independently gave another linear factor algorithm for this case, and more generally worked on the p-mean welfare objective, which subsumes NSW as a special case. They give tight approximation algorithms for the p-mean welfare with XOS valuations under value oracles, a special case of subadditive functions. [15] studied the NSW objective under XOS valuations with demand oracles, and give a sub-linear factor algorithm for this setting.

Maximin share. MMS is a popular fairness notion, first introduced by [23]. Since then, the MMS problem has been extensively studied for various special cases. First, for the goods manna setting with additive valuation functions, [20] showed that in some restricted cases MMS allocations always exist. A notable result from [70] showed that MMS allocations may not always exist but 2/3-MMS allocations always do. A series of works studied the efficient computation of 2/3-MMS allocations for any n [1, 16, 43]. [45] showed that a 3/4-MMS allocation always exists. Most recently [44] showed that a $(3/4 + 1/(12n))$-MMS allocation always exists. Finding MMS values is hard but a PTAS for this problem is known [79]. This PTAS can be used to find a $(3/4 + 1/(12n) - \epsilon)$-MMS allocation for $\epsilon > 0$ in polynomial time. There is also a strongly polynomial time algorithm to find 3/4-MMS allocation [44]. Notable works on the goods manna case prior to these state of the art results that introduced novel techniques are [37, 43, 54, 55].

Better approximation algorithms are known when the number of agents is some constant. For three agents, [1] showed that a 7/8-MMS allocation always exists. This factor was later improved to 8/9 in [46]. For four agents, [45] showed that a 4/5-MMS allocation always exist.

The MMS problem has been studied under various other models in the goods manna setting like with asymmetric agents [37], group fairness [13, 28], beyond additive valuations [16, 45, 60], in matroids [46], with additional constraints [46, 19], for agents with externalities [22, 2], with graph constraints [18, 62], and with strategic agents [14]. In the chores manna setting too, weighted MMS [7], and asymmetric agents [6] notions have been investigated.

[10] first studied the MMS problem with a chores manna. They introduced an algorithm for finding 2-MMS allocations (Our definition of α-MMS for the mixed manna is consistent for agents with positive as well as negative MMS values. We define α as smaller than 1, and consider $1/\alpha$-MMS valued bundles as α-MMS. Prior results for the chores manna have $\alpha > 1$ and ask for $\alpha \cdot$MMS valued bundles. We state the approximation factors as defined in the original papers, and ask the reader to invert them when relating with ours). [16] improved the previous result by showing an algorithm for a 4/3-MMS allocation. Later, [49] improved this result to a 11/9-MMS allocation. They also showed a PTAS to find $(11/9 + \epsilon)$-MMS allocation and a polynomial time algorithm to find a 5/4-MMS allocation.

We initiate the study of the MMS problem for a mixed manna [53], and the MMS+PO problem.

2 Nash Welfare

in this section we discuss the details of the asymmetric additive NSW problem. We will prove the following approximation result.

Theorem 2.1. *Given an instance of the asymmetric additive* NSW *problem, algorithm* SMatch *returns an allocation* $\mathbf{x}$ *with* NSW *value at least* $1/2n$ *times the optimal objective value. That is,* $\mathsf{NSW}(\mathbf{x}) \geq \frac{1}{2n}\mathsf{OPT}$.

SMatch is a single pass algorithm that allocates up to one item to every agent per iteration such that the NSW objective is locally maximized. An issue with a naive single pass, locally optimizing greedy approach is that the initial iterations work on highly limited information. As shown in Example 2.1, such algorithms can result in outcomes with very low NSW even for symmetric agents with additive valuation functions.

Example 2.1. *Consider* 2 *agents* A, B *with weights* 1 *each, and* $m+1$ *items. The valuations of* A *and* B *for the first item are* $M+\epsilon$ *and* M *respectively. Agent* A *also values each of the remaining items at* 1*, while* B *only values the last of these at* 1*, and has* 0 *valuation for the remaining* $(m-1)$ *items. An allocation that optimizes the* NSW *of the agents will allocate the first item to* B*, and allocate all the remaining items to* A*. The optimal* NSW *objective is* $(Mm)^{1/2}$*. A repeated matching algorithm, in the first iteration, will allocate the first item to* A*, and the last to* B*. No matching can now give non zero valuation to* B*. The maximum* NSW *objective that can be generated is* $((M+\epsilon+m-1)1)^{1/2} < \sqrt{M+m}$*. Thus, for an appropriate value of* M*, the ratio of* OPT *to* NSW *will depend on* m*.*

In this example, although agent A can be allocated an item of high valuation later, the algorithm does not *know* this initially. Algorithm 1 resolves this issue by pre-computing an approximate value that the agents will receive in later iterations, and uses this information in the edge weight definitions when allocating the first items. Let us see the details of SMatch.

