(Almost) Envy-Free, Proportional and Efficient Allocations of an Indivisible Mixed Manna[*]

Vasilis Livanos[†]
livanos3@illinois.edu

Ruta Mehta[‡]
rutameht@illinois.edu

Aniket Murhekar[§]
aniket2@illinois.edu

Abstract

We study the problem of finding fair and efficient allocations of a set of indivisible items to a set of agents, where each item may be a good (positively valued) for some agents and a bad (negatively valued) for others, i.e., a mixed manna. As fairness notions, we consider arguably the strongest possible relaxations of envy-freeness and proportionality, namely envy-free up to any item (EFX and EFX_0), and proportional up to the maximin good or any bad (PropMX and $PropMX_0$). Our efficiency notion is Pareto-optimality (PO).

We study two types of instances: (i) *Separable*, where the item set can be partitioned into goods and bads, and (ii) *Restricted mixed goods (RMG)*, where for each item j, every agent has either a non-positive value for j, or values j at the same $v_j > 0$. We obtain polynomial-time algorithms for the following:

- Separable instances: $PropMX_0$ allocation.

- RMG instances: Let *pure bads* be the set of items that everyone values negatively.
    - PropMX allocation for general pure bads.
    - EFX+PropMX allocation for identically-ordered pure bads.
    - EFX+PropMX+PO allocation for identical pure bads.

Finally, if the RMG instances are further restricted to *binary mixed goods* where all the v_j's are the same, we strengthen the results to guarantee EFX_0 and $PropMX_0$ respectively.

1 Introduction

The problem of fair division is concerned with allocating items to agents in a *fair* and *efficient* manner. Formally introduced by Steinhaus [30], fair division is an active area of research studied across fields like computer science and economics. Most work has focused on the fair division of *goods*: items which provide non-negative *value* (or utility) to the agents to whom they are allocated. However, several practical scenarios involve *bads* (or chores). Bads are items which impose a *cost* (or disutility) to the agent to whom they are allocated. Generalizing both settings, we study fair division of a set M of indivisible items, where each $j \in M$ can be a good for some agents and a bad for others – a *mixed manna*. Examples of mixed manna include splitting assets and liabilities when dissolving a partnership, dividing tasks among various team members, and deciding teaching assignments between faculty. The valuation of agent i for a set $S \subseteq M$ of items is defined by an

[†]University of Illinois at Urbana-Champaign, USA
[‡]University of Illinois at Urbana-Champaign, USA
[§]University of Illinois at Urbana-Champaign, USA

additive function $v_i(S) = \sum_{j \in S} v_{ij}$ where $v_{ij} \in \mathbb{R}$ is the value agent i has for item j. We say that item j is a good for agent i if $v_{ij} \geq 0$, otherwise it is a bad for her.

Arguably, the two most popular fairness notions are of envy-freeness (EF) [17, 31] and proportionality (Prop) [30]. EF requires that every agent (weakly) prefers her own allocation than anyone else's, while Prop requires that every agent gets her proportional value, i.e., $\frac{1}{n}$-fraction of her value for all the items. When items are *divisible*, both EF and Prop allocations are known to exist. However, in the case of *indivisible* items, neither may exist for very simple examples, like allocating one good among two agents who value it equally.

One of the strongest relaxations of EF for indivisible items is *EF up to any item (EFX)* [14]. That is, for goods manna, every agent i does not envy any other agent k after removal of a (positively-valued) good from k's bundle, and for bads manna the envy vanishes after removal of any bad from i's bundle. Generalizing these to mixed manna, we say that an allocation is EFX if each agent i does not envy another agent k after either removal of a positively-valued good from k's bundle or removal of a bad from i's bundle.

The analogous relaxation for proportionality is the notion of *proportionality up to any item* (PropX), defined by Aziz et al. [5]. For goods, an allocation is said to be PropX if every agent can receive her proportional share after the addition to her bundle of any one good not in her bundle. Similarly, for bads, an allocation is said to be PropX if every agent can receive her proportional share after removing any one bad assigned to her. While PropX need not exist for goods [26, 5], Li et al. [24] showed that a PropX allocation always exists for bads and can be computed in polynomial time. Since PropX need not exist for goods, Baklanov et al. [6] introduce the notion of *proportionality up to the maximin good (PropM)*, a weaker version of PropX. They show that PropM allocations always exist for the setting of goods and agents with additive valuations, and that they can be computed in polynomial time.

Relaxations of Prop have not been studied previously for indivisible mixed manna. Combining the strongest possible guarantees, we define the following relaxation for mixed manna: *proportionality up to the maximin good or any bad (PropMX)*. Informally, an allocation is said to be PropMX if every agent can receive her proportional share after the addition to her bundle of the *maximin* positively-valued good for that agent, or after the removal of any one bad assigned to her.

In addition to being fair, it is important for allocations to be *efficient*. The standard notion of economic efficiency is Pareto optimality (PO). An allocation is said to be PO if no other allocation makes an agent better off without making someone else worse off.

In this paper we ask if it possible to achieve all of EFX, PropMX, and PO in a single allocation. And if yes, then can it be computed in polynomial time. The answer is not known even for goods (bads) only manna. In particular, existence of an EFX allocation is a celebrated open question even for goods (bads) manna [29]. And therefore the focus has been on resolving special cases (see Section 1.2 for a detailed discussion). Furthermore, it has been noted that mixed manna is significantly harder to handle than the goods (bads) manna [12, 23]. Next we describe our contributions in this context.

1.1 Our Contributions.

We study the problem of computing EFX+PO and PropMX+PO allocations for mixed manna instances. Given that computing EFX (and EFX+PO) allocations is a challenging problem even for general instances of goods, a significant amount of past works have focused on sub-classes by restricting the values that agents have for the items, see [3, 27, 10, 19, 1, 2] and references therein for practical scenarios involving different sub-classes. Among these, the valuation classes of identical [28, 10, 1, 2], identical order preferences (IDO) [28, 24], restricted additive [7], and binary [10, 21, 11]

are well-studied for goods (bads) manna. We study all of these for mixed manna.

First, to extend these to mixed manna we need to partition the items into three sets: the set M^+ of *mixed goods*, which are valued positively by at least one agent; the set M^0 of *dummy bads*, which are not valued positively by any agent but may be valued at zero by some; and set M^- of *pure bads* which are valued negatively by all agents.

We consider the following:

- *Separable* instances: For every item j, all the agents value it either non-negatively or negatively. That is the item set can be partitioned in to goods (non-negatively valued) and bads (negatively valued).

- *Restricted Mixed goods* instances: For every item $j \in M^+$ there exists a value $v_j > 0$ such that if an agent values j positively, then she values it at v_j. Furthermore, if $v_j = v_{j'}$ for all $j, j' \in M^+$, then the instance is called a *binary mixed goods* instance.

Note that the separable and RMG instances are incomparable. We obtain following results for these settings.

- For **Restricted Mixed Goods**, we give polynomial-time algorithms for computing allocation that is

  - PropMX+EFX+PO when agents have identical valuation over the pure bads M^- (Theorem 2).
  - PropMX+EFX when agents have identical ordinal preference (IDO) over the pure bads (Theorem 5).
  - PropMX for the case of **general pure bads** (Theorem 6).

