Parameterized Guarantees for Almost Envy-Free Allocations

Siddharth Barman
Indian Institute of Science
Bangalore, India
barman@iisc.ac.in

Debajyoti Kar
Indian Institute of Science
Bangalore, India
debajyotikar@iisc.ac.in

Shraddha Pathak
Penn State University
State College, USA
shraddha.sunilpathak@gmail.com

ABSTRACT

We study fair allocation of indivisible goods among agents with additive valuations. We obtain novel approximation guarantees for two of the strongest fairness notions in discrete fair division, namely envy-free up to the removal of any positively-valued good (EFx) and pairwise maximin shares (PMMS). Our approximation guarantees are in terms of an instance-dependent parameter $\gamma \in (0, 1]$ that upper bounds, for each indivisible good in the given instance, the multiplicative range of *nonzero* values for the good across the agents.

First, we consider allocations wherein, between any pair of agents and up to the removal of any positively-valued good, the envy is multiplicatively bounded. Specifically, the current work develops a polynomial-time algorithm that computes a $\left(\frac{2\gamma}{\sqrt{5+4\gamma}-1}\right)$-approximately EFx allocation for any given fair division instance with range parameter $\gamma \in (0, 1]$. For instances with $\gamma \geq 0.511$, the obtained approximation guarantee for EFx surpasses the previously best-known approximation bound of $(\phi - 1) \approx 0.618$, here ϕ denotes the golden ratio.

Furthermore, we study multiplicative approximations for PMMS. For fair division instances with range parameter $\gamma \in (0, 1]$, the current paper develops a polynomial-time algorithm for finding allocations wherein the PMMS requirement is satisfied, between every pair of agents, within a multiplicative factor of $\frac{5}{6}\gamma$. En route to this result, we obtain novel existential and computational guarantees for $\frac{5}{6}$-approximately PMMS allocations under restricted additive valuations.

KEYWORDS

Discrete Fair Division; Envy-Freeness up to Any Good (EFx); Pairwise Maximin Share (PMMS); Restricted Additive Valuations

ACM Reference Format:
Siddharth Barman, Debajyoti Kar, and Shraddha Pathak. 2024. Parameterized Guarantees for Almost Envy-Free Allocations. In *Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), Auckland, New Zealand, May 6 – 10, 2024*, IFAAMAS, 9 pages.

1 INTRODUCTION

Discrete fair division is an active field of research that studies fair allocation of indivisible goods among agents with individual preferences [6, 39]. Extending the reach of classic fair division literature, which predominantly focused on divisible goods [14, 38], this budding field addresses resource-allocation settings wherein the underlying resources have to be integrally allocated and cannot be fractionally assigned among the agents. Motivating real-world instantiations of such allocation settings include fair division of public housing units [11, 23], courses [15], and food donations [5, 41]. The widely-used platform Spliddit.org [33, 43] provides fair division solutions for oft-encountered allocation scenarios.

Classic fairness criteria—such as envy-freeness [30, 44] and proportionality [24]—were formulated considering divisible goods and cannot be upheld in the context of indivisible goods. In particular, envy-freeness requires that every agent is assigned a bundle that she values at least as much as any other agent's bundle, and, indeed, an envy-free allocation does not exist if we have to assign a single indivisible good among multiple agents. Motivated by such considerations, a key conceptual thrust in discrete fair division has been the development of meaningful analogs of classic fairness notions.

A notably compelling analog of envy-freeness is obtained by considering envy-freeness up to the removal of any positively-valued good (EFx). Introduced by Caragiannis et al. [18], EFx considers the elimination of envy between any pair of agents via the hypothetical removal of any positively-valued good from the other agent's bundle.[1] Despite significant interest and research efforts, the universal existence of EFx allocations remains unsettled. In fact, establishing the existence of EFx allocations is widely regarded as one of central open questions in discrete fair division [42].

As yet, the existence of *exact* EFx allocations has been established in rather specific settings; see related works mentioned below. Hence, to obtain fairness guarantees in broader contexts, recent works have further focused on computing allocations that are approximately EFx.[2] In particular, the work of Plaut and Roughgarden [40] provides an exponential-time algorithm for finding allocations wherein, between any pair of agents and up to the removal of any good, the envy is multiplicatively bounded by a factor of $1/2$; this result for $\frac{1}{2}$-approximately EFx allocations holds under subadditive valuations. Subsequently, for additive valuations Amanatidis et al. [8] developed a polynomial-time algorithm for finding $(\phi - 1)$-approximately EFx allocations, here ϕ is the golden ratio. Farhadi et al. [26] developed an efficient algorithm with the same approximation factor of $(\phi - 1) \approx 0.618$.

The current work contributes to this thread of research by developing novel approximation guarantees for EFx and related fairness notions. Our guarantees are in terms of an instance-dependent parameter $\gamma \in (0, 1]$. In particular, for each (indivisible) good g in the given fair division instance, we define the range parameter

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

[1]As in [18], we consider EFx with the removal of any *positively-valued* good from the other agent's bundle. A stronger version, wherein one requires envy-freeness up to the removal of every good (positively valued or not), has also been considered in the literature; see, e.g., [21].

[2]Such algorithmic results imply commensurate existential guarantees for approximately EFx allocations.

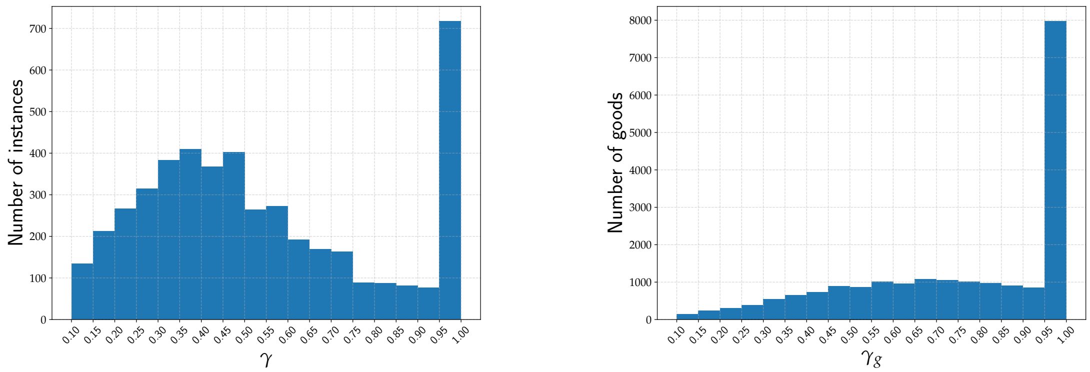

Figure 1: Histograms from Spliddit.org data.

$\gamma_g \in (0, 1]$ as ratio between the smallest *nonzero* value of g and largest value for g among the agents (see equation (1)). Furthermore, we define the range parameter γ for the given instance as the minimum γ_g across all goods g. Hence, the parameter γ bounds, for each indivisible good g, the multiplicative range of *nonzero* values for g among the agents.

Note that a high $\gamma \in (0, 1]$ does not rule out drastically different values across different goods. In particular, for restricted additive valuations (see, e.g., [4]) the parameter $\gamma = 1$; recall that, under restricted additive valuations, every good g has an associated base value, and each agent either values g at zero or at its base value. Further, $\gamma = 1$ under binary, additive valuations. Also, for instances in which there are only two possible values, $0 < a < b$, for all the goods, we have $\gamma = a/b$; such instances have been studied recently in fair division; see, e.g., [3, 25, 27, 32].

Interestingly, higher values of γ are prevalent across user submitted instances at Spliddit.org [33, 43]. As mentioned previously, this platform has been widely used to solve, in particular, discrete fair division problems submitted by users. We consider the data gathered at Spliddit.org (over several years) and evaluate the range parameters of the goods and instances: Figure 1 plots the number of (user-submitted) discrete fair division instances associated with different bins of the range parameter $\gamma \in (0, 1]$. The figure also provides a similar histogram for the range parameters γ_g of the goods g. Here, for every submitted instance and each participating agent, we set the goods' values below 10% of the agent's value for the grand bundle to zero – we perform this calibration to account for users' aversion toward reporting zero valuations. Overall, these observations highlight the relevance of focussing on fair division instances with relatively high range parameter γ; specifically, 85.23% indivisible goods have $\gamma_g \geq 0.51$ and 47.53% of submitted instances have range parameter $\gamma \geq 0.51$.