Notation. In the following discussion, we denote the instance by a set $\mathcal{A}$ of n agents with weights η_i for all $i \in \mathcal{A}$, a set $\mathcal{M}$ of m indivisible items, and additive valuations $v_i : 2^{\mathcal{M}} \rightarrow \mathbb{R}_+$, where $v_i(\mathcal{S})$ is the value of agent $i \in \mathcal{A}$ for a set of items $\mathcal{S} \subseteq \mathcal{M}$. Let $\mathbf{x}_i = \{h_i^1, \ldots, h_i^{\tau_i}\}$ denote the set of items received by agent i in SMatch. We use $\mathbf{x}_i^* = \{g_i^1, \ldots, g_i^{\tau_i^*}\}$, τ_i and τ_i^* to denote the set of items in i's optimal bundle, and the number of items in $\mathbf{x}_i$ and $\mathbf{x}_i^*$ respectively. Then for every i, all items in $\mathbf{x}_i$ and $\mathcal{M}$ are ranked according to the decreasing utilities as per v_i. $\mathcal{M}_{i,[a:b]}$ denote the items ranked from a to b according to agent i in $\mathcal{M}$, and $\mathbf{x}_{i,1:t}$ is the total allocation to agent i from the first t matching iterations. We also use $\mathcal{M}_{i,k}$ to denote the k^{th} ranked item of agent i from the entire set of items. For all i, we define u_i as the minimum value for the remaining set of items upon removing at most $2n$ items from $\mathcal{M}$, i.e., $u_i = \min_{\mathcal{S} \subseteq \mathcal{M}, |\mathcal{S}| \leq 2n} v_i(\mathcal{M} \setminus \mathcal{S}) = \mathcal{M}_{i,[2n+1,m]}$.[3]

Algorithm. SMatch works in a single pass. For every agent, the algorithm first computes the value of $m-2n$ least valued items and stores this in u_i. SMatch then defines a weighted complete bipartite graph $\Gamma(\mathcal{A}, \mathcal{M}, \mathcal{W})$ with the agents and goods forming the two sets of vertices, and the edge weights defined as $w(i,j) = \eta_i \log\left(v_i(j) + \frac{u_i}{n}\right)$. It allocates one item to each agent along the edges of a maximum weight matching of Γ. It then starts allocating items via repeated matchings. Until all the items are allocated, SMatch iteratively defines graphs $\Gamma(\mathcal{A}, \mathcal{M}^{rem}, \mathcal{W})$ with $\mathcal{M}^{rem}$ denoting the set of unallocated items and edge weights defined as $w(i,j) = \eta_i \log\left(v_i + v_i(j)\right)$, where v_i is the

[3] As the valuation functions are monotone, the minimum value will be obtained by removing exactly $2n$ items. The less than accounts for the case when the number of items in $\mathcal{M}$ is fewer than $2n$.

valuation of agent i for items that are allocated to her. SMatch then allocates at most one item to each agent according to a maximum weight matching of Γ.

Algorithm 1: SMatch for the Asymmetric Additive NSW problem

Input : A set $\mathcal{A}$ of n agents with weights η_i, $\forall i \in \mathcal{A}$, a set $\mathcal{M}$ of m indivisible items, and additive valuations $v_i : 2^{\mathcal{M}} \to \mathbb{R}_+$, where $v_i(\mathcal{S})$ is the valuation of agent $i \in \mathcal{A}$ for a set of items $\mathcal{S} \subseteq \mathcal{M}$.

Output: An allocation that approximately optimizes the NSW.