- For the special case of **binary mixed goods**, we strengthen the previous results to EFX_0 and $PropMX_0$ (Theorem 3) respectively.[1]

- In showing the above results, we show that an EFX+PO allocation can be computed in polynomial-time for a **goods** only manna with **restricted valuations** (Theorem 1). This class is orthogonal to the class of IDO goods, for which EFX allocations are known to be efficiently computable [28].

- For **separable** instances, we present an algorithm which returns, in polynomial time, an allocation that is $PropMX_0$ (Theorem 4).

We note that as a corollary we obtain some of the results of [1] and [2] for the binary and identical mixed manna. We observe that EFX (resp. EFX_0) implies PropMX (resp. $PropMX_0$), and therefore whenever we get EFX we get PropMX, but not vice-versa. Furthermore, we show via a counterexample that one cannot hope to obtain a $PropMX_0$+PO allocation, even for the goods manna (Appendix A.2), and hence the first two results for the RMG instances can not be strengthened.

The subclasses we consider are interesting from a practical viewpoint as well. In many settings, there is a subset of "interested agents" for an item who all value the item the same, and the rest of the agents do not value it. For example, in a partnership dissolution, a risky asset j could be seen as a good by some agents, who all value it at v_j (because the forecasted return is v_j), but other agents

---

[1]EFX_0 is a stronger notion where the EFX condition wrt goods also considers zero valued items. Similarly, $PropMX_0$ is a stronger notion where the maximin good in the definition of PropMX is allowed to have zero value.

might not be interested or even value it negatively due to the inherent risk. This is captured by restricted (mixed) goods. Likewise, identical and binary preferences often arise in practice; binary valuations can be used by agents to indicate approval or indifference towards a particular good.

1.2 Other Related Work

EF1 and EFX. The envy-cycle algorithm [25] showed that allocations that are envy-free up to the removal of the most valued good (EF1) can be computed in polynomial-time; EF1 is a weaker fairness notion than EFX. This algorithm was adapted to show that EFX allocations for IDO goods can be computed in polynomial-time [28]. For bads, [11] showed a variant of envy-cycle algorithm (using the top-envy graph) can be adapted for computing an EF1 allocation for separable instances (which they refer to as doubly monotone instances). In the goods setting, EFX allocations exist for 3 agents [15], and for the class of bivalued instances [3].

EF1/EFX + PO. The following results are for the case of goods. Barman et al. [9] show that EF1+PO allocations exist for and can be computed in pseudo-polynomial time. Recently, [27] showed that for k-ary instances with constant k or for instances with constantly many agents, an EF1+PO allocation can be computed in polynomial-time. For bivalued instances, an EFX+PO allocation is poly-time computable [19]. For bivalued instances of chores, [20] showed that an EF1+PO is polynomial-time computable. For a mixed manna with only two agents, [4] showed that an EF1+PO allocation is polynomial-time time computable.

Mixed Manna. For mixed manna, [5] showed that a Prop1+PO allocation of an indivisible mixed manna is polynomial-time computable. For the special cases of identical or ternary utilities, [1, 2] showed that an EFX+PO allocation can be polynomial-time. The competitive division of a mixed manna has also been studied [18, 12].

1.3 Organization

Section 2 sets up various technical preliminaries on fairness notions and their relations. Section 3 describes our results for the RMG setting with identical bads. In Section 4 we discuss our result for Separable instances. In Section 5 we extend our results for the RMG setting for IDO and general bads. We discuss the results along with some open questions in Section 6. Appendix A contains two counterexamples of interest.

2 Preliminaries

Problem instance. A *fair division instance* is a tuple (N, M, V), where $N = [n]$ is a set of $n \in \mathbb{N}$ agents, $M = [m]$ is a set of $m \in \mathbb{N}$ indivisible items, and $V = \{v_1, \ldots, v_n\}$ is a set of utility functions, one for each agent $i \in N$. Each utility function $v_i : M \to \mathbb{R}$ is specified by m numbers $v_{ij} \in \mathbb{R}$, one for each item $j \in M$, which denotes the value agent i has for receiving item j. When $v_{ij} \geq 0$ for every $i \in N, j \in M$, we call the instance a *goods* instance. In the case of $v_{ij} < 0$ for every $i \in N, j \in M$, we call the instance a *bads* instance. Finally, when we place no restrictions on v_{ij}, the instance is called a *mixed manna* instance. We assume that the value functions are additive, that is, for every agent $i \in N$, and for $S \subseteq M$, $v_i(S) = \sum_{j \in S} v_{ij}$. For notational ease, we write $v(S - j)$ instead of $v(S \setminus \{j\})$ and $v(S + j)$ instead of $v(S \cup \{j\})$. Throughout this paper, unless stated otherwise, we assume every fair division instance is a mixed manna instance.

Partitioning the mixed manna. In the mixed manna setting, a useful way of partitioning the set of items M is into the three sets M^+, M^0, and M^-, where:

1. $M^+ = \{j \in M : \exists i \in N, \ v_{ij} > 0\}$ is the set of *mixed goods*,

2. $M^0 = \{j \in M : \forall i \in N, \ v_{ij} \leq 0 \text{ and } \exists i \in N, \ v_{ij} = 0\}$ is the set of *dummy bads*, and

3. $M^- = \{j \in M : \forall i \in N, \ v_{ij} < 0\}$ is the set of *pure bads*.

In other words, M^+ comprises of the items which are pure goods for some agent, and may be bads or dummies for others; M^0 comprises of items which are dummy for some agent, and are not goods for anyone; M^- comprises of items which are bads for everyone.

Instance types. We call a fair division instance (N, M, V) a *separable instance* if we can partition M into $M^{\geq 0}$ and M^-, where $M^{\geq 0} := \{j \in M \mid \forall i \in N, \ v_{ij} \geq 0\}$, the set of items which are not bads for any agent, and M^- is the set of pure bads.

We also define the setting of *restricted mixed goods*, in which for every $j \in M^+$, there exists a value $v_j > 0$ such that for all $i \in N$, if $v_{ij} > 0$, then $v_{ij} = v_j$ (notice that if $v_{ij} \leq 0$ for a mixed good j, then no restrictions are placed on v_{ij}). A special case of the restricted mixed goods setting is the *binary mixed goods* setting, where for all $j, j' \in M^+$, $v_j = v_{j'}$.

An instance is called *identical ordering (IDO)*, if all agents have the same ordinal preference for all items, i.e., there exists an ordering of the items in M such that for all agents $i \in N$, $v_{i1} \leq v_{i2} \leq \cdots \leq v_{im}$. A special case of an IDO instance is the *identical* setting, in which for every $j \in M$, $v_{ij} = v_{i'j}$ for all $i, i' \in N$.

Allocation. An *allocation* $\mathbf{x}$ of items to agents is an n-partition $\mathbf{x}_1, \ldots, \mathbf{x}_n$ of the items, where agent i is allotted the bundle $\mathbf{x}_i \subseteq M$, and gets a total utility of $v_i(\mathbf{x}_i)$.