Our Results and Techniques. We develop a polynomial-time algorithm that computes a $\left(\frac{2\gamma}{\sqrt{5+4\gamma}-1}\right)$-EFx allocation for any given fair division instance with additive valuations and range parameter $\gamma \in (0, 1]$ (Theorem 3.8 in Section 3). Note that the obtained approximation factor $\frac{2\gamma}{\sqrt{5+4\gamma}-1} > \gamma$ for all $\gamma \in (0, 1)$; see Figure 2. Furthermore, for any $\gamma \geq 0.511$, the obtained guarantee $\frac{2\gamma}{\sqrt{5+4\gamma}-1} \geq 0.618$. Hence, for instances with $\gamma \geq 0.511$, our EFx result surpasses the best-known approximation bound of $(\phi - 1) \approx 0.618$ [7, 26].[3] As mentioned previously, a notable fraction of discrete fair division instances submitted at Splitddit.org have such a high γ.

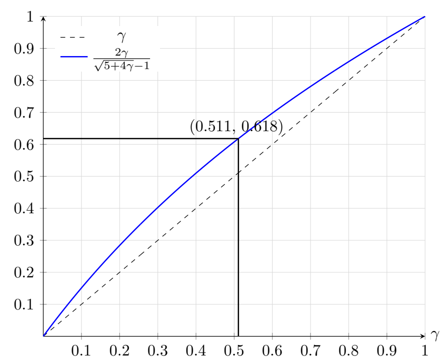

Figure 2: Approximation factor $\frac{2\gamma}{\sqrt{5+4\gamma}-1}$ achieved for EFx.

Our EFx algorithm is based on an adroit extension of the envy-cycle-elimination method of Lipton et al. [36]. We assign goods iteratively while considering them in decreasing order of their base values (formally defined in Section 2). Here, for each agent, the first assigned good is judiciously selected via a look-ahead policy; see Section 3 for details.

Another relevant fairness criterion for indivisible goods is the notion of pairwise maximin shares (PMMS). Specifically, for any agent i, a 1-out-of-2 maximin share from a set S is defined as the maximum value that the agent can guarantee for herself by proposing a partition of S into two subsets—X and $S \backslash X$—and then receiving the minimum valued one between the two. An allocation is said to be PMMS-fair if every agent i receives at least her 1-out-of-2 maximin share when considering—for each other agent j—the set S as the union of the bundles assigned to i and j. Under additive valuations, PMMS is, in fact, a stronger notion than EFx: any PMMS allocation

[3] Recall that ϕ denotes the golden ratio.

is guaranteed to be EFx [18]. As for EFx, the existence of PMMS allocations is an interesting open question [6].

In this paper, we study multiplicative approximations for PMMS. It is known that, under additive valuations, there always exists an allocation in which the PMMS requirement is satisfied between every pair of agents, i and j, within a multiplicative factor of 0.781 [35]. For fair division instances with range parameter $\gamma \in (0, 1]$, the current paper develops a polynomial-time algorithm for finding allocations wherein the PMMS requirement is satisfied between every pair of agents, i and j, within a multiplicative factor of $\frac{5}{6}\gamma$ (Theorem 4.2 in Section 4). This algorithmic result implies the existence of $\frac{5}{6}\gamma$-approximately PMMS allocations for instances with $\gamma \in (0, 1]$. En route to this result, we obtain novel existential and computational guarantees for $\frac{5}{6}$-approximately PMMS allocations under restricted additive valuations.

Additional Related Work. Caragiannis et al. [18] formulated the notion of EFx and showed that, under additive valuations, any (exact) PMMS allocation is also EFx. Gourvès et al. [34] considered a notion similar to EFx in the context of matroids.

Despite marked research efforts, the existence of exact EFx allocations has been established in specific settings: EFx allocations are known to exist when agents have identical valuations [40], when the number of agents is two [40] or three [20], when the number of goods is at most three more than the number of agents [37], or when agents have binary [10], bi-valued [7, 31], or restricted additive [4, 16] valuations.

Towards relaxations of EFx, recent works have also addressed EFx allocations with charity. The objective here is to achieve EFx by assigning a subset of goods and (possibly) leaving some goods unallocated as charity. Chaudhury et al. [22] proved the existence of a partial EFx allocation that leaves at most $(n-1)$ goods unassigned; here n denotes the number of agents in the fair division instance. This bound was later improved to $(n-2)$ goods by Berger et al. [13]. It was also shown in [13] that for four agents, an EFx allocation can always be obtained by discarding at most one good.

Caragiannis et al. [17] focused on the economic efficiency of EFx allocations. They showed the existence of partial EFx allocations with at least half the optimal Nash social welfare. This was further generalized in the work of Feldman et al. [29], establishing the existence of partial allocations that are simultaneously α-EFx and guarantee a $1/(\alpha + 1)$ fraction of the maximum Nash welfare.

Another line of work combines $(1 - \varepsilon)$-approximation guarantees with charity. The work of Chaudhury et al. [21] introduced the notion of rainbow cycle number, $R(d)$, and proved that any appropriate bound on $R(d)$ translates to a sublinear upper bound on the number of unallocated goods. In particular, Chaudhury et al. [21] showed that $R(d) \in O(d^4)$, implying an upper bound of $O_\epsilon(n^{4/5})$ on the number of unassigned goods. This was subsequently improved to $O_\epsilon(n^{2/3})$ [2, 12] and recently to $O_\epsilon(\sqrt{n \log n})$ [1, 19].

2 NOTATION AND PRELIMINARIES

We study the problem of fairly allocating $m \in \mathbb{Z}_+$ indivisible goods among $n \in \mathbb{Z}_+$ agents. The cardinal preferences of the agents $i \in [n]$, over the goods, are specified by valuation functions $v_i : 2^{[m]} \mapsto \mathbb{R}_{\geq 0}$. In particular, $v_i(S)$ denotes the value that agent $i \in [n]$ has for any subset of goods $S \subseteq [m]$. Hence, denoting $[m] = \{1, \ldots, m\}$ as the set of goods and $[n] = \{1, \ldots, n\}$ as the set of agents, a discrete fair division instance is a tuple $\langle [n], [m], \{v_i\}_{i=1}^n \rangle$.

This work considers additive valuations: $v_i(S) = \sum_{g \in S} v_i(\{g\})$, for any agent $i \in [n]$ and all subsets of goods $S \subseteq [m]$. We will write the value of any good $g \in [m]$ for agent $i \in [n]$ as $v_i(g)$ and write P_i to denote the set of goods positively valued by agent i, i.e., $P_i \coloneqq \{g \in [m] \mid v_i(g) > 0\}$. We will assume, without loss of generality that for each agent i, the set P_i is nonempty – for an agent i with $P_i = \emptyset$ every allocation is fair and, hence, such an agent can be disregarded without impacting any of our fair division algorithms. Along similar lines, we will assume, throughout, that for each good g, there exists at least one agent with a nonzero value for g. To simplify notation, for any subset $S \subseteq [m]$ and good $g \in [m]$, we write $S + g$ to denote $S \cup \{g\}$ and $S - g$ to denote $S \setminus \{g\}$.

Range Parameter. Our approximation guarantees—associated with the fairness notions defined below—are in terms of an instance-dependent range parameter $\gamma \in (0, 1]$. This parameter essentially upper bounds, for each good g, the multiplicative range of *nonzero* values for g across the agents. Formally, for instance $\langle [n], [m], \{v_i\}_{i=1}^n \rangle$ and each good $g \in [m]$, we define the range parameter γ_g as ratio of the smallest nonzero and largest value for g,[4] i.e.,

$$\gamma_g \coloneqq \frac{\min_{j : v_j(g) > 0} \ v_j(g)}{\max_i \ v_i(g)} \tag{1}$$

Furthermore, the range parameter for the instance is defined as $\gamma \coloneqq \min_{g \in [m]} \ \gamma_g$.

The parameter γ bounds the range for nonzero values for each good individually: for each good $g \in [m]$ and for any pair of agents $i, j \in [n]$, with a nonzero value for g, we have $v_i(g) \geq \gamma \ v_j(g)$. Therefore, for each good g, we can define a *base value* $\overline{v}(g) \coloneqq \sqrt{\gamma_g} \ \max_{i \in [n]} \ v_i(g)$ and note the following useful property:[5] For every agent $i \in [n]$ and each good $g \in [m]$, either $v_i(g) = 0$ or

$$v_i(g) \in \left[\sqrt{\gamma}\overline{v}(g), \ \frac{1}{\sqrt{\gamma}}\overline{v}(g) \right] \tag{2}$$

Indeed, by the definitions of the base value $\overline{v}(g)$ and γ, we have $\max_{i \in [n]} v_i(g) = \frac{1}{\sqrt{\gamma_g}}\overline{v}(g) \leq \frac{1}{\sqrt{\gamma}}\overline{v}(g)$. Furthermore,

$$\min_{j : v_j(g) > 0} \ v_j(g) = \gamma_g \ \max_{i \in [n]} v_i(g) = \sqrt{\gamma_g} \ \overline{v}(g) \geq \sqrt{\gamma} \ \overline{v}(g).$$

Observe that, when $\gamma = 1$, the value $v_i(g) \in \{0, \overline{v}(g)\}$, for each agent $i \in [n]$. That is, $\gamma = 1$ corresponds to the well-studied class of restricted additive valuations. Also, for any collection of additive valuations, the parameter $\gamma > 0$.