$\mathbf{x}_i \leftarrow \emptyset, u_i \leftarrow v_i(\mathcal{M}_{i,[2n+1:m]}) \quad \forall i \in [n]$ // value of all except the top $2n$ goods
Define weighted complete bipartite graph $\Gamma(\mathcal{A}, \mathcal{M}, \mathcal{W})$ with weights
  $\mathcal{W} = \{w(i,j) \mid w(i,j) = \eta_i \log\left(v_i(j) + \frac{u_i}{n}\right), \forall i \in \mathcal{A}, j \in \mathcal{M}\}$
Compute a maximum weight matching $\mathcal{M}$ for Γ
$\mathbf{x}_i \leftarrow \mathbf{x}_i \cup \{j \mid (i,j) \in \mathcal{M}\}, \ \forall i \in \mathcal{A}$ // allocate items according to $\mathcal{M}$
$\mathcal{M}^{rem} \leftarrow \mathcal{M} \backslash \{j \mid (i,j) \in \mathcal{M}\}$ // update set of unallocated items
while $\mathcal{M}^{rem} \neq \emptyset$ **do**
  Define weighted complete bipartite graph $\Gamma(\mathcal{A}, \mathcal{M}^{rem}, \mathcal{W})$ with weights
   $\mathcal{W} = \{w(i,j) \mid w(i,j) = \eta_i \log(v_i(j) + v_i(\mathbf{x}_i)), i \in \mathcal{A}, j \in \mathcal{M}^{rem}\}$
  Compute a maximum weight matching $\mathcal{M}$ for Γ
  $\mathbf{x}_i \leftarrow \mathbf{x}_i \cup \{j \mid (i,j) \in \mathcal{M}\}, \ \forall i \in \mathcal{A}$ // allocate items according to $\mathcal{M}$
  $\mathcal{M}^{rem} \leftarrow \mathcal{M}^{rem} \backslash \{j \mid (i,j) \in \mathcal{M}\}$ // remove allocated items
end
Return $\mathbf{x}$

To establish the guarantee of Theorem 2.1, we first prove two technical lemmas.

Lemma 2.1. $v_i(h_i^t) \geq v_i(\mathcal{M}_{i,tn})$.

Proof. Since every iteration of SMatch allocates at most n items, at the start of iteration t at most $(t-1)n$ items are allocated. Thus at least n items from $\mathcal{M}$ ranked between 1 to tn by agent i are still unallocated. In the t^{th} iteration the agent will thus get an item with value at least the value of $\mathcal{M}_{i,tn}$ and the lemma follows. □

Lemma 2.2. $v_i(h_i^2, \ldots, h_i^{\tau_i}) \geq \frac{u_i}{n}$.

Proof. Using Lemma 2.1 and since $v_i(\mathcal{M}_{i,tn}) \geq v_i(\mathcal{M}_{i,tn+k}), \ \forall k \in [n-1]$

$$v_i(h_i^t) \geq \frac{1}{n} v_i(\mathcal{M}_{i,[tn:(t+1)n-1]}) \ .$$

Thus,

$$v_i(h_i^2, \ldots, h_i^{\tau_i}) = \sum_{t=2}^{\tau_i} v_i(h_i^t) \geq \frac{1}{n} \sum_{t=2}^{\tau_i} (v_i(\mathcal{M}_{i,[tn:(t+1)n-1]})$$

As at most n items are allocated in every iteration, agent i receives items for at least $\lfloor \frac{m}{n} \rfloor$ iterations.[4]

[4] Here we assume that the agents have non-zero valuation for every item. If it does not, the other case is also straightforward and the lemma continues to hold.

This implies that $(\tau_i + 1)n \geq m$ and hence,

$$\begin{aligned} v_i(h_i^2, \ldots, h_i^{\tau_i}) &\geq \frac{1}{n}(v_i(\mathcal{M}_{i,[2n:m-1]})) \\ &\geq \frac{1}{n}(v_i(\mathcal{M}_{i,[2n+1:m]})) = \frac{1}{n}u_i. \end{aligned}$$

The second inequality follows as $v_i(\mathcal{M}_{i,2n}) \geq v_i(\mathcal{M}_{i,m})$. □

We now prove our main theorem. Let W denote the sum of agent weights, that is, $W = \sum_i \eta_i$.

Proof of Theorem 2.1.

$$\begin{aligned} \mathsf{NSW}(\mathbf{x}) &= \prod_{i=1}^{n} \left(v_i \left(h_i^1, \ldots, h_i^{\tau_i} \right)^{\eta_i} \right)^{\frac{1}{W}} \\ &= \prod_{i=1}^{n} \left(\left(v_i(h_i^1) + v_i(h_i^2 \ldots, h_i^{\tau_i}) \right)^{\eta_i} \right)^{\frac{1}{W}} \\ &\geq \prod_{i=1}^{n} \left(\left(v_i(h_i^1) + \frac{u_i}{n} \right)^{\eta_i} \right)^{\frac{1}{W}}, \end{aligned}$$

where the last inequality follows from Lemma 2.2. During the allocation of the first item h_i^1, items g_i^1 of all agents are available. Thus, allocating each agent her own g_i^1 is a feasible first matching and we get

$$\mathsf{NSW}(\mathbf{x}) \geq \prod_{i=1}^{n} \left(\left(v_i(g_i^1) + \frac{u_i}{n} \right)^{\eta_i} \right)^{\frac{1}{W}}.$$