Pareto-optimality. An allocation $\mathbf{y}$ Pareto-dominates an allocation $\mathbf{x}$ if $v_i(\mathbf{y}_i) \geq v_i(\mathbf{x}_i), \forall i \in N$ and there exists $h \in N$ s.t. $v_h(\mathbf{y}_h) > v_h(\mathbf{x}_h)$. An allocation is said to be *Pareto-optimal* (PO) if no Pareto-allocation dominates it.

Welfare functions. Given an allocation $\mathbf{x}$:

1. the (utilitarian) social welfare $\mathsf{SW}(\mathbf{x})$ of $\mathbf{x}$ is the sum of agents' utilities under $\mathbf{x}$, i.e., $\mathsf{SW}(\mathbf{x}) = \sum_{i \in N} v_i(\mathbf{x}_i)$.

2. the Nash welfare $\mathsf{NW}(\mathbf{x})$ of $\mathbf{x}$ is the geometric mean of the agents' utilities under $\mathbf{x}$, i.e., $\mathsf{NW}(\mathbf{x}) = (\prod_i v_i(\mathbf{x}_i))^{1/n}$.

Any allocation $\mathbf{x}$ that maximizes the social welfare or the Nash welfare is Pareto-optimal, since a dominating allocation will have higher welfare than $\mathbf{x}$, which is not possible.

Fairness notions. We now define the fairness notions of interest.

Definition 1. (Envy-freeness and its relaxations.) An allocation $\mathbf{x}$ is said to be:

1. *Envy-free* if for all $i, h \in N$, $v_i(\mathbf{x}_i) \geq v_i(\mathbf{x}_h)$.

2. *Envy-free up to any item* (EFX) if for all $i, h \in N$ either

(i) $v_i(\mathbf{x}_i) \geq v_i(\mathbf{x}_h - g) \quad \forall g \in \mathbf{x}_h$ s.t. $v_{ig} > 0$, or

(ii) $v_i(\mathbf{x}_i - c) \geq v_i(\mathbf{x}_h) \quad \forall c \in \mathbf{x}_i$ s.t. $v_{ic} < 0$.

3. EFX_0 if for all $i, h \in N$ either

(i) $v_i(\mathbf{x}_i) \geq v_i(\mathbf{x}_h - g) \quad \forall g \in \mathbf{x}_h$ s.t. $v_{ig} \geq 0$, or

(ii) $v_i(\mathbf{x}_i - c) \geq v_i(\mathbf{x}_h) \quad \forall c \in \mathbf{x}_i$ s.t. $v_{ic} < 0$.

The difference between the definitions of EFX and EFX_0 is that EFX allows for the envy of an agent i towards agent h to disappear after removing any positively-valued item from the bundle of h, whereas in EFX_0 this envy must disappear after removing any non-negative valued item. Thus, EFX_0 is a stronger notion than EFX, and it is easy to see that any EFX_0 allocation is EFX, but not vice-versa.

We say that an agent i *envies* an agent h if $v_i(\mathbf{x}_i) < v_i(\mathbf{x}_h)$. Likewise, we say that an agent i *EFX-envies* (resp. EFX_0-envies) an agent h if neither conditions (i) nor (ii) of (2) (resp. (3)) hold for i with respect to h.

Definition 2. (Envy Graph.) The *envy-graph* of an allocation $\mathbf{x}$ is a directed graph $G_{\mathbf{x}} = (N, E)$ where each agent is a node and there exists an edge from agent i to agent h if and only if $v_i(\mathbf{x}_i) < v_i(\mathbf{x}_h)$, i.e. if and only if i envies h. The *top envy-graph* $G_{\mathbf{x}}^* = (N, E)$ is a directed graph where there is an edge from $i \in N$ to $h \in N$ if i envies h and additionally $h \in \mathsf{argmax}_{i' \neq i} v_i(\mathbf{x}_{i'})$.

A *source* in $G_{\mathbf{x}}$ is an agent with in-degree zero, i.e., an agent who nobody envies. A *sink* in $G_{\mathbf{x}}$ is an agent with out-degree zero, i.e., and agent who envies nobody. Notice that if there exists a (directed) cycle C in the envy-graph for some allocation, then we can reallocate bundles among the agents in C in the reverse order and all agents in C receive a bundle that they prefer over their current bundle, while the utility of agents outside of C does not change. This procedure is called *envy-cycle elimination* and results in a Pareto-improvement.

Definition 3. (Proportionality and its relaxations.) An allocation $\mathbf{x}$ is said to be:

1. *Proportional* if for all agents $i \in N$, we have $v_i(\mathbf{x}_i) \geq \mathsf{Prop}_i$, where $\mathsf{Prop}_i = \frac{1}{n} \cdot v_i(M)$ is the *proportional share* of an agent.

2. *Proportional up to the maximin good* (PropM) for a goods instance if, for all $i \in N$:

$$v_i(\mathbf{x}_i) + d_i(\mathbf{x}) \geq \mathsf{Prop}_i,$$

where $d_i(\mathbf{x}) = \max_{i' \neq i} \min_{\substack{j \in \mathbf{x}_{i'} \\ v_{ij} > 0}} v_{ij}$. The corresponding item maximizing the expression of $d_i(\mathbf{x})$ is called the *maximin good* of i for $\mathbf{x}$.

3. PropM_0 for a goods instance if for all $i \in N$:

$$v_i(\mathbf{x}_i) + d_i(\mathbf{x}) \geq \mathsf{Prop}_i,$$

where $d_i(\mathbf{x}) = \max_{i' \neq i} \min_{\substack{j \in \mathbf{x}_{i'} \\ v_{ij} \geq 0}} v_{ij}$. Note that PropM_0 is a more demanding condition than PropM, and it is easy to see that any PropM_0 allocation is PropM, but not vice versa.

4. *Proportional up to any bad* (PropX) for a bads instance if for all $i \in N$ and $\forall c \in \mathbf{x}_i$:

$$v_i(\mathbf{x}_i - c) \geq \mathsf{Prop}_i.$$

One can analogously define PropX for goods, however PropX allocations of goods need not always exist. We include an example due to Aziz, Moulin and Sandomirskiy [26, 5] in Appendix A.1. For the mixed manna setting, we combine the definitions above:

5. *Proportional up to the maximin good or any bad* (PropMX) for a MiXed manna instance, if for all $i \in N$ either:

   (i) $v_i(\mathbf{x}_i) + d_i(\mathbf{x}) \geq \mathsf{Prop}_i$, where $d_i(\mathbf{x}) = \max\limits_{i' \neq i} \min\limits_{\substack{j \in \mathbf{x}_{i'} \\ v_{ij} > 0}} v_{ij}$, or

   (ii) $\forall c \in \mathbf{x}_i$ such that $v_{ic} < 0$, $v_i(\mathbf{x}_i - c) \geq \mathsf{Prop}_i$.

6. PropMX$_0$ for a MiXed manna instance, if for all $i \in N$ either:

   (i) $v_i(\mathbf{x}_i) + d_i(\mathbf{x}) \geq \mathsf{Prop}_i$, where $d_i(\mathbf{x}) = \max\limits_{i' \neq i} \min\limits_{\substack{j \in \mathbf{x}_{i'} \\ v_{ij} \geq 0}} v_{ij}$, or

   (ii) $\forall c \in \mathbf{x}_i$ such that $v_{ic} < 0$, $v_i(\mathbf{x}_i - c) \geq \mathsf{Prop}_i$. As before, any PropMX$_0$ allocation is PropMX, but not vice-versa.