Allocations. An allocation $\mathcal{A} = (A_1, A_2, \ldots, A_n)$ refers to an n-tuple of pairwise disjoint subsets $A_1, A_2, \ldots, A_n \subseteq [m]$ (i.e., $A_i \cap A_j = \emptyset$, for all $i \neq j$). Here, each subset A_i is assigned to agent $i \in [n]$ and is referred to as a bundle. An allocation can be partial, i.e., $\cup_{i=1}^n A_i \subsetneq [m]$. Hence, for disambiguation, we will use the term allocation to denote partitions in which all the goods have been assigned and, otherwise, use the term partial allocation.

Fairness Notions. We next define the fairness constructs considered in this work.

[4] If all the agents have zero value for a good g, i.e., if $\max_i \ v_i(g) = 0$, then, by convention, we set $\gamma_g = 1$.

[5] Equivalently, one can set the base value $\overline{v}(g)$ as the geometric mean of the maximum and the minimum nonzero values for g, i.e., $\overline{v}(g) = \sqrt{\min_{j : v_j(g) > 0} \ v_j(g) \cdot \max_i \ v_i(g)}$.

Definition 2.1 (α-EFx). For parameter $\alpha \in (0,1)$ and fair division instance $\langle [n], [m], \{v_i\}_i \rangle$, an allocation $\mathcal{A} = (A_1, \ldots, A_n)$ is said to be α-approximately envy-free up to the removal of any positively-valued good (α-EFx) iff, for every pair of agents $i, j \in [n]$ and for every good $g \in A_j \cap P_i$ (i.e., for every good in agent j's bundle that is positively valued by i), we have $v_i(A_i) \geq \alpha \, v_i(A_j - g)$.

Setting $\alpha = 1$ in Definition 2.1 gives us the notion of exact EFx allocations. We also address a share-based notion of fairness. In particular, for any agent $i \in [n]$ and subset of goods $S \subseteq [m]$, we define the 1-out-of-2 maximin share, $\mu_i^{(2)}(S)$, as the maximum value that agent i can guarantee for herself by proposing a partition of S into two subsets—X and $S \setminus X$—and then receiving the minimum valued one between the two, $\mu_i^{(2)}(S) \coloneqq \max_{X \subseteq S} \ \min\{v_i(X), v_i(S \setminus X)\}$.

An allocation $(A_1, \ldots, A_n)$ is said to be fair with respect to pairwise maximin shares (PMMS) iff every agent $i \in [n]$ receives at least her 1-out-of-2 maximin share when considering A_i and the bundle of any other agent j, i.e., $v_i(A_i) \geq \mu_i^{(2)}\left(A_i \cup A_j\right)$, for all $j \neq i$. Approximately PMMS-fair allocations are defined as follows.

Definition 2.2 (α-PMMS). For parameter $\alpha \in (0,1)$ and instance $\langle [n], [m], \{v_i\}_i \rangle$, an allocation $\mathcal{A} = (A_1, \ldots, A_n)$ is said to be α-PMMS iff $v_i(A_i) \geq \alpha \, \mu_i^{(2)}(A_i \cup A_j)$, for every pair of agents $i \neq j$.

Remark. *Note that an α-EFx allocation (or an α-PMMS allocation) continues to be α-approximately fair even if one scales the agents' valuations (i.e., sets $v_i(g) \leftarrow s_i \, v_i(g)$, for all agents i and goods g, with agent-specific factors $s_i > 0$). By contrast, such a heterogeneous scaling can change the parameter γ. This, however, is not a limitation, since, for any given instance, we can efficiently find scaling factors that induce the maximum possible range parameter; see full version of the paper [9] for details. One can view this scaling as a preprocessing step which can potentially improve γ.*

3 FINDING APPROXIMATELY EFx ALLOCATIONS

This section develops an algorithm that computes a $\left(\frac{2\gamma}{\sqrt{5+4\gamma}-1}\right)$-EFx allocation in polynomial-time, for any given fair division instance with range parameter $\gamma \in (0,1]$. This algorithmic result also implies the guaranteed existence of such approximately EFx allocations.

Our algorithm is based on an extension of the envy-cycle-elimination method of [36]. Analogous to Lipton et al. [36], we use the construct of an envy-graph (Definition 3.1); however, while considering these graphs we restrict attention to certain subsets of agents. Section 3.1 provides the definitions and results for envy-cycle elimination. Section 3.2 details our algorithm.

We note that, via a somewhat simpler algorithm, one can efficiently find γ-EFx allocations. However, the current work provides a stronger guarantee, since $\frac{2\gamma}{\sqrt{5+4\gamma}-1} > \gamma$, for all $\gamma \in (0,1)$. This strengthening highlights the scope of improvements for the EFx criterion and brings out interesting technical features of the envy-cycle elimination algorithm.

3.1 Envy-Cycle Elimination

Definition 3.1 (Envy graph). For any (partial) allocation $\mathcal{A} = (A_1, \ldots, A_n)$ and any subset of agents $N \subseteq [n]$, the envy graph $\mathcal{G}(N, \mathcal{A})$ is a directed graph on $|N|$ vertices. Here, the vertices represent the agents in N and a directed edge from vertex $i \in N$ to vertex $j \in N$ is included in $\mathcal{G}(N, \mathcal{A})$ iff $v_i(A_i) < v_i(A_j)$.

One can always ensure (by reassigning the bundles among the agents) that the envy-graph is acyclic. In particular, if $\mathcal{G}(N, \mathcal{A})$ contains a cycle $i_1 \to i_2 \to \ldots \to i_k \to i_1$, then we can resolve the cycle by reassigning the bundles as follows: $A_{i_t} = A_{i_{t+1}}$ for all $1 \leq t < k$ and $A_{i_k} = A_{i_1}$. Note that, here, every agent in the cycle receives a higher-valued bundle, and for the agents not in the cycle, the assigned bundles remain unchanged. Hence, such a reassignment of bundles ensures that envy edges either shift or are eliminated, and no new envy edges are formed. Overall, a polynomial number of such reassignments (cycle eliminations) yield an acyclic envy graph. This observation is formally stated in Lemma 3.2.

Lemma 3.2. *Given any partial allocation $\mathcal{A} = (A_1, \ldots, A_n)$ and any subset of agents $N \subseteq [n]$, we can reassign the bundles A_i-s among the agents $i \in N$ and find, in polynomial time, another partial allocation $\mathcal{B} = (B_1, \ldots, B_n)$ with the properties that (i) the envy-graph $\mathcal{G}(N, \mathcal{B})$ is acyclic and (ii) the values $v_i(B_i) \geq v_i(A_i)$, for all agents $i \in [n]$.*

It is relevant to note that any acyclic envy-graph $\mathcal{G}(N, \mathcal{B})$ necessarily admits a source vertex $s \in N$. Such an agent $s \in N$ is not envied by any agent $i \in N$, under the current partial allocation $\mathcal{B}$. In summary, Lemma 3.2 implies that we can always update a partial allocation (by reassigning the bundles) such that the agents' valuations do not decrease and we find an unenvied agent s.

Utilizing this envy-cycle elimination framework, the algorithm of Lipton et al. [36] achieves the fairness notion of envy-freeness up to the removal of one good (EF1). In particular, this EF1 algorithm starts with empty bundles ($A_i = \emptyset$ for all agents $i \in [n]$) and iteratively assigns the goods. During each iteration, the algorithm updates the current partial allocation $\mathcal{A}$ to ensure that the envy-graph $\mathcal{G}([n], \mathcal{A})$ is acyclic and, hence, identifies an unenvied agent $s \in [n]$. To maintain EF1 as an invariant, it suffices to allocate *any* unassigned good to the agent s.

Our algorithm also follows the envy-cycle elimination framework, however it allocates the goods in a judicious order. In particular, for each agent, the first assigned good is intricately selected. The next section details our algorithm and establishes the main EFx result of this work.

3.2 Look-Ahead Assignment

Our algorithm, LaBase (Algorithm 1), starts with empty bundles ($A_i = \emptyset$ for all i), initializes $Z = [n]$ as the set of agents that have zero goods, and then iteratively assigns the goods. In each iteration, LaBase considers an unassigned good with highest base value.[6] That is, the algorithm considers the good $\widehat{g} \in \arg\max_{h \in U} \ \overline{v}(h)$; throughout the algorithm's execution, U denotes the set of (currently) unassigned goods. The algorithm also considers—in relevant cases and for a chosen parameter $\eta \in (0,1]$—a look-ahead set $U_\eta \coloneqq \{g \in U \mid \overline{v}(g) \geq \eta \ \max_{h \in U} \ \overline{v}(h)\}$. Note that U_η consists of

[6] Recall that, for an instance with range parameter γ, the base value of a good g is defined as $\overline{v}(g) \coloneqq \sqrt{\gamma_g} \ \max_i v_i(g)$.