Now, $u_i = \min_{\mathcal{S} \in \mathcal{M}, |\mathcal{S}| \leq 2n} v_i(\mathcal{M} \setminus \mathcal{S})$. Suppose we define, $\mathcal{S}_i^* = \arg\min_{|\mathcal{S}| \leq 2n, \mathcal{S} \subseteq \mathbf{x}_i^*} v_i(\mathbf{x}_i^* \setminus \mathcal{S})$, then $v_i(\mathbf{x}_i^* \setminus \mathcal{S}_i^*) \leq u_i$. To see this, let $\mathcal{S}_i = \arg\min_{\mathcal{S} \in \mathcal{M}, |\mathcal{S}| \leq 2n} v_i(\mathcal{M} \setminus \mathcal{S})$. Now, $u_i = v_i(\mathcal{M} \setminus \mathcal{S}_i) \geq v_i(\mathbf{x}_i^* \setminus \mathcal{S}_i) \geq v_i(\mathbf{x}_i^* \setminus \mathcal{S}_i^*)$. Thus,

$$\begin{aligned} \mathsf{NSW}(\mathbf{x}) &\geq \prod_{i=1}^{n} \left(\left(\frac{1}{2n} v_i(\mathcal{S}_i^*) + \frac{1}{n} v_i(\mathbf{x}_i^* \setminus \mathcal{S}_i^*) \right)^{\eta_i} \right)^{\frac{1}{W}} \\ &\geq \frac{1}{2n} \prod_{i=1}^{n} \left(v_i(\mathbf{x}_i^*)^{\eta_i} \right)^{\frac{1}{W}} \\ &= \frac{1}{2n} \mathsf{OPT}. \end{aligned}$$

□

Remark 2.1. SMatch *extends to give an $O(n)$ approximation for the case of budget-additive valuations. The changes required in the algorithm are $u_i = min(c_i, \mathcal{M}_{1,2n+1:m})$ where c_i is the utility cap for agent i. Also, the edge weights in the bipartite graphs will use the marginal utility of each item over already received items. We leave the proof details for the full version.*

Remark 2.2. *When* SMatch *is applied to the instance of Example 2.1, it results in a better allocation than that of a naive repeated matching approach. Stage 1 of* SMatch *computes u_i as $m - 2n$ and 0 for A and B respectively. When this value is included in the edge weight of the first bipartite graph Γ, the resulting matching gives B the first item, and A some other item. Subsequently A gets all remaining items, resulting in an allocation having optimal* NSW.

2.1 Overview of techniques for other results

Algorithms RepReMatch. The main idea used in this algorithm too is informally as follows.

Lemma (informal). *For $k = O(n)$ and for every agent i, after removing a set S_i of k items that minimizes i's valuation for the remaining items, repeatedly matching the remaining items $\mathcal{M}\backslash(\cup_i S_i)$ to locally maximize the NSW objective gives every agent an allocation of value at least a $1/n$ fraction of her valuation for the remaining set of items, i.e, $v_i(\mathcal{M}\backslash(\cup_i S_i))/n$.*

It is known from [74] that finding a set of minimum valuation with lower bounded cardinality for monotone submodular functions is inapproximable within $\sqrt{m/\ln m}$ factor, where m is the number of items. Hence, the above lemma by itself is insufficient for the submodular NSW problem. For this case, we prove a lemma that implies the following statement.

Lemma (informal). *When items are iteratively matched to agents to locally maximize the weighted geometric mean of agents' valuations from their matched items, then if the set of items allocated in the first $\log n + 1$ matchings are released, then computing one (re)-match with these results in a matching where every agent gets of value at least as much as the highest valued item from her NSW optimizing allocation.*

RepReMatch combines these ideas by performing a repeated matching, un-matching the first $O(\log n)$ matchings and then re-matching them.

Submodular NSW with constant number of agents. This is a different approach that uses techniques of maximizing submodular functions over matroids developed in [29], and a reduction of fair division problems to the problem of maximizing a submodular function over matroids from [77]. At a high level, we first guess the values of all agents in the optimal NSW allocation. Then we consider the multilinear extensions of agent valuation functions, and using the following result find a fractional allocation that gives every agent a bundle of the guessed optimal value.