Relating the fairness properties. First notice that for any mixed manna instance, every agent i that has $v_i(M) \leq 0$ can be trivially PropMX$_0$ satisfied by allocating no items to them. Therefore, for the remainder of the paper, we assume that $v_i(M) > 0$ for all agents.

It is easy to see that in the case of an instance with bads only, EFX (resp.EFX$_0$) implies PropX([24], Lemma 3.2). We extend this observation and show that this implication continues to hold in the mixed manna setting.

Lemma 1. *Consider a mixed manna instance and let $\mathbf{x}$ be an EFX allocation for that instance. Then $\mathbf{x}$ is also PropMX. Further, if $\mathbf{x}$ is EFX$_0$, then $\mathbf{x}$ is also PropMX$_0$.*

Proof. Since $\mathbf{x}$ is EFX, we know that for all $i, h \in N$, either

$$v_i(\mathbf{x}_i) \geq v_i(\mathbf{x}_h) - v_{ig} \quad \forall g \in \mathbf{x}_h \text{ s.t. } v_{ig} > 0, \tag{1}$$

or

$$v_i(\mathbf{x}_i) - v_{ic} \geq v_i(\mathbf{x}_h) \quad \forall c \in \mathbf{x}_i \text{ s.t. } v_{ic} < 0. \tag{2}$$

Fix an agent $i \in N$ and let $N^+ \subseteq N$ denote the set of agents that i does not envy up to any item because of (1), and $N^- \subseteq N$ denote the set of agents that i does not envy up to any item because of (2). If, in (1), we select $g \in \mathbf{x}_h$ to be the minimum-value pure good of $\mathbf{x}_h$ according to i's valuation function, and then select the maximum such good over all $h \in N^+$, we get that this item is the maximin good for agent i, and thus, for all $h \in N^+$

$$v_i(\mathbf{x}_i) + d_i(\mathbf{x}) \geq v_i(\mathbf{x}_h). \tag{3}$$

We sum up (3) for all $h \in N^+$ and (2) for all $h \in N^-$, making sure the case of $h = i$ is counted only once, and then sum both inequalities together, and we get that, for all $c \in \mathbf{x}_i$ such that $v_{ic} < 0$,

$$n \cdot v_i(\mathbf{x}_i) + |N^+| \cdot d_i(\mathbf{x}) - |N^-| \cdot v_{ic} \geq v_i(M) \iff$$
$$v_i(\mathbf{x}_i) + \frac{|N^+|}{n} \cdot d_i(\mathbf{x}) - \frac{|N^-|}{n} \cdot v_{ic} \geq \frac{1}{n} \cdot v_i(M) = \mathsf{Prop}_i.$$

Notice that, for any $c \in \mathbf{x}_i$ such that $v_{ic} < 0$, $\frac{|N^+|}{n} \cdot d_i(\mathbf{x}) - \frac{|N^-|}{n} \cdot v_{ic}$ is a convex combination of $d_i(\mathbf{x})$ and $-v_{ic}$, and thus $\frac{|N^+|}{n} \cdot d_i(\mathbf{x}) - \frac{|N^-|}{n} \cdot v_{ic} \leq \max\{d_i(\mathbf{x}), -v_{ic}\}$. Therefore, either

$$v_i(\mathbf{x}_i) + d_i(\mathbf{x}) \geq \mathsf{Prop}_i,$$

7

or, for all $c \in \mathbf{x}_i$ such that $v_{ic} < 0$,

$$v_i(\mathbf{x}_i - c) \geq \mathsf{Prop}_i,$$

thus showing that $\mathbf{x}$ is PropMX.

Finally, if $\mathbf{x}$ is EFX_0, all inequalities still hold if we let $d_i(\mathbf{x}) = \max_{i' \neq i} \min_{\substack{j \in \mathbf{x}_{i'} \\ v_{ij} \geq 0}} v_{ij}$, and thus it follows that $\mathbf{x}$ is also PropMX_0. $\qquad \square$

The following lemma shows allocating items to agents who value them at the highest possible value results in a PO allocation. For any $j \in M$, let $\eta_j = \max_{i \in N} v_{ij}$.

Lemma 2. *Let $\mathbf{x}$ be an allocation in which every item $j \in M$ is allocated to an agent i s.t. $v_{ij} = \eta_j$. Then $\mathbf{x}$ is PO and maximizes the social welfare.*

Proof. The social welfare of $\mathbf{x}$ is $\mathsf{SW}(\mathbf{x}) = \sum_{i \in N} \sum_{j \in \mathbf{x}_i} v_{ij} = \sum_{i \in N} \sum_{j \in \mathbf{x}_i} \eta_j = \sum_{j \in M} \eta_j$. Further, for any allocation $\mathbf{y}$, $\mathsf{SW}(\mathbf{y}) = \sum_{i \in N} \sum_{j \in \mathbf{y}_i} v_{ij} \leq \sum_{i \in N} \sum_{j \in \mathbf{y}_i} \eta_j \leq \sum_{j \in M} \eta_j$, since $v_{ij} \leq \eta_j$ for every i, j. Hence, $\mathbf{x}$ maximizes the social welfare.

If $\mathbf{x}$ is not PO, then there must be an allocation $\mathbf{y}$ which dominates $\mathbf{x}$, i.e., $v_i(\mathbf{y}_i) \geq v_i(\mathbf{x}_i)$ for every $i \in N$ and $v_h(\mathbf{y}_h) > v_h(\mathbf{x}_h)$ for some $h \in N$. Then $\mathsf{SW}(\mathbf{y}) > \mathsf{SW}(\mathbf{x})$. This gives $\sum_{j \in M'} \eta_j \geq \mathsf{SW}(\mathbf{y}) > \mathsf{SW}(\mathbf{x}) = \sum_{j \in M} \eta_j$, which is a contradiction. Hence $\mathbf{x}$ is PO. $\qquad \square$

3 The Restricted Mixed Goods Setting

In this section, we investigate whether the fairness notions of EFX, EFX_0, PropMX or PropMX_0 can be achieved in conjunction with the efficiency notion of Pareto-optimality (PO) in the mixed manna setting with restricted mixed goods. We first note that PropMX_0 and PO are not compatible by presenting an instance in Appendix A.2 for which no allocation is PropMX_0+PO, even when no item is a bad. This also shows that EFX_0+PO allocations needn't exist. Thus, we consider the existence of PropMX+PO and EFX+PO allocations.

3.1 EFX+PO for Restricted Goods

We begin by showing that we can obtain an EFX+PO allocation for the setting of *pure goods with restricted valuations*. In such an instance (N, M, V), for every $j \in M$, there exists a $v_j > 0$ s.t. $v_{ij} \in \{0, v_j\}$ for every $i \in N$.

Theorem 1. *Given a fair division instance of pure goods with restricted valuations, an allocation that is EFX, PO and maximizes the utilitarian social welfare can be computed in polynomial-time.*

Proof. We prove this theorem by showing that Algorithm 1, which is based on the EnvyCycle procedure, computes an EFX+PO allocation for this setting.

First note that Algorithm 1 always allocates a good j to an agent i who values it at the highest, i.e., $v_{ij} = v_j$. Hence by Lemma 2, the allocation is PO at every step of the algorithm.