Algorithm 1 LaBase: Look-ahead assignment guided by base values

Input: Instance $\mathcal{I} = \langle [n], [m], \{v_i\}_i \rangle$ with range parameter γ.
Output: A complete allocation $\mathcal{A} = (A_1, \ldots, A_n)$.

1: Initialize bundle $A_i = \emptyset$, for each agent $i \in [n]$. Set parameter $\eta \coloneqq \frac{\sqrt{5+4\gamma}-1}{2} \in (0, 1]$.
2: Also, write $U = [m]$ to denote the set of unassigned goods and $Z = [n]$ to denote the set of agents with an empty bundle.
3: **while** $U \neq \emptyset$ **do**
4: Select good $\widehat{g} \in \arg\max_{h \in U} \overline{v}(h)$.
5: **if** there exists an agent $i \in Z$ such that $v_i(\widehat{g}) > 0$ **then**
6: Define set $U_\eta \coloneqq \{g \in U \mid \overline{v}(g) \geq \eta \ \max_{h \in U} \overline{v}(h)\}$.
7: Select a good $f_i \in \arg\max_{h \in U_\eta} v_i(h)$ and set $A_i = \{f_i\}$. Update $Z \leftarrow Z - i$, and $U \leftarrow U - f_i$.
8: **else if** $v_i(\widehat{g}) = 0$ for all agents $i \in Z$ (or $Z = \emptyset$) **then**
9: Consider the envy-graph $\mathcal{G}([n]\setminus Z, \mathcal{A})$ and resolve envy-cycles in the graph, if any (Lemma 3.2).
10: Set $s \in [n] \setminus Z$ to be a source vertex in $\mathcal{G}([n]\setminus Z, \mathcal{A})$. Update $A_s \leftarrow A_s + \widehat{g}$ and $U \leftarrow U - \widehat{g}$.
11: **end if**
12: **end while**
13: **return** allocation $\mathcal{A} = (A_1, \ldots, A_n)$.

all goods with relatively large base values among the current set of unassigned goods.

The algorithm essentially ensures that the first good assigned to each agent i is of relatively high value under v_i and, at the same time, this good also has a sufficiently high base value. To achieve these two (somewhat complementary) requirements together, the algorithm first checks if there exists an agent i that currently has an empty bundle (i.e., $i \in Z$) and positively values the good $\widehat{g}$ (see Line 5 in Algorithm 1). Such an agent i is allowed to select its most preferred good from the set U_η, i.e., agent i receives as its first good $f_i \in \arg\max_{h \in U_\eta} v_i(h)$; note that f_i is selected considering the agent i's valuation v_i and not the base values. Lemma 3.4 below shows that the two stated (complementary) requirements are satisfied via this first-good selection.

Otherwise, if all the agents with empty bundles (i.e., all agents $i \in Z$) have zero value for the good $\widehat{g}$, then we restrict attention to the remaining agents $[n] \setminus Z$ and assign $\widehat{g}$ following the envy-cycle elimination framework: we ensure that—for the current partial allocation $\mathcal{A}$—the envy-graph $\mathcal{G}([n] \setminus Z, \mathcal{A})$ is acyclic, identify a source $s \in [n] \setminus Z$ in the graph, and assign $\widehat{g}$ to agent s.

One can directly verify that LaBase (Algorithm 1) terminates in polynomial time. The remainder of the section establishes the EFx guarantee achieved by this algorithm. All the missing proofs appear in the full version of the paper [9].

Lemma 3.3. *At the end of any iteration of Algorithm 1, let $\mathcal{A} = (A_1, \ldots, A_n)$ be the maintained partial allocation and let i be any agent contained in the set Z. Then, for all $j \in [n]$, the bundle A_j contains at most one good that is positively valued good by i. That is, the EFx condition holds for $i \in Z$ against all other agents $j \in [n]$.*

The following lemma establishes a key property of the first good assigned to each agent. Recall that, for any given instance with range parameter γ, the algorithm sets $\eta \coloneqq \frac{\sqrt{5+4\gamma}-1}{2}$; this choice of η is guided by an optimization consideration which appears in the analysis below.

Lemma 3.4. *Consider any iteration of the algorithm wherein the if-condition (Line 5) executes, and let $i \in [n]$ be the agent that receives the good f_i in that iteration. Then,*

$$v_i(f_i) \geq \frac{\gamma}{\eta} v_i(g) \qquad \text{for all } g \in U \text{ and}$$
$$\overline{v}(f_i) \geq \eta \ \overline{v}(g) \qquad \text{for all } g \in U.$$

Here, U is the set of unassigned goods during that iteration.

The lemma below complements Lemma 3.4 by addressing the else-if condition in Algorithm 1.

Lemma 3.5. *Consider any iteration of the algorithm in which the else-if condition (Line 8) executes, and let $s \in [n]$ be the agent that receives the good $\widehat{g}$ in that iteration. Then,*

$$\overline{v}(\widehat{g}) \geq \overline{v}(h), \qquad \text{for all } h \in U.$$

Here, U is the set of unassigned goods during that iteration.

Proof. The lemma follows directly from the selection criterion for good $\widehat{g}$ in Line 4 of the algorithm. □

Note that, for each assigned bundle A_s, we can associate an order of inclusion with all the goods in A_s. The bundles are reassigned among the agents (in Line 9), but—bundle wise—the inclusion order remains well defined. In particular, for any bundle A_s and any agent $i \in [n]$, we can index the goods in $A_s \cap P_i = \{g_1, g_2, \ldots, g_k\}$ such that the good g_1 was assigned before g_2 during the algorithm's execution, g_2 before g_3, and so on. The following corollaries consider different values of the count k and establish useful value relations between these goods $g_1, \ldots, g_k$.

Corollary 3.6. *Consider any iteration of the algorithm in which the else-if condition (Line 8) executes, assigning good $\widehat{g}$ to agent $s \in [n]$. Let $\mathcal{A} = (A_1, \ldots, A_n)$ be the partial allocation among the agents at the end of the iteration. Then, for any agent $i \in [n]$ with $|A_s \cap P_i| \leq 1$, the EFx guarantee holds for agent i against s.*

Proof. Since the set $A_s \cap P_i$ contains at most one good, the EFx condition holds for agent i against s (see Definition 2.1). □

Corollary 3.7. *Consider any iteration of the algorithm in which the else-if condition (Line 8) executes, assigning good $\widehat{g}$ to agent $s \in [n]$. Let $\mathcal{A} = (A_1, \ldots, A_n)$ be the partial allocation among the agents at the end of the iteration. Also, let $i \in [n]$ be an agent such that $A_s \cap P_i = \{g_1, \ldots, g_k\}$, for count $k \geq 2$. Then, we have*

$$v_i(g_1) \geq \eta\gamma \ v_i(\widehat{g}) \quad \text{and}$$
$$v_i(g_t) \geq \gamma \ v_i(\widehat{g}) \quad \text{for all indices } t \in \{2, \ldots, k\}.$$

Here, goods g_1 to g_k are indexed in order of inclusion.

We now establish the main result of this section.

Theorem 3.8. *Given any fair division instance $\mathcal{I} = \langle [n], [m], \{v_i\}_i \rangle$ with range parameter $\gamma \in (0, 1]$, Algorithm 1 computes a $\left(\frac{2\gamma}{\sqrt{5+4\gamma}-1}\right)$-EFx allocation in polynomial time.*

Proof. We prove, inductively, that in the algorithm each maintained partial allocation $\mathcal{A}$ is $\left(\frac{2\gamma}{\sqrt{5+4\gamma}-1}\right)$-EFx and, hence, the returned allocation also satisfies the stated guarantee.

The base case of this induction argument holds, since the initial allocation (comprised of empty bundles) is EFx. Now, consider any iteration of the algorithm and let $\mathcal{A}' = (A'_1, \ldots, A'_n)$ denote the partial allocation at the beginning of the considered iteration and $\mathcal{A} = (A_1, \ldots, A_n)$ be the allocation at the end of the iteration. We will assume, via the induction hypothesis, that the partial allocation $\mathcal{A}'$ is $\left(\frac{2\gamma}{\sqrt{5+4\gamma}-1}\right)$-EFx and prove that so is the updated one $\mathcal{A}$.

Towards this, we will analyze the assignment through the if-block (Line 5) and the else-if block (Line 8) separately. That is, we will establish the approximate EFx guarantee for $\mathcal{A}$ considering the following two complementary cases - Case I: The if-condition executes in considered iteration or Case II: The else-if condition executes in considered iteration.