Theorem 2.2. *[29] Consider monotone submodular functions $f_1, \ldots, f_n : 2^N \to \mathbb{R}_+$, their multilinear extensions $F_i : [0,1]^N \to \mathbb{R}_+$ and a matroid polytope $P \subseteq [0,1]^N$. There is a polynomial time algorithm which, given $V_1, ..., V_n \in \mathbb{R}_+$, either finds a point $x \in P$ such that $F_i(x) \geq (1 - 1/e)V_i$ for each i, or returns a certificate that there is no point $x \in P$ such that $F_i(x) \geq V_i$ for all i.*

This allocation is then rounded using a swap rounding randomized algorithm to obtain an integral allocation of items. The following lower tail bound proves that with high probability, the loss in the function value by swap rounding is not too much.

Theorem 2.3. *[29] Let $f : \{0,1\}^n \to \mathbb{R}_+$ be a monotone submodular function with marginal values in $[0,1]$, and $F : [0,1]^n \to \mathbb{R}_+$ its multilinear extension. Let $(x_1, ..., x_n) \in P(M)$ be a point in a matroid polytope and $(X_1, ..., X_n) \in \{0,1\}^n$ a random solution obtained from it by randomized swap rounding. Let $\mu_0 = F(x_1, ..., x_n)$ and $\delta > 0$. Then*

$$Pr[f(X_1, ..., X_n) \leq (1 - \delta)\mu_0] \leq e^{-\mu_0 \delta^2/8}.$$

Hardness of approximation. The submodular ALLOCATION problem is to maximize the sum of valuations of agents over integral allocations of items. [52] describe a reduction of MAX-3-COLORING, which is NP-Hard to approximate within a constant factor, to ALLOCATION. We prove that this reduction also establishes the same hardness for the submodular NSW problem.

3 Maximin Share

In this section we present an overview of our main result, namely a PTAS for the OPT-α-MMS+PO problem. The key is to design a PTAS for finding an α-MMS and PO allocation for a given α, referred as the α-MMS + PO problem. OPT-α-MMS + PO can then be solved by performing any exponentially fast search for the optimal value of α.

In what follows, $[k]$ denotes the set $\{1, 2 \cdots, k\}$, and for $c \in \mathbb{R}$, c^+ denotes $\max\{c, 0\}$.

Our PTAS for the α-MMS + PO problem takes as input an instance denoted by $(\mathcal{N}, \mathcal{M}, \mathcal{V})$, where $\mathcal{N}$ is the set of agents, $\mathcal{M}$ the manna of items, and $\mathcal{V}$ is the set of valuation functions of all agents, a parameter $\alpha \in (0, 1]$, and constants $\epsilon, \gamma > 0$, and it either finds an allocation that is $(\alpha - \epsilon)^+$-MMS+γ-PO allocation, or correctly reports that an α-MMS allocation does not exist; the latter may very well be the case for *any* $\alpha \in (0, 1]$ as shown by our non-existence result.

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

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

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

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

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

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

Finally, we show that the rounded solution, call it $\mathcal{A}^r$, is γ-PO, by showing that for an allocation to γ-Pareto dominate $\mathcal{A}^r$, it must be an α-MMS allocation and have higher welfare than $\mathcal{A}$. This proof is quite involved and uses several new ideas, including the way we round the LP solution, to prove Pareto optimality of the integral allocation.

In the remaining section, we will formalize the ideas for the OPT-α-MMS problem, with some proofs removed for brevity. Also, at times we will refer to $\overline{\mathsf{MMS_i}}$ as $\tilde{\mu}_i$ and to $\mathsf{MMS_i}$ as μ_i.

The following bound on the $\mathsf{MMS_i}$ values will be useful in the analysis, and follows from the observation that the MMS value of an agent is at most the proportional value of all items, and that $|v_i(\mathcal{M})| = n,\ \forall i \in \mathcal{N}$.

Lemma 3.1. *For each agent $i \in \mathcal{N}$, $\mathsf{MMS_i} \le 1$ if $v(\mathcal{M}) \ge 0$, otherwise $\mathsf{MMS_i} \le -1$.*

3.1 BIG and SMALL Items

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

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

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

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

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

We will show the size of BIG is constant. For this, we make two useful observations.

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

The claim follows from the guarantees of the PTAS for computing MMS values. The next claim follows from condition 2 of the problem.

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

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

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

3.2 LP for Allocating SMALL Items, and Rounding

Given a partition $B^\pi = (B_1, \ldots, B_n)$ of BIG items, next we show an LP to find a *fractional* allocation of SMALL items such that together with B^π this allocation gives at least α-$\overline{\mathsf{MMS}}$ value to every agent. If there exists an α-MMS allocation where the BIG items are allocated as per B^π then we show that the LP has to be feasible.