We now show that the partial allocation $\mathbf{x}$ is EFX at every step of Algorithm 1. Initially, $\mathbf{x}$ is an empty allocation (Line 2) and hence is trivially EFX. Inductively, let $\mathbf{x}$ be an EFX allocation of goods 1 through $j - 1$, for some $j \in [m]$, and let $G_{\mathbf{x}}$ be the envy-graph defined by $\mathbf{x}$. To maintain PO, we must allocate item j to an agent i s.t. $v_{ij} > 0$. Among the set of such agents N_j for an item j (Line 5), we allocate j to an agent i who is a source in the sub-graph G_j of $G_{\mathbf{x}}$ induced by N_j. We argue below that such an agent always exists by showing that $G_{\mathbf{x}}$ is acyclic. Intuitively, i is the right choice because giving the good j to an agent $h \in N_j$ who is already envied by some agent k can cause k to envy h even after the removal of j.

Algorithm 1 EFX+PO For Restricted Goods

Input: Restricted Goods Instance (N, M, V)

Output: Allocation $\mathbf{x}$

1: Order and relabel the goods so that $v_1 \geq v_2 \geq \cdots \geq v_m > 0$
2: $\mathbf{x} \leftarrow (\emptyset, \ldots, \emptyset)$ ▷ Initial empty allocation
3: **while** $M \neq \emptyset$ **do**
4:   Pick $j \in M$
5:   $N_j \leftarrow \{i \in N : v_{ij} = v_j\}$ ▷ Agents who value j positively
6:   Let $G_{\mathbf{x}}$ be the envy-graph defined by $\mathbf{x}$ ▷ Def. 2
7:   Let $G_j = G_{\mathbf{x}}[N_j]$ be the sub-graph of $G_{\mathbf{x}}$ induced by N_j
8:   Let $i \in N_j$ be a *source* in G_j ▷ Def. 2
9:   $\mathbf{x}_i \leftarrow \mathbf{x}_i + j$ ▷ Assign j to i
10:   $M \leftarrow M - j$
11: **return** $\mathbf{x}$

More formally, we show that the allocation $\mathbf{x}'$ is EFX, where $\mathbf{x}'_i = \mathbf{x}_i + j$ and $\mathbf{x}'_h = \mathbf{x}_h$ for every $h \neq i$. Note that any violation of the EFX condition in $\mathbf{x}'$ must involve the agent i. Since $\mathbf{x}$ is EFX, we see that for any agent $h \neq i$, $v_i(\mathbf{x}'_i) = v_i(\mathbf{x}_i) + v_{ij} \geq v_i(\mathbf{x}_h - g) = v_i(\mathbf{x}'_h - g)$ for every $g \in \mathbf{x}_h$ with $v_{ig} > 0$. Thus i does not EFX-envy any other agent in $\mathbf{x}'$.

Consider $h \in N_j$. Since i is a source in G_j, h does not envy i. Hence $v_j(\mathbf{x}_h) \geq v_h(\mathbf{x}_i)$. Then for any $g \in \mathbf{x}'_i$ with $v_{hg} > 0$, we have $v_h(\mathbf{x}'_i - g) = v_h(\mathbf{x}_i) + v_{hj} - v_{hg} \leq v_h(\mathbf{x}_i) = v_h(\mathbf{x}'_h)$, since $\mathbf{x}$ is EFX and we ordered the goods so that $v_{hg} \geq v_{hj}$. Thus, h does not EFX-envy i in $\mathbf{x}'$.

Finally, consider $h \notin N_j$. Then $v_{hj} = 0$. Thus $v_h(\mathbf{x}'_h) = v_h(\mathbf{x}_h) \geq v_h(\mathbf{x}_i - g) = v_h(\mathbf{x}'_i - g)$ for any $g \in \mathbf{x}'_i$ s.t. $v_{hg} > 0$, since $\mathbf{x}$ is EFX. This shows that h does not EFX-envy i in $\mathbf{x}'$.

In conclusion, $\mathbf{x}'$ is EFX. Inductively, this shows that the algorithm always maintains EFX. Further, since a good is given to an agent who values it positively, the allocation must be PO throughout the execution of the algorithm. We now show that the envy graph $G_{\mathbf{x}}$ corresponding to a partial allocation $\mathbf{x}$ is acyclic, ensuring the presence of a source agent (Line 8). Assume for the sake of contradiction, there is a cycle C in $G_{\mathbf{x}}$. Then reallocating bundles along the cycle strictly improves the utility of every agent in the cycle, and does not change the utility of any agent not in the cycle. Hence, this is a Pareto-improvement over $\mathbf{x}$. This contradicts the fact that $\mathbf{x}$ is PO. Therefore, $G_{\mathbf{x}}$ cannot have cycles. $\square$

3.2 EFX+PO for Restricted Mixed Goods and Identical Bads

We now use the result of the preceding section for computing EFX+PO allocations in the mixed setting comprising of *restricted mixed goods and identical bads*. Recall that for such instances:

1. for every $j \in M^+$, there exists $v_j \in \mathbb{Z}^+$ s.t. for every $i \in N$ with $v_{ij} > 0$, it holds that $v_{ij} = v_j$.

2. for every $j \in M^-$, there exists $v_j \in \mathbb{Z}^-$ s.t. for every $i \in N$, it holds that $v_{ij} = v_j$.

We show that:

Theorem 2. *Given a fair division instance with restricted mixed goods and identical bads, an allocation that is EFX, PropMX, PO and maximizes the social welfare be computed in polynomial-time.*

Algorithm 2 EFX+PO for Restricted Mixed Goods & Id. Bads
___
Input: Instance (N, M, V) with restricted mixed goods and identical bads
Output: Allocation $\mathbf{x}$
1: Partition M into M^+, M^0, M^- ▷ See Sec. 2
     Phase 1: Allocating M^+ ....
2: Let V' be given by
$$v'_{ij} = \begin{cases} v_{ij}, \text{ if } v_{ij} > 0 \\ 0, \text{ if } v_{ij} \leq 0, \end{cases}$$

     for every $i \in N$ and $j \in M^+$
3: $\mathbf{x} \leftarrow$ ALGORITHM $1(N, M^+, V')$ ▷ Allocate M^+ by running Algorithm 1 for items in M^+ with modified values
     Phase 2: Allocating M^0 ....
4: **while** $M^0 \neq \emptyset$ **do**
5:     Pick $j \in M^0$
6:     Let $i \in N$ be such that $v_{ij} = 0$
7:     $\mathbf{x}_i \leftarrow \mathbf{x}_i + j$ ▷ Assign j to i
8:     $M^0 \leftarrow M^0 - j$
     Phase 3: Allocating M^- ....
9: Order bads in M^- according to $\prec$ s.t. $j \prec j'$ iff $-v_j \geq -v_{j'}$
10: **while** $M^- \neq \emptyset$ **do**
11:     Pick smallest $j \in M^-$ according to $\prec$
12:     Let $G_{\mathbf{x}}$ be the envy-graph defined by $\mathbf{x}$ ▷ Def. 2
13:     Let i be a *sink* in $G_{\mathbf{x}}$ ▷ Def. 2
14:     $\mathbf{x}_i \leftarrow \mathbf{x}_i + j$ ▷ Assign j to i
15:     $M^- \leftarrow M^- - j$
16: **return** $\mathbf{x}$
___

We prove Theorem 2 by showing that our Algorithm 2 computes an EFX+PO allocation for the given setting.