Case I: The if-condition executes. Here, the updated allocation $\mathcal{A} = (A_1, \ldots, A_n)$ is obtained from the starting allocation $\mathcal{A}' = (A'_1, \ldots, A'_n)$ by assigning the good f_i to the selected agent i, who initially has an empty bundle ($A'_i = \emptyset$). Hence, we have $A_j = A'_j$ for all agents $j \neq i$ and $A_i = \{f_i\}$. In this case we show that the approximate EFx guarantee continues to hold under $\mathcal{A}$ between any pair of agents $a, b \in [n]$. Note that, if $a, b \neq i$, then, inductively, the guarantee holds. For $a = i$, note that the value $v_i(A_i) > v_i(A'_i) = 0$ and, hence (as in $\mathcal{A}'$), the approximate EFx guarantee continues to hold for agent $a = i$ against any other agent b (with bundle $A'_b = A_b$). Finally, we consider the approximate EFx guarantee for agents $a \in [n]$ against agent $b = i$: since $|A_i| = 1$, we in fact have EFx against $b = i$. This completes the analysis under Case I.

Case II: The else-if condition executes. Here, note that the approximate EFx guarantee satisfied by the (starting) partial allocation $\mathcal{A}'$ is upheld even after the envy-cycle resolutions in Line 9. That is, the partial allocation obtained after Line 9 (and before the assignment of the good $\widehat{g}$) is also $\left(\frac{2\gamma}{\sqrt{5+4\gamma}-1}\right)$-EFx. This claim follows from Lemma 3.2: in envy-cycle elimination we only reassign the bundles and the valuations of the n agents do not decrease. With a slight abuse of notation, we will reuse $\mathcal{A}' = (A'_1, \ldots, A'_n)$ to denote this resolved partial allocation. In particular, for the $\left(\frac{2\gamma}{\sqrt{5+4\gamma}-1}\right)$-EFx partial allocation $\mathcal{A}'$ the envy-graph $\mathcal{G}([n] \setminus Z, \mathcal{A}')$ is acyclic. Now, the algorithm selects a source s in this graph and assigns good $\widehat{g}$ to agent s. Hence, in the updated allocation $\mathcal{A} = (A_1, \ldots, A_n)$, we have $A_i = A'_i$ for all agents $i \neq s$ and $A_s = A'_s + \widehat{g}$.

In this case as well, we will establish the approximate EFx guarantee under $\mathcal{A}$ for every pair of agents $a, b \in [n]$. First, note that, for any agent a that is contained in set Z at the end of the considered iteration, EFx holds under $\mathcal{A}$ (see Lemma 3.3). Also, if in the considered pair $a, b \in [n]$, the agent $b \neq s$, then the approximate EFx guarantee (against b) carries forward from the allocation $\mathcal{A}'$, since $v_a(A_a) \geq v_a(A'_a)$ and $A_b = A'_b$.

Hence, the remainder of the proof addresses the guarantee for agents $a \in [n] \setminus Z$ against agent $b = s$. Note that if the assigned good $\widehat{g}$ is of zero value for agent $a \in [n] \setminus Z$ (i.e., $v_a(\widehat{g}) = 0$), then (as in the starting allocation $\mathcal{A}'$), the desired approximate EFx guarantee holds for a against $A_s = A'_s + \widehat{g}$. Therefore, for the subsequent analysis we assume that $v_a(\widehat{g}) > 0$. Recall that P_a denotes the set of goods positively valued by agent a, i.e., $P_a = \{g \in [m] \mid v_a(g) > 0\}$. Write count $k' \coloneqq |A'_s \cap P_a|$ and note that $|A_s \cap P_a| = k' + 1$, since $A_s = A'_s + \widehat{g}$ and $\widehat{g} \in P_a$.

Based on the value of the count $k' = |A'_s \cap P_a|$, we have the following sub-cases. In each sub-case, we establish the stated approximate EFx guarantee, under $\mathcal{A}$, for agents $a \in [n] \setminus Z$ and against agent s, thereby completing the proof.

Case II(a): $|A'_s \cap P_a| = 0$. Here, $|A_s \cap P_a| = k' + 1 = 1$ and, hence, Corollary 3.6 (invoked with $i = a$) directly implies that EFx holds for agent a against s in the allocation $\mathcal{A}$.

Case II(b): $|A'_s \cap P_a| = 1$. Write g_1 to the denote the good that constitutes the singleton $|A'_s \cap P_a|$. In addition, since $\widehat{g} \in A_s$ and $v_a(\widehat{g}) > 0$, we have $A_s \cap P_a = \{g_1, \widehat{g}\}$.

Note that, in the current context, s is a source in the envy graph $\mathcal{G}([n] \setminus Z, \mathcal{A}')$ and agent $a \in [n] \setminus Z$. Hence, $v_a(A_a) = v_a(A'_a) \geq v_a(A'_s) = v_a(g_1) = v_a(A_s - \widehat{g})$. This inequality implies envy-freeness for agent a against A_s upon the removal of $\widehat{g}$.

We next address the removal of g_1 from A_s, i.e., we compare $v_a(A_a)$ and $v_a(\widehat{g})$. Since agent $a \in [n] \setminus Z$, this agent must have been assigned a good, f_a, in an earlier iteration. During that iteration, the following containment must have held: $\widehat{g} \in U$; note that the set of unassigned goods is monotonically decreasing. Hence, invoking Lemma 3.4 (with $i = a$), we obtain $v_a(f_a) \geq \frac{\gamma}{\eta}\, v_a(\widehat{g})$. Since the valuation of any agent does not decrease throughout the execution of the algorithm (see, in particular, Lemma 3.2), $v_a(A_a) = v_a(A'_a) \geq v_a(f_a) \geq \frac{\gamma}{\eta}\, v_a(\widehat{g})$.

With $\eta = \frac{\sqrt{5+4\gamma}-1}{2}$, we obtain the desired $\left(\frac{2\gamma}{\sqrt{5+4\gamma}-1}\right)$-EFx guarantee for allocation $\mathcal{A}$ in Case II(b): $v_a(A_a) \geq \frac{\gamma}{\eta} v_a(A_s - g)$ for all $g \in A_s \cap P_a$.

Case II(c): $|A'_s \cap P_a| \geq 2$. Write $g_1, \ldots, g_{k'}$ to denote all the $k' = |A'_s \cap P_a| \geq 2$ goods in the set $A'_s \cap P_a$; these goods are indexed in order of inclusion. Also, note that $A_s \cap P_a = \{g_1, g_2, \ldots, g_{k'}, \widehat{g}\}$. We will show that, in the current sub-case as well, it holds that

$$v_a(A_a) \geq \frac{\gamma}{\eta} v_a(A_s - g) \ \text{ for all } g \in A_s \cap P_a = \{g_1, \ldots, g_{k'}, \widehat{g}\} \quad (3)$$

The desired inequality (3) directly holds for $g = \widehat{g}$: agent $a \in [n] \setminus Z$ and s is a source in the envy-graph $\mathcal{G}([n] \setminus Z, \mathcal{A}')$, hence, we have $v_a(A_a) = v_a(A'_a) \geq v_a(A'_s) = v_a(A_s - \widehat{g})$. Next, we establish the desired inequality (3) for $g = g_1$ and subsequently for $g = g_t$, with indices $t \in \{2, \ldots, k'\}$. Using again the facts that $a \in [n] \setminus Z$ and s is a source in the envy-graph $\mathcal{G}([n] \setminus Z, \mathcal{A}')$, we have

$$\begin{aligned} v_a(A_a) &= v_a(A'_a) \geq v_a(A'_s) = v_a(A'_s - g_1) + v_a(g_1) \\ &\geq v_a(A'_s - g_1) + \eta\gamma\, v_a(\widehat{g}) \\ &\qquad\qquad \text{(applying Corollary 3.7 with } i = a) \\ &= \eta\gamma\, v_a\left(A'_s - g_1 + \widehat{g}\right) \ + \ (1 - \eta\gamma)\, v_a(A'_s - g_1) \\ &= \eta\gamma\, v_a\left(A_s - g_1\right) \ + \ (1 - \eta\gamma)\, v_a(A'_s - g_1) \qquad (4) \end{aligned}$$