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

$$\max \quad \sum_{i \in \mathcal{N}} \left(\sum_{j \in \mathsf{SMALL}_i^+} v_{ij} x_{ij} - \sum_{j \in \mathsf{SMALL}_i^-} |v_{ij}| x_{ij} \right) \tag{4}$$

$$\text{s.t.} \quad \sum_{j \in \mathsf{SMALL}_i^+} v_{ij} x_{ij} - \sum_{j \in \mathsf{SMALL}_i^-} |v_{ij}| x_{ij} \geq c_i, \qquad \forall i \in \mathcal{N} \tag{5}$$

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

$$\sum_{i \in \mathcal{N}} x_{ij} \geq 1, \qquad \forall j \in \mathsf{SMALL}^- \tag{7}$$

$$x_{ij} \geq 0, \qquad \forall i \in \mathcal{N}, j \in \mathcal{M}. \tag{8}$$

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

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

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

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

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

Next we describe the notion of envy graph from [61], a directed graph corresponding to any allocation, that will be used to round the LP solutions.

Envy Graph and Cycle Elimination. Given a set of agents $\mathcal{N}$ and an *integral* allocation $\mathcal{A}$ of a set of items among them, each node in the graph corresponds to an agent in $\mathcal{N}$. There is a directed edge $(i \to k)$ corresponding to agents i and k if agent i values agent k's allocation more than her own. It is shown in [61] that the allocation can be modified so that its corresponding envy graph is acyclic, and no agent's valuation decreases. This is done by giving each agent in a cycle the bundle of her successor. The graph is updated and the process repeated until all cycles are eliminated. This process can be done efficiently, as shown in [61].

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

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

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

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

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

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

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

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

The algorithm combines these ideas by enumerating all partitions of the BIG items and for each partition, trying to solve the LP and rounding it. We finish by showing that the algorithm returns an $(\alpha-\epsilon)^+$-MMS allocation if an α-MMS allocation exists.

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

Proof. Consider agent i^t. Since i^t corresponds to a sink node in the envy graph, from Claim 3.3 and Lemmas 3.5 and 3.1, her value for her bundle in $\mathcal{A}^r$ is at least $1-\epsilon/2 \geq 1-\epsilon \geq (\alpha-\epsilon)\mu_{i^t}$ if $\tilde{\mu}_{i^t} \geq 0$, and $-1-\epsilon/2 \geq -1-\epsilon \geq (1+\epsilon)\mu_{i^1} \geq \frac{1}{(\alpha-\epsilon)}\mu_{i^1}$ otherwise. Next, every agent i except i^t, according to constraint (5) of the LP, receives a bundle of value at least c_i from SMALL in the fractional allocation of SMALL corresponding to $\mathcal{A}$. Thus, for all $i \neq i^t$, their value for their bundle in $\mathcal{A}^r$ is at least $v_i(B_i) + c_i - n\epsilon\cdot|\tilde{\mu}_i|/(2n) \geq \min\{({}^1/_\alpha)\tilde{\mu}_i, \alpha\tilde{\mu}_i\} - \epsilon\cdot|\tilde{\mu}_i|/2$, from Lemma 3.5 and by definition of c_i. Combined with Claim 3.1, when $\tilde{\mu}_i \geq 0$, this is at least $(\alpha-\epsilon/2)\tilde{\mu}_i \geq (\alpha-\epsilon/2)(1-\epsilon/2)\mu_i \geq (\alpha-\epsilon)\mu_i$. When $\tilde{\mu}_i < 0$, this value is at least $\frac{1}{\alpha}\tilde{\mu}_i + \epsilon\tilde{\mu}_i/2$. As $(\frac{1}{\alpha}+\epsilon/2) \leq \frac{1}{(\alpha-\epsilon/2)}$, and $\tilde{\mu}_i < 0$, along with Claim 3.1, $(\frac{1}{\alpha}+\epsilon/2)\tilde{\mu}_i \geq \frac{1}{(\alpha-\epsilon/2)(1-\epsilon/2)}\mu_i \geq \frac{1}{(\alpha-\epsilon)}\mu_i$. □

Since by construction, the LP has to be feasible whenever BIG items are allocated as per an α-MMS allocation, we get as a corollary that if an α-MMS allocation exists, the algorithm returns an $(\alpha - \epsilon)^+$-MMS allocation.

4 Further Work

This article documents our progress on the problem of computing fair and efficient divisions of indivisible items. Our work opens several avenues. We discuss some of these in this section.