We divide the execution of the algorithm into three phases: Phase 1 allocates items in M^+ (Lines 2-3), followed by Phase 2 for allocating items in M^0 (Lines 4-9), and finally Phase 3 for allocating items in M^- (Lines 10-16). We show that at each iteration of the algorithm, the partial allocation of items allocated so far is always EFX+PO.

We first describe Phase 1. Given the instance $I = (N, M^+, V)$, we consider the instance $I' = (N, M^+, V')$ with values modified by changing the negative values to zero (see Line 2). Then, I' is an instance of goods with restricted additive valuations.

Using Algorithm 1, we obtain an allocation $\mathbf{x}$ in Line 3 which is EFX+PO for the instance I'. We show that $\mathbf{x}$ is also EFX for I, thus showing that:

Lemma 3. *The allocation at the end of Phase 1 is EFX+PO.*

Proof. As discussed above, the allocation $\mathbf{x}$ at the end of Phase 1 is EFX+PO for the instance $I' = (N, M^+, V')$ with modified values. We now show that $\mathbf{x}$ is EFX for the original instance $I = (N, M^+, V)$ as well. To this end, observe that since $\mathbf{x}$ is PO for I', no agent i has an item j which she values at 0. Otherwise transferring j to an agent h who values j positively is a Pareto-improvement; such an agent h exists because $j \in M^+$. Thus, $j \in \mathbf{x}_i$ implies $v'_{ij} > 0$. However, by

construction this implies $v'_{ij} = v_{ij}$ (see Line 2). Thus,

$$v'_i(\mathbf{x}_i) = v_i(\mathbf{x}_i). \tag{4}$$

Fix a pair of agents i, h. For any $j \notin \mathbf{x}_i$, $v_{ij} \leq v'_{ij}$ by construction. Hence:

$$v_i(\mathbf{x}_h - g) \leq v'_i(\mathbf{x}_h - g), \tag{5}$$

for any $g \in \mathbf{x}_h$ with $v_{ig} > 0$. Since $\mathbf{x}'$ is EFX for I', $v'_i(\mathbf{x}_i) \geq v'_i(\mathbf{x}_h - g)$. Together with (4) and (5), we obtain:

$$v_i(\mathbf{x}_i) \geq v_i(\mathbf{x}_h - g),$$

for any $g \in \mathbf{x}_h$ with $v_{ig} > 0$. This shows that $\mathbf{x}$ is EFX for I at the end of Phase 1. Since each item j is allocated to an agent i who values it at the highest $(v_{ij} = v_j)$, Lemma 2 shows $\mathbf{x}$ is PO. $\square$

Next, we describe Phase 2 which allocates items in M^0. To ensure that the allocation is PO, we must allocate each item $j \in M^0$ to an agent i s.t. $v_{ij} = 0$. Intuitively, since the EFX condition requires that the envy between agents disappear after the removal of an item of *positive* value, items of M^0 will not cause any new EFX-envy.

Lemma 4. *The allocation at the end of Phase 2 is EFX+PO.*

Proof. Suppose $\mathbf{x}$ is a partial EFX allocation prior to allocating an item $j \in M^0$. Suppose we allocate j to an agent i with $v_{ij} = 0$, to obtain an allocation $\mathbf{x}'$. Since $\mathbf{x}_h = \mathbf{x}'_h$ for every $h \neq i$, any possible violation of the EFX condition in $\mathbf{x}'$ must involve agent i. First, since i does not get a bad, the utility of i does not change, and neither does the utility of any other agent. Hence i will not EFX-envy any other agent in $\mathbf{x}'$. Now since $\mathbf{x}$ is EFX, we have for any other $h \neq i$, $v_h(\mathbf{x}'_h) = v_h(\mathbf{x}_h) \geq v_h(\mathbf{x}_i - g) \geq v_h(\mathbf{x}'_i - g)$, for any $g \in \mathbf{x}'_i$ with $v_{hg} > 0$. Hence $\mathbf{x}'$ is EFX.

Finally, Lemma 2 once again shows that the allocation at the end of Phase 2 is PO, since any $j \in M^+ \cup M^0$ is allocated to an agent i who values it at the highest-possible value. $\square$

We finally describe Phase 3 which allocates items of M^-. Since the bads form an identical instance, i.e. a given bad j has the same value $v_j < 0$ for all agents, we first sort the items in non-increasing order of disutility, i.e. we order the bads using $\prec$ where $j \prec j'$ iff $-v_{ij} \geq -v_{ij'}$.

Let $\mathbf{x}$ be a partial EFX+PO allocation items in M^+, M^0 and some items of M^-. We must now decide how to allocate the next bad $j \in M^-$ in the order defined by $\prec$. For this, we consider the envy-graph $G = (N, E)$ (see Def. 2). We argue that the way we allocate items ensures that G will have at least one *sink* agent, i.e., an agent who does not envy any agent. Then, we give j to a sink i (Line 14). Intuitively, this is the right choice, as i is already well-off in terms of value as i does not envy any other agent. Hence we should give the bad to i instead of any other agent who has lesser value. We now formally show that:

Lemma 5. *The allocation at the end of Phase 3 is EFX+PO.*

Proof. Suppose $\mathbf{x}$ is a partial EFX allocation prior to allocating an item $j \in M^-$. Suppose we allocate j to a sink agent i in the envy-graph $G_\mathbf{x}$ to obtain an allocation $\mathbf{x}'$. We will show later that G is acyclic, ensuring such a sink-agent exists. For sake of contradiction assume $\mathbf{x}'$ is not EFX. Since $\mathbf{x}_h = \mathbf{x}'_h$ for every $h \neq i$, any possible violation of the EFX condition must involve agent i. We claim that i is EFX towards any other agent $h \neq i$ in $\mathbf{x}'$. To see this, note that for any bad $j' \in \mathbf{x}'_i$, $v_i(\mathbf{x}'_i - j') = v_i(\mathbf{x}_i) + v_{ij} - v_{ij'} \geq v_i(\mathbf{x}_i)$ since $j' < j$. Now since i is a sink, $v_i(\mathbf{x}_i) \geq v_i(\mathbf{x}_h)$, thus implying that $v_i(\mathbf{x}'_i - j') \geq v_i(\mathbf{x}_h)$ for every bad $j' \in \mathbf{x}'_i$ and any agent $h \neq i$, showing that i does not EFX-envy any other agent.