The last equality follows from $A_s = A'_s + \widehat{g}$. In the current sub-case, $|A'_s \cap P_a| \geq 2$. Hence, there exists at least one good $g_2 \in (A'_s - g_1)$. For g_2, Corollary 3.7 (applied with $i = a$) gives us $v_a(g_2) \geq \gamma v_a(\widehat{g})$, i.e., we obtain $v_a(A'_s - g_1) \geq \gamma\, v_a(\widehat{g})$. Adding $\gamma\, v_a(A'_s - g_1)$ to both sides of the last inequality gives us

$$\begin{aligned}(1+\gamma)\, v_a(A'_s - g_1) &\geq \gamma \left(v_a(A'_s - g_1) + v_a(\widehat{g})\right)\\ &= \gamma\, v_a(A'_s + \widehat{g} - g_1) = \gamma\, v_a(A_s - g_1) \qquad (5)\end{aligned}$$

Equation (5) reduces to $v_a(A'_s - g_1) \geq \frac{\gamma}{1+\gamma} v_a(A_s - g_1)$. The last inequality and (4) give us

$$\begin{aligned}v_a(A_a) &\geq \eta\gamma\, v_a\left(A_s - g_1\right) + (1-\eta\gamma)\,\frac{\gamma}{1+\gamma} v_a\left(A_s - g_1\right)\\ &= \left(\eta\gamma + (1-\eta\gamma)\,\frac{\gamma}{1+\gamma}\right) v_a\left(A_s - g_1\right)\\ &= \left(\frac{\gamma(1+\eta)}{1+\gamma}\right) v_a(A_s - g_1). \qquad (6)\end{aligned}$$

For parameter $\eta = \frac{\sqrt{5+4\gamma}-1}{2}$, it holds that $\left(\frac{\gamma(1+\eta)}{1+\gamma}\right) = \frac{\gamma}{\eta}$; in fact, η is specifically chosen to satisfy this equality.[7] Hence, equation (6) leads to the desired inequality (3) for $g = g_1$.

Finally, to establish inequality (3) for $g = g_t$, with index $t \in \{2, \ldots, k'\}$, we start with

$$\begin{aligned}v_a(A_a) = v_a(A'_a) &\geq v_a(A'_s) = v_a(A'_s - g_t) + v_a(g_t)\\ &\geq v_a(A'_s - g_t) + \gamma v_a(\widehat{g})\\ &\qquad\text{(applying Corollary 3.7 with } i = a\text{)}\\ &= \gamma\, v_a\left(A'_s - g_t + \widehat{g}\right) + (1-\gamma)\, v_a(A'_s - g_t)\\ &= \gamma\, v_a\left(A_s - g_t\right) + (1-\gamma)\, v_a(A'_s - g_t) \qquad (7)\end{aligned}$$

Since $g_1 \in (A'_s - g_t) \cap P_a$, using Corollary 3.7 (for g_1 and $i = a$) we obtain $v_a(A'_s - g_t) \geq v_a(g_1) \geq \eta\gamma\, v_a(\widehat{g})$. The last inequality is equivalent to $v_a(A'_s - g_t) \geq \frac{\eta\gamma}{1+\eta\gamma} v_a(A'_s - g_t + \widehat{g}) = \frac{\eta\gamma}{1+\eta\gamma} v_a(A_s - g_t)$. Using the last inequality and equation (7), we obtain

$$\begin{aligned}v_a(A_a) &\geq \gamma\, v_a\left(A_s - g_t\right) + (1-\gamma)\,\frac{\eta\gamma}{1+\eta\gamma} v_a(A_s - g_t)\\ &= \left(\gamma + (1-\gamma)\,\frac{\eta\gamma}{1+\eta\gamma}\right) v_a(A_s - g_t)\\ &= \frac{\gamma(1+\eta)}{1+\eta\gamma} v_a(A_s - g_t) \qquad (8)\end{aligned}$$

Since $\eta \leq 1$, the approximation factor $\frac{\gamma(1+\eta)}{1+\eta\gamma} \geq \frac{\gamma(1+\eta)}{1+\gamma} = \frac{\gamma}{\eta}$; recall that η is specifically chosen to satisfy the last equality. Therefore, from (8), we obtain the desired inequality (3) for $g = g_t$, with indices $t \in \{2, \ldots, k'\}$. This completes the analysis for Case II(c).

Overall, in each case, the allocation $\mathcal{A}$ (maintained by the algorithm at the end of the considered iteration) upholds the EFx guarantee with approximation factor $\frac{\gamma}{\eta} = \left(\frac{2\gamma}{\sqrt{5+4\gamma}-1}\right)$. This completes the inductive argument and establishes the theorem. □

[7] Equivalently, η is set as the positive root of the quadratic equation $\eta^2 + \eta = 1 + \gamma$.

4 FINDING APPROXIMATELY PMMS ALLOCATIONS

This section develops a polynomial time algorithm for finding $\frac{5}{6}\gamma$-PMMS allocations for fair division instances $\mathcal{I}$ with range parameter $\gamma \in (0, 1]$.

To obtain this result we first establish a reduction from the given instance $\mathcal{I}$ to an instance $\widetilde{\mathcal{I}}$ with range parameter 1 (Theorem 4.1). The reduction is approximation preserving up to a factor of γ: any α-PMMS allocation (bearing a relevant property) in $\widetilde{\mathcal{I}}$ is guaranteed to be an $\alpha\gamma$-PMMS allocation in $\mathcal{I}$.

With this reduction in hand, we proceed to develop (in Section 5) a polynomial time algorithm that computes a $\frac{5}{6}$-PMMS allocation (with the relevant property) for instances with range parameter 1. This algorithmic result and the reduction give us the stated $\frac{5}{6}\gamma$-PMMS approximation guarantee (Theorem 4.2).

Theorem 4.1. *Given any fair division instance $\mathcal{I} = \langle [n], [m], \{v_i\}_i \rangle$ with range parameter γ, we can efficiently construct another instance $\widetilde{\mathcal{I}} = \langle [n], [m], \{\widetilde{v}_i\}_i \rangle$ with range parameter 1 such that*

- *For each agent $i \in [n]$, the subset of positively valued goods, P_i, remains unchanged, and*
- *Any α-PMMS allocation $\mathcal{A} = (A_1, \ldots, A_n)$ in $\widetilde{\mathcal{I}}$ with the property that $A_i \subseteq P_i$, for all $i \in [n]$, is an $\alpha\gamma$-PMMS allocation in $\mathcal{I}$.*

A complete proof of Theorem 4.1 appears in the full version [9]; we provide a proof sketch here.

Proof Sketch. To construct instance $\widetilde{\mathcal{I}} = \langle [n], [m], \{\widetilde{v}_i\}_i \rangle$ from the given instance $\mathcal{I}$, we set the agents' additive valuations $\widetilde{v}_i$ as follows: for each agent i and good g, if $v_i(g) = 0$, then we set $\widetilde{v}_i(g) = 0$. Otherwise, if $v_i(g) > 0$ (i.e., $g \in P_i$), then set $\widetilde{v}_i(g)$ as the base value of the good, $\widetilde{v}_i(g) = \overline{v}(g)$. By construction, the range parameter is equal to 1 in $\widetilde{\mathcal{I}}$.

Recall that $\mu_i^{(2)}(S)$ denotes the pairwise maximin share of agent i, for subset of goods $S \subseteq [m]$, in $\mathcal{I}$. Also, write $\widetilde{\mu}_i^{(2)}(S)$ to denote the pairwise maximin share in $\widetilde{\mathcal{I}}$. One can show that, for any $S \subseteq [m]$, it holds that $\sqrt{\gamma}\, \mu_i^{(2)}(S) \leq \widetilde{\mu}_i^{(2)}(S)$.

Now, consider any α-PMMS allocation $\mathcal{A} = (A_1, \ldots, A_n)$ in $\widetilde{\mathcal{I}}$ with the property that $A_i \subseteq P_i$, for all $i \in [n]$. Under $\mathcal{A}$ and for all agents i, j, we have $\widetilde{v}_i(A_i) \geq \alpha\widetilde{\mu}_i^{(2)}(A_i \cup A_j)$. Using this inequality and the bound between the pairwise maximin shares, it can be shown that $v_i(A_i) \geq \alpha\gamma\, \mu_i^{(2)}(A_i \cup A_j)$ for all agents $i, j \in [n]$. Therefore, the stated approximate PMMS guarantee holds for allocation $\mathcal{A}$ in instance $\mathcal{I}$. This completes the proof sketch.

With Theorem 4.1 in hand, we focus our attention on instances with $\gamma = 1$. For such instances, in Section 5 we develop a polynomial-time algorithm for computing $\frac{5}{6}$-PMMS allocations. The algorithm developed in Section 5 in fact finds a $\frac{5}{6}$-PMMS allocation $(A_1, \ldots, A_n)$ in which for each agent i we have $A_i \subseteq P_i$ (see Lemma 5.2 in Section 5). Hence, using Theorem 4.1 and Theorem 5.1, we obtain the main result for finding approximately PMMS allocations:

Theorem 4.2. *For any given fair division instance $\mathcal{I} = \langle [n], [m], \{v_i\}_i \rangle$, with range parameter $\gamma \in (0, 1]$, we can compute a $\frac{5}{6}\gamma$-PMMS allocation in polynomial time.*

5 APPROXIMATE PMMS WITH RANGE PARAMETER 1

This section develops a polynomial-time algorithm (Algorithm 2) that computes $\frac{5}{6}$-PMMS allocations for instances with range parameter 1. The following theorem is the main result of this section.