4.1 The MMS + PO problem for a goods or a chores manna

One can either work on the OPT-α-MMS problem for a goods manna with non-constantly many agents, or a weaker version, such as computing a 1/2-MMS + PO allocation. Note that when an α-MMS allocation exists, then so does an α-MMS + PO allocation, as allocations that Pareto dominate an α-MMS allocation are also α-MMS. While simple greedy techniques guarantee a 1/2-MMS allocation, ensuring PO along with this is still open. Unlike for the mixed manna, high valued items can usually be assigned easily with a goods or a chores manna. Hence, one can presume that the constantly many agents assumption can be relaxed. Our LP and rounding techniques from [53] ensure a PO allocation of the low valued items, which may help in designing such an algorithm.

A challenge for this problem is the following non-existence result by observing the example from [12]. They give an example showing that an allocation that is simultaneously a 1.44 factor approximately optimal NSW and also fPO (fractionally Pareto optimal) cannot exist. The same example, extended to n agents instead of 3, shows that a linear factor MMS and fPO allocation cannot exist. Thus, if one were to resolve PO, then market based techniques will probably not work as they ensure fPO, and not just PO. A novel technique to ensure PO, possibly an extension of the one we show in [53], may help here.

4.2 Any price share (APS)

A novel fairness notion called the Any price share (APS) was recently introduced in [11]. The paper establishes that the APS value of every agent is at least their MMS value. However, [11] also show instances where the APS value of an agent is strictly better than their MMS value. Hence, unlike the MMS problem, there is no notion of an APS allocation, as APS is a value defined independent of allocations of the items. Also, as the highest value that *every* agent receives in any allocation with identical agents is at most their MMS, this shows that allocations where every agent receives a bundle of APS or higher value, termed APS allocations by [11], may not exist even with identical agents.

The relations between APS and MMS from [11] show that the APS and MMS are closely related, unlike previous notions like the NSW or envy-related notions: there are counter examples that show allocations optimal for these other notions have a poor approximation guarantee for MMS. Further, the primal and dual definitions of the APS value are insightful. [11] also give pseudopolynomial time algorithms to compute the APS values of agents using these. These observations encourage the possibility of algorithms to compute α-APS allocations, for α better than the state of the art known for the MMS problem. As APS is better than MMS, this will lead to improving the best known results for the MMS problem as well.

4.3 Nash welfare with asymmetric agents and beyond additive valuations

A series of works has improved upon our work on the NSW problem with symmetric agents and valuation functions that are far more general than additive [See Section 1.1 for details]. An interesting question to resolve is to find a sublinear factor approximation to the optimal NSW for such valuation functions for the asymmetric agents case.

5 Conclusion

We study the fair and efficient division problem of indivisible items for two notions, the NSW [41] and MMS + PO [53]. Our works have introduced new problems that have already resulted in extensive follow up work, and has introduced new problems and new techniques. In the future, we wish to leverage this work by modifying and developing more techniques and study this problem further.

References

[1] Georgios Amanatidis, Evangelos Markakis, Afshin Nikzad, and Amin Saberi. Approximation algorithms for computing maximin share allocations. *ACM Trans. Algorithms*, 13(4):52:1–52:28, 2017.

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

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

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

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

[6] Haris Aziz, Hau Chan, and Bo Li. Maxmin share fair allocation of indivisible chores to asymmetric agents. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1787–1789.

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

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

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

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

[11] Moshe Babaioff, Tomer Ezra, and Uriel Feige. Fair-share allocations for agents with arbitrary entitlements. *arXiv preprint arXiv:2103.04304*, 2021.

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

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

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

[15] Siddharth Barman, Anand Krishna, Pooja Kulkarni, and Shivika Narang. Sublinear approximation algorithm for nash social welfare with xos valuations. *arXiv preprint arXiv:2110.00767*, 2021.

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

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

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

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

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

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

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

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

[24] Ioannis Caragiannis, David Kurokawa, Herve Moulin, Ariel Procaccia, Nisarg Shah, and Junxing Wang. The unreasonable fairness of maximum Nash welfare. In *17th*, pages 305–322, 2016.

[25] Suchan Chae and Herve Moulin. Bargaining among groups: an axiomatic viewpoint. *International Journal of Game Theory*, 39:71–88, 2010.

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

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

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

[29] Chandra Chekuri, Jan Vondrak, and Rico Zenklusen. Dependent randomized rounding via exchange properties of combinatorial structures. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 575–584. IEEE, 2010.