Now suppose some agent $h \neq i$ EFX-envies i in $\mathbf{x}'$. Since $\mathbf{x}$ was EFX, either (1) $v_h(\mathbf{x}_h) \geq v_h(\mathbf{x}_i - g)$ for every $g \in \mathbf{x}_i$ with $v_{hg} > 0$, or (2) $v_h(\mathbf{x}_h - j') \geq v_h(\mathbf{x}_i)$ for every $j' \in \mathbf{x}_h$ with $v_{hj'} < 0$. If Condition (1) holds, then we also obtain $v_h(\mathbf{x}'_h) = v_h(\mathbf{x}_h) \geq v_h(\mathbf{x}'_i - g)$ for every $g \in \mathbf{x}_i$ with $v_{hg} > 0$, since $\mathbf{x}'_h = \mathbf{x}_h$ and $\mathbf{x}'_i = \mathbf{x}_i + j$ but $v_{hj} \leq 0$. Thus h continues to be EFX towards i in $\mathbf{x}'$. If Condition (2) holds, then, using $v_{hj} \leq 0$ we obtain $v_h(\mathbf{x}'_h - j') = v_h(\mathbf{x}_h - j') \geq v_h(\mathbf{x}_i) \geq v_h(\mathbf{x}'_i)$ for every $j' \in \mathbf{x}'_h$ with $v_{hj'} < 0$, once again showing that h does not EFX-envy i in $\mathbf{x}'$.

Thus $\mathbf{x}'$ is EFX. Further, note that any item is allocated to an agent who values it at the highest. By Lemma 2, the allocation remains PO throughout the execution of the algorithm. As argued before, if the envy-graph $G_\mathbf{x}$ corresponding to some partial allocation $\mathbf{x}$ has a cycle, then reallocating along the cycle gives a Pareto-improvement. Since $\mathbf{x}$ is always PO, this means that $G_\mathbf{x}$ is always acyclic, ensuring the presence of sink agents (Line 19). Thus, the allocation at the end of Phase 3 is EFX+PO. □

In conclusion, Algorithm 2 returns and EFX+PO allocation for the restricted mixed goods and identical bads setting. Note that any item is allocated to an agent who values it at the highest, hence the allocation also maximizes the social welfare by Lemma 2. Finally, Lemma 1 implies that any EFX allocation is PropMX. This proves Theorem 2.

3.3 EFX_0+PO for Binary Mixed Goods and Identical Bads

Next, we show that for the special case of instances with binary mixed goods and identical bads, an EFX_0+PO allocation can be computed in polynomial time.

Theorem 3. *Given a fair division instance with binary mixed goods and identical bads, an allocation that is EFX_0, $PropMX_0$, PO and maximizes the social welfare be computed in polynomial-time.*

Recall that in this setting, there exists an $a > 0$ such that for all $j \in M^+$, if $v_{ij} > 0$, then $v_{ij} = a$. In the restricted mixed goods setting, an EFX_0+PO allocation need not even exist (Appendix A.2). Further, even for the binary mixed goods case, Algorithm 2 cannot provide EFX_0 guarantee because it may assign a mixed good j to an agent i, and there could be another agent h that envies i but $v_{hj} = 0$, thus causing h to EFX_0-envy i.

We circumvent this issue by using a different approach than Algorithm 2 for allocating M^+ in Phase 1 and M^0 in Phase 2. Specifically, we modify the values of items in M^+ as before to convert it into a binary goods instance I', and then use algorithm ALG-BINARY of [3]. This algorithm is adapted from [16, 10], and computes a Nash-welfare maximizing allocation of I' which is also EFX_0+PO ([3], Theorem 3.1). Afterwards, we adapt Phase 2 of Algorithm 2, paying extra care to which agents we allocate the items in M^0. Finally, Phase 3 of Algorithm 2 remains the same.

We prove Theorem 3 by showing that Algorithm 3 computes an EFX_0+PO for the given setting. We first describe Phase 1 of Algorithm 3. Given the instance $I = (N, M^+, V)$, we consider the instance $I' = (N, M^+, V')$ with modified values, as in the restricted setting. Notice that I' is an instance with binary values. For such instances, an allocation which maximizes the Nash welfare can be computed in polynomial-time [16, 10], and further a Nash welfare maximizing allocation which is also EFX_0+PO can be computed in polynomial-time [3] via the procedure ALG-BINARY of [3] (which we rename here to BINARYGOODS to avoid confusion). Using this, we obtain an allocation $\mathbf{x}$ in Line 3 which is EFX_0+PO for the instance I'. Like in Lemma 3, we show that $\mathbf{x}$ is also EFX_0 for I.

Lemma 6. *The allocation at the end of Phase 1 is EFX_0+PO.*

12

Algorithm 3 EFX_0+PO for Binary Mixed Goods & Id. Bads
___
Input: Instance (N, M, V) with binary mixed goods and identical bads
Output: Allocation $\mathbf{x}$

1: Partition M into M^+, M^0, M^- $\hspace{3cm}$ ▷ See Sec.2
$\quad$ *Phase 1: Allocating M^+* ....
2: Let V' be given by
$$v'_{ij} = \begin{cases} v_{ij}, \text{ if } v_{ij} > 0 \\ 0, \text{ if } v_{ij} \leq 0, \end{cases}$$
$\quad$ for every $i \in N$ and $j \in M^+$
3: $\mathbf{x} \leftarrow \text{BinaryGoods}(N, M^+, V')$ $\quad$ ▷ Allocate M^+ via BinaryGoods for items in M^+ with modified values
$\quad$ *Phase 2: Allocating M^0* ....
4: **while** $M^0 \neq \emptyset$ **do**
5: $\quad$ Pick $j \in M^0$
6: $\quad$ Let $N_j := \{i \in N \mid v_{ij} = 0\}$
7: $\quad$ Let $G_{\mathbf{x}}$ be the envy-graph defined by $\mathbf{x}$ $\hspace{2cm}$ ▷ Def. 2
8: $\quad$ Let $G_j = G_{\mathbf{x}}[N_j]$ be the sub-graph of $G_{\mathbf{x}}$ induced by N_j
9: $\quad$ Let $i \in N_j$ be a *source* in G_j $\hspace{3cm}$ ▷ Def. 2
10: $\quad$ $\mathbf{x}_i \leftarrow \mathbf{x}_i + j$ $\hspace{3cm}$ ▷ Assign j to i $M^0 \leftarrow M^0 - j$
$\quad$ *Phase 3: Allocating M^-* ....
11: Order bads in M^- according to $\prec$ s.t. $j \prec j'$ iff $-v_j \geq -v_{j'}$
12: **while** $M^- \neq \emptyset$ **do**
13: $\quad$ Pick smallest $j \in M^-$ according to $\prec$
14: $\quad$ Let $G_{\mathbf{x}}$ be the envy-graph defined by $\mathbf{x}$ $\hspace{2cm}$ ▷ Def. 2
15: $\quad$ Let i be a sink in G $\hspace{4cm}$ ▷ Def. 2
16: $\quad$ $\mathbf{x}_i \leftarrow \mathbf{x}_i + j$ $\hspace{3cm}$ ▷ Assign j to i $M^- \leftarrow M^- - j$
17: **return** $\mathbf{x}$