Theorem 5.1. *For any given instance $\widetilde{\mathcal{I}} = \langle [n], [m], \{\widetilde{v}_i\}_i \rangle$, with range parameter 1, we can compute a $\frac{5}{6}$-PMMS allocation in polynomial time.*

Throughout this section, we denote the given instance (with range parameter 1) as $\widetilde{\mathcal{I}} = \langle [n], [m], \{\widetilde{v}_i\}_i \rangle$. Here, the pairwise maximin share of agent i, for any subset of goods $S \subseteq [m]$, will be denoted as $\widetilde{\mu}_i^{(2)}(S) \coloneqq \max_{Y \subseteq S} \ \min\{\widetilde{v}_i(Y), \widetilde{v}_i(S \setminus Y)\}$. Also, write $\overline{P}(g) \coloneqq \{i \in [n] \mid \widetilde{v}_i(g) > 0\}$ as the set of agents that positively value good $g \in [m]$. Note that, for instance $\widetilde{\mathcal{I}}$, with range parameter equal to one, the base value of each good g satisfies $\overline{v}(g) = \widetilde{v}_i(g)$ for all $i \in \overline{P}(g)$. Also, recall that P_i denotes the set of goods positively valued by i, i.e., $P_i = \{g \in [m] \mid \widetilde{v}_i(g) > 0\}$.

Our PMMS algorithm (Algorithm 2) starts with empty bundles for the agents and assigns the goods iteratively. In each iteration, the algorithm selects, among the currently unassigned goods U, one with the maximum base value, i.e., it selects $\widehat{g} \in \arg\max_{h \in U} \overline{v}(h)$. The algorithm then considers the envy-graph restricted to the agents who positively value $\widehat{g}$ and the partial allocation $\mathcal{A}$ at the start of the iteration. In particular, we consider the directed graph $\widetilde{\mathcal{G}}(\overline{P}(\widehat{g}), \mathcal{A})$ wherein a directed edge exists from vertex $i \in \overline{P}(\widehat{g})$ to vertex $j \in \overline{P}(\widehat{g})$ iff $\widetilde{v}_i(A_i) < \widetilde{v}_i(A_j)$.

We will show that the graph $\widetilde{\mathcal{G}}(\overline{P}(\widehat{g}), \mathcal{A})$ is always acyclic (Lemma 5.2). That is, we can always find a source s in the graph $\widetilde{\mathcal{G}}(\overline{P}(\widehat{g}), \mathcal{A})$, without having to resolve envy cycles in it. The algorithm assigns the good $\widehat{g}$ to the source agent $s \in \overline{P}(\widehat{g})$. Note that restricting the envy graph to $\overline{P}(\widehat{g})$ ensures that $\widehat{g}$ is only assigned to an agent who positively values it.[8]

Algorithm 2 Allocation in decreasing order of base values

Input: Instance $\widetilde{\mathcal{I}} = \langle [n], [m], \{\widetilde{v}_i\}_i \rangle$ with range parameter 1.
Output: A complete allocation $\mathcal{A} = (A_1, \ldots, A_n)$.

1: Initialize $A_i = \emptyset$, for each agent $i \in [n]$, and write $U \coloneqq [m]$.
2: **while** $U \neq \emptyset$ **do**
3: Select an unassigned good $\widehat{g} \in \arg\max_{h \in U} \overline{v}(h)$, and consider the envy graph $\widetilde{\mathcal{G}}(\overline{P}(\widehat{g}), \mathcal{A})$.
4: Set $s \in \overline{P}(\widehat{g})$ to be a source vertex in $\widetilde{\mathcal{G}}(\overline{P}(\widehat{g}), \mathcal{A})$.
{$\widetilde{\mathcal{G}}(\overline{P}(\widehat{g}), \mathcal{A})$ is guaranteed to be acyclic (Lemma 5.2).}
5: Update $A_s \leftarrow A_s + \widehat{g}$ and $U \leftarrow U - \widehat{g}$.
6: **end while**
7: **return** Allocation $\mathcal{A} = (A_1, \ldots, A_n)$.

Lemma 5.2. *Let $\widetilde{\mathcal{G}}(\overline{P}(\widehat{g}), \mathcal{A})$ be any envy-graph considered during the execution of Algorithm 2 (in Line 3). Then, $\widetilde{\mathcal{G}}(\overline{P}(\widehat{g}), \mathcal{A})$ is acyclic. In addition, for any partial allocation $\mathcal{A} = (A_1, \ldots, A_n)$ considered in the algorithm, we have $A_i \subseteq P_i$, for all agents $i \in [n]$.*

[8]This design decision helps us in establishing the property $A_i \subseteq P_i$, which is required in Theorem 4.1.

Lemma 5.2 ensures that Algorithm 2 executes successfully. Furthermore, one can directly verify that the algorithm terminates in polynomial time; the missing proofs appear in the full version [9]. Hence, the remainder of the section addresses the approximate PMMS guarantee achieved by Algorithm 2. Towards this, we provide two supporting results (Lemmas 5.3 and 5.4). The following lemma shows that, in each iteration of the algorithm, the approximate PMMS guarantee is preserved for the agent receiving the good.

Lemma 5.3. *Consider any iteration of Algorithm 2 wherein agent s receives the good $\widehat{g}$. Let $\mathcal{A}' = (A'_1, \ldots, A'_n)$ denote the partial allocation maintained by the algorithm at the beginning of the iteration and $\mathcal{A} = (A_1, \ldots, A_n)$ be the one at the end. If α-PMMS guarantee held under allocation $\mathcal{A}'$ for agent s, then the guarantee also holds for s under $\mathcal{A}$. That is, if $\widetilde{v}_s(A'_s) \geq \alpha \widetilde{\mu}_s^{(2)}(A'_s + A'_j)$, for all $j \in [n]$, then $\widetilde{v}_s(A_s) \geq \alpha \widetilde{\mu}_s^{(2)}(A_s + A_j)$.*

We next show that, for each agent i and among the positively valued goods P_i, the goods are assigned in decreasing order of $\widetilde{v}_i(\cdot)$.

Lemma 5.4. *Consider any iteration of Algorithm 2, wherein agent s receives the good $\widehat{g}$, and let $\mathcal{A} = (A_1, \ldots, A_n)$ denote the partial allocation maintained by the algorithm at the end of the iteration. Then, for each agent $i \in [n]$, good $\widehat{g}$ is the least valued good in the set $A_i \cup (A_s \cap P_i)$, i.e., $\widetilde{v}_i(g) \geq \widetilde{v}_i(\widehat{g})$ for all goods $g \in A_i \cup (A_s \cap P_i)$.*

Theorem 5.1 follows from the above-mentioned lemmas and via an involved case analysis. A complete proof of this theorem is given in the full version of the paper [9].

6 CONCLUSION AND FUTURE WORK

The current work establishes approximation guarantees for EFx and PMMS criteria in terms of range parameter γ of fair division instances. In particular, we develop an efficient algorithm for computing $\left(\frac{2\gamma}{\sqrt{5+4\gamma}-1}\right)$-EFx allocations. The work also proves that $\frac{5}{6}\gamma$-PMMS allocations can be computed in polynomial time.

For these approximation factors, improving the dependence on γ is a relevant direction for future work. Indeed, this research direction provides a quantitative route to progress toward the EFx conjecture. In particular, it would be interesting to establish the existence of exact EFx allocation for instances in which, say, γ is at least $1/2$. Another interesting specific question here is to study the existence of exact maximin share (MMS) allocations for restricted additive valuations. Notably, recent (non)examples, which show that MMS allocations do not always exist, have $\gamma = 0.9$ [28]. In this range-parameter framework, studying the tradeoffs between fairness and welfare is another relevant direction of future work.

ACKNOWLEDGMENTS

Siddharth Barman gratefully acknowledges the support of a SERB Core research grant (CRG/2021/006165) and a Walmart (WMGTS) CSR Grant. Debajyoti Kar was generously supported by a Google PhD Fellowship.

REFERENCES

[1] Hannaneh Akrami, Noga Alon, Bhaskar Ray Chaudhury, Jugal Garg, Kurt Mehlhorn, and Ruta Mehta. 2023. EFX: A Simpler Approach and an (Almost) Optimal Guarantee via Rainbow Cycle Number. In *24th ACM Conference on Economics and Computation, EC 2023*. Association for Computing Machinery, Inc, 61.