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

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

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

[33] Dagmawi Mulugeta Degefu, Weijun He, and Liang Yuan. Monotonic bargaining solution for allocating critically scarce transboundary water. *Water Resources Management*, 31:2627–2644, 2017.

[34] Dagmawi Mulugeta Degefu, Weijun He, Liang Yuan, and Jian Hua Zhao. Water allocation in transboundary river basins under water scarcity: a cooperative bargaining approach. *Water Resources Management*, 30:4451–4466, 2016.

[35] Dagmawi Mulugeta Degefu, He Weijun, Yuan Liang, Min An, and Zhang Qi. Bankruptcy to surplus: Sharing transboundary river basin's water under scarcity. *Water Resources Management*, 32:2735–2751, 2018.

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

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

[38] U. Feige, Ariel Sapir, and Laliv Tauber. A tight negative example for mms fair allocations. *ArXiv*, abs/2104.04977, 2021.

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

[40] Jugal Garg, Edin Husić, and László A Végh. Approximating nash social welfare under rado valuations. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1412–1425, 2021.

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

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

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

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

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

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

[47] Houba H, Van der Laan G, and Zeng Y. Asymmetric Nash solutions in the river sharing problem. *Strategic Behavior and the Environment*, 4:321–360, 2014.

[48] J. Harsanyi and R. Selten. A generalized Nash solution for two-person bargaining games with incomplete information. *Management Science*, 18:80–106, 1972.

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

[50] E. Kalai. Nonsymmetric Nash solutions and replications of 2-person bargaining. *International Journal of Game Theory*, 6:129–133, 1977.

[51] Frank Kelly. Charging and rate control for elastic traffic. *European Transactions on Telecommunications*, 8:33–37, 1997.

[52] Subhash Khot, Richard Lipton, Evangelos Markakis, and Aranyak Mehta. Inapproximability results for combinatorial auctions with submodular utility functions. 52(1):3–18, 2008.

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

[54] David Kurokawa, Ariel D. Procaccia, and Junxing Wang. When can the maximin share guarantee be guaranteed? In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, page 523–529. AAAI Press, 2016.

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

[56] Annick Laruellea and Federico Valenciano. Bargaining in committees as an extension of Nash's bargaining theory. *Journal of Economic Theory*, 132:291–305, 2007.

[57] Euiwoong Lee. Apx-hardness of maximizing nash social welfare with indivisible items. *Information Processing Letters*, 122:17–20, 2017.

[58] Wenzheng Li and Jan Vondrák. A constant-factor approximation algorithm for nash social welfare with submodular valuations. *arXiv preprint arXiv:2103.10536*, 2021.

[59] Wenzheng Li and Jan Vondrák. Estimating the nash social welfare for coverage and other submodular valuations. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1119–1130. SIAM, 2021.

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

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

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

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

[64] Herve Moulin. *Fair Division and Collective Welfare.* MIT Press, 2003.

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

[66] John Nash. The bargaining problem. *Econometrica*, 18(2):155–162, 1950.

[67] Nhan-Tam Nguyen, Trung Thanh Nguyen, Magnus Roos, and Jorg Rothe. Computational complexity and approximability of social welfare optimization in multiagent resource allocation. *Autonom. Agents & Multi-Agent Syst.*, 28(2):256–289, 2014.

[68] Trung Thanh Nguyen and Jörg Rothe. Minimizing envy and maximizing average Nash social welfare in the allocation of indivisible goods. *Discrete Applied Mathematics*, 179:54–68, 2014.

[69] Noam Nisan, Éva Tardos, Tim Roughgarden, and Vijay Vazirani, editors. *Algorithmic Game Theory.* 2007.

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

[71] Jack Robertson and William Webb. *Cake-cutting algorithms: Be fair if you can.* CRC Press, 1998.

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

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

[74] Zoya Svitkina and Lisa Fleischer. Submodular approximation: Sampling-based algorithms and lower bounds. *SIAM Journal on Computing*, 40(6):1715–1737, 2011.

[75] W. Thomson. Replication invariance of bargaining solutions. *Int. J. Game Theory*, 15:59–63, 1986.

[76] Hal R Varian. Equity, envy, and efficiency. *Journal of Economic Theory*, 9(1):63 – 91, 1974.

[77] Jan Vondrák. Optimal approximation for the submodular welfare problem in the value oracle model. In *40th*, pages 67–74, 2008.

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

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

[80] S. Yu, E. C. van Ierland, H.-P. Weikard, and X. Zhu. Nash bargaining solutions for international climate agreements under different sets of bargaining weights. *International Environmental Agreements: Politics, Law and Economics*, 17:709–729, 2017.

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