Proof. As discussed above, we know that the allocation $\mathbf{x}$ at the end of Phase 1 is EFX_0+PO for the instance $I' = (N, M^+, V')$ with modified values. We now show that $\mathbf{x}$ is EFX_0 for the original instance $I = (N, M^+, V)$ as well. To this end, observe that, since $\mathbf{x}$ is PO for I', no agent i has an item j which she values at 0. Otherwise transferring j to an agent h who values j positively is a Pareto-improvement; such an agent h exists because $j \in M^+$. Thus, $j \in \mathbf{x}_i$ implies $v'_{ij} > 0$. However, by construction this implies $v_{ij} > 0$ (see Line 2). Hence, for any agent i and item j, if $j \in x_i$, then $v_{ij} > 0$, i.e., every agent gets items that are goods for them. Thus,

$$v'_i(\mathbf{x}_i) = v_i(\mathbf{x}_i). \tag{6}$$

Fix a pair of agents $i, h \in N$. For any $j \notin \mathbf{x}_i$, $v_{ij} \leq v'_{ij}$ by construction. Hence:

$$v_i(\mathbf{x}_h - g) \leq v'_i(\mathbf{x}_h - g), \tag{7}$$

for any $g \in \mathbf{x}_h$. Since $\mathbf{x}'$ is EFX_0 for I', $v'_i(\mathbf{x}_i) \geq v'_i(\mathbf{x}_h - g)$. Together with (6) and (7), we obtain:

$$v_i(\mathbf{x}_i) \geq v_i(\mathbf{x}_h - g),$$

for any $g \in \mathbf{x}_h$. This shows that $\mathbf{x}$ is EFX_0 for I at the end of Phase 1. Further each item j is allocated to some agent i who values it at the highest-possible value, thus showing that $\mathbf{x}$ is PO. $\quad\square$

Next, we describe Phase 2 which allocates items in M^0. The only difference between Algorithms 2 and 3 in this Phase is that, while in the former we allocated every item $j \in M^0$ to an arbitrary agent which has value 0 for j, in the latter, out of all agents i which have $v_{ij} = 0$, we allocate j to a source of the induced subgraph of the envy-graph. This ensures that no other agent h who envies i has $v_{hj} = 0$, which ensures the EFX_0 condition is not violated for h. We prove:

Lemma 7. *The allocation at the end of Phase 2 is EFX_0+PO.*

Proof. Consider the partial allocation $\mathbf{x}$ right before we allocate an item $j \in M^0$, and assume that it is EFX_0+PO. Suppose we allocate j to i. By the algorithm's description, we know that i is a source in G_j. First, since i's valuation for her bundle does not change, and no other agent's bundle changes, i continues to satisfy the EFX_0 condition. Next, we can partition $N - i$ into two sets, the set N_{-i}^E of agents who envy i and the set N_{-i}^{NE} of agents who do not envy i. For every agent $h \in N_{-i}^{NE}$, they continue to not envy i even after we allocate j, since $v_h(\mathbf{x}_i)$ can not increase, while $v_h(\mathbf{x}_h)$ remains the same. Therefore, all $h \in N_{-i}^{NE}$ continue to satisfy the EFX_0 condition. Finally, since i is a source in G_j, we know that every agent $h \in N_{-i}^E$ has $v_{hj} < 0$, else they would have an edge to i and i would not be a source. Therefore, j is not considered among the items $g \in \mathbf{x}_i$ for which $v_{hg} \geq 0$, and thus all $h \in N_{-i}^E$ continue to satisfy the EFX_0 condition. $\square$

Phase 3 of Algorithm 3 is exactly the same as Phase 3 of Algorithm 2 and, since allocating pure bads does not affect items for which agents have value 0, the allocation is EFX_0+PO after Phase 3, due to Lemma 5.

In conclusion, for instances with binary mixed goods and identical bads, Algorithm 3 computes an EFX_0+PO allocation in polynomial-time. Algorithm 3 allocates an item to an agent who values it at the highest, hence the allocation also maximizes the social welfare by Lemma 2. Finally, Lemma 1 implies that any EFX_0 allocation is $PropMX_0$. This proves Theorem 3.

4 $PropMX_0$ for Separable Instances

In this section, we consider separable instances, in which all agents agree on the set of goods and bads. In other words, we can partition M into $M^{\geq 0}$ and M^- such that $v_{ij} \geq 0$ for all $j \in M^{\geq 0}$ and $v_{ij} < 0$ for all $j \in M^{\geq 0}$, for every agent $i \in N$. We show that:

Theorem 4. *Given a separable fair division instance (N, M, V), a $PropMX_0$ allocation can be computed in polynomial-time.*

We prove this theorem by presenting an algorithm which returns a $PropMX_0$ allocation for this instance, combining ideas from previously known algorithms for instances with goods only [6] and pure bads instances [24].

The idea behind the algorithm is to first obtain a $PropMX_0$ allocation with respect to $M^{\geq 0}$ and then start allocating M^-. We show how this allocation of bads can be performed while maintaining $PropMX_0$ in the case of an IDO instance, in which all agents have the same ordinal preference for all bads in M^-. Recall that in an IDO instance, there exists an ordering of the bads in M^- such that for all agents $i \in N$, $v_{i1} \leq v_{i2} \leq \cdots \leq v_{im^-}$, where $m^- = |M^-|$. Notice that in an IDO instance, the cardinal preferences can be significantly different across different agents. Afterwards, in Section 4.2, we present a reduction from [24], from an instance which is IDO for bads to a general instance for bads, and show that it holds even in the mixed manna setting. This allows us to obtain a $PropMX_0$ allocation for any separable instance.

4.1 Separable Instances with IDO Bads

Our algorithm first uses the algorithm of Baklanov et al. [6] for goods to obtain a PropM_0 allocation with respect to $M^{\geq 0}$. Note that this allocation will also trivially be PropMX_0, since we have only assigned items from $M^{\geq 0}$. Also, Baklanov et al. do not differentiate between PropM and PropM_0, and thus mention their result as giving a PropM allocation, but in fact their allocation satisfies the stronger guarantee of being PropM_0. Afterwards, we run the envy-cycle elimination algorithm of Li et al. [24] for pure bads, and show that it obtains an allocation that is PropMX_0 with respect to M.

For the remainder of this section, we use GOODS to refer to Algorithm 1 of [6], and we assume that the bads in M^- are ordered such that $v_{i1} \leq v_{i2} \leq \cdots \leq v_{im^-}$ (recall that $v_{ij} < 0$ for all $i \in N, j \in M^-$, i.e. the bads are ordered from most painful to least painful). Furthermore, given any allocation $\mathbf{x}$ of a subset of items, we create the top-envy graph of $\mathbf{x}$, $G_{\mathbf{x}}^*$. Consider a cycle C of $G_{\mathbf{x}}^*$. We call a *reallocation according to C* a new allocation $\mathbf{x}^C$ where we reallocate the bundles of C backwards along the cycle.

Algorithm 4 Separable Instances with IDO Bads

Input: Separable instance (N, M, V)
Output: Allocation $\mathbf{x}$

  1: $M^{\geq 0} \leftarrow \{j \in M : \forall i \in N, \ v_{ij} \geq 0\}$
  2: $M^- \leftarrow M \setminus M^{\geq 0}$
  3: $\mathbf{x}^{\geq 0} \leftarrow \text{GOODS}(M^{\geq 0})$ $\triangleright \text{PropM}_0$ Algorithm from [6]
  4: $\mathbf{x} \leftarrow \mathbf{x}^{\geq 0}$
  5: **for** $j \leftarrow 1$ to m^- **do**
  6:     **while** $\nexists$ any sinks in $G_{\mathbf{x}}^*$ **do**
  7:         Let C be a cycle in $G_{\mathbf{x}}^*$
  8:         Reallocate