[2] Hannaneh Akrami, Bhaskar Ray Chaudhury, Jugal Garg, Kurt Mehlhorn, and Ruta Mehta. 2022. EFX allocations: Simplifications and improvements. *arXiv preprint arXiv:2205.07638* (2022).

[3] Hannaneh Akrami, Bhaskar Ray Chaudhury, Martin Hoefer, Kurt Mehlhorn, Marco Schmalhofer, Golnoosh Shahkarami, Giovanna Varricchio, Quentin Vermande, and Ernest van Wijland. 2022. Maximizing Nash social welfare in 2-value instances. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 4760–4767.

[4] Hannaneh Akrami, Rojin Rezvan, and Masoud Seddighin. 2022. An EF2X Allocation Protocol for Restricted Additive Valuations. In *Thirty-First International Joint Conference on Artificial Intelligence*. IJCAI, 17–23.

[5] Martin Aleksandrov, Haris Aziz, Serge Gaspers, and Toby Walsh. 2015. Online fair division: analysing a food bank problem. In *Proceedings of the 24th International Conference on Artificial Intelligence*. 2540–2546.

[6] Georgios Amanatidis, Haris Aziz, Georgios Birmpas, Aris Filos-Ratsikas, Bo Li, Hervé Moulin, Alexandros A Voudouris, and Xiaowei Wu. 2022. Fair division of indivisible goods: A survey. *arXiv preprint arXiv:2208.08782* (2022).

[7] Georgios Amanatidis, Georgios Birmpas, Aris Filos-Ratsikas, Alexandros Hollender, and Alexandros A Voudouris. 2021. Maximum Nash welfare and other stories about EFX. *Theoretical Computer Science* 863 (2021), 69–85.

[8] Georgios Amanatidis, Evangelos Markakis, and Apostolos Ntokos. 2020. Multiple birds with one stone: Beating 1/2 for EFX and GMMS via envy cycle elimination. *Theoretical Computer Science* 841 (2020), 94–109.

[9] Siddharth Barman, Debajyoti Kar, and Shraddha Pathak. 2023. Parameterized Guarantees for Almost Envy-Free Allocations. *arXiv preprint arXiv:2312.13791* (2023).

[10] Siddharth Barman, Sanath Kumar Krishnamurthy, and Rohit Vaish. 2018. Greedy Algorithms for Maximizing Nash Social Welfare. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. 7–13.

[11] Nawal Benabbou, Mithun Chakraborty, Ayumi Igarashi, and Yair Zick. 2020. Finding fair and efficient allocations when valuations don't add up. In *International Symposium on Algorithmic Game Theory*. Springer, 32–46.

[12] Benjamin Aram Berendsohn, Simona Boyadzhiyska, and László Kozma. 2022. Fixed-point cycles and approximate EFX allocations. In *47th International Symposium on Mathematical Foundations of Computer Science (MFCS 2022)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

[13] Ben Berger, Avi Cohen, Michal Feldman, and Amos Fiat. 2022. Almost full EFX exists for four agents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 4826–4833.

[14] Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia. 2016. *Handbook of Computational Social Choice* (1st ed.). Cambridge University Press, USA.

[15] Eric Budish, Gérard P. Cachon, Judd B. Kessler, and Abraham Othman. 2017. Course Match: A Large-Scale Implementation of Approximate Competitive Equilibrium from Equal Incomes for Combinatorial Allocation. *Oper. Res.* 65 (2017), 314–336.

[16] Franklin Camacho, Rigoberto Fonseca-Delgado, Ramón Pino Pérez, and Guido Tapia. 2021. Beyond identical utilities: buyer utility functions and fair allocations. *arXiv:2109.08461* (2021).

[17] Ioannis Caragiannis, Nick Gravin, and Xin Huang. 2019. Envy-freeness up to any item with high Nash welfare: The virtue of donating items. In *Proceedings of the 2019 ACM Conference on Economics and Computation*. 527–545.

[18] Ioannis Caragiannis, David Kurokawa, Hervé Moulin, Ariel D Procaccia, Nisarg Shah, and Junxing Wang. 2019. The unreasonable fairness of maximum Nash welfare. *ACM Transactions on Economics and Computation (TEAC)* 7, 3 (2019), 1–32.

[19] Shayan Chashm Jahan, Masoud Seddighin, Seyed-Mohammad Seyed-Javadi, and Mohammad Sharifi. 2022. Rainbow Cycle Number and EFX Allocations:(Almost) Closing the Gap. *arXiv e-prints* (2022), arXiv–2212.

[20] Bhaskar Ray Chaudhury, Jugal Garg, and Kurt Mehlhorn. 2020. EFX exists for three agents. In *Proceedings of the 21st ACM Conference on Economics and Computation*. 1–19.

[21] Bhaskar Ray Chaudhury, Jugal Garg, Kurt Mehlhorn, Ruta Mehta, and Pranabendu Misra. 2021. Improving EFX guarantees through rainbow cycle number. In *Proceedings of the 22nd ACM Conference on Economics and Computation*. 310–311.

[22] Bhaskar Ray Chaudhury, Telikepalli Kavitha, Kurt Mehlhorn, and Alkmini Sgouritsa. 2021. A little charity guarantees almost envy-freeness. *SIAM J. Comput.* 50, 4 (2021), 1336–1358.

[23] Yongheng Deng, Tien Foo Sing, and Chaoqun Ren. 2013. The story of Singapore's public housing: From a nation of home-seekers to a nation of homeowners. In *The future of public housing*. Springer, 103–121.

[24] Lester E Dubins and Edwin H Spanier. 1961. How to cut a cake fairly. *The American Mathematical Monthly* 68, 1P1 (1961), 1–17.

[25] Soroush Ebadian, Dominik Peters, and Nisarg Shah. 2022. How to Fairly Allocate Easy and Difficult Chores. In *21st International Conference on Autonomous Agents and Multiagent Systems*.

[26] Alireza Farhadi, MohammadTaghi Hajiaghayi, Mohamad Latifian, Masoud Seddighin, and Hadi Yami. 2021. Almost envy-freeness, envy-rank, and nash social welfare matchings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 5355–5362.

[27] Uriel Feige. 2022. Maximin fair allocations with two item values. *Unpublished Manuscript* (2022).

[28] Uriel Feige, Ariel Sapir, and Laliv Tauber. 2021. A tight negative example for MMS fair allocations. In *International Conference on Web and Internet Economics*. Springer, 355–372.

[29] Michal Feldman, Simon Mauras, and Tomasz Ponitka. 2023. On optimal tradeoffs between EFX and nash welfare. *arXiv preprint arXiv:2302.09633* (2023).

[30] Duncan Karl Foley. 1966. *Resource allocation and the public sector.* Yale University.

[31] Jugal Garg and Aniket Murhekar. 2023. Computing fair and efficient allocations with few utility values. *Theoretical Computer Science* 962 (2023), 113932.

[32] Jugal Garg, Aniket Murhekar, and John Qin. 2022. Fair and efficient allocations of chores under bivalued preferences. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 5043–5050.

[33] Jonathan Goldman and Ariel D Procaccia. 2015. Spliddit: Unleashing fair division algorithms. *ACM SIGecom Exchanges* 13, 2 (2015), 41–46.

[34] Laurent Gourvès, Jérôme Monnot, and Lydia Tlilane. 2014. Near Fairness in Matroids.. In *ECAI*, Vol. 14. 393–398.

[35] David Kurokawa. 2017. Algorithms for Fair Division. *PhD Thesis, Carnegie Mellon University* (2017).

[36] Richard J Lipton, Evangelos Markakis, Elchanan Mossel, and Amin Saberi. 2004. On approximately fair allocations of indivisible goods. In *Proceedings of the 5th ACM Conference on Electronic Commerce*. 125–131.

[37] Ryoga Mahara. 2023. Extension of additive valuations to general valuations on the existence of EFX. *Mathematics of Operations Research* (2023).

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

[39] Hervé Moulin. 2019. Fair division in the internet age. *Annual Review of Economics* 11 (2019), 407–441.

[40] Benjamin Plaut and Tim Roughgarden. 2020. Almost envy-freeness with general valuations. *SIAM Journal on Discrete Mathematics* 34, 2 (2020), 1039–1068.

[41] Canice Prendergast. 2017. How food banks use markets to feed the poor. *Journal of Economic Perspectives* 31, 4 (2017), 145–162.

[42] Ariel D Procaccia. 2020. Technical perspective: An answer to fair division's most enigmatic question. *Commun. ACM* 63, 4 (2020), 118–118.

[43] Nisarg Shah. 2017. Spliddit: two years of making the world fairer. *XRDS: Crossroads, The ACM Magazine for Students* 24, 1 (2017), 24–28.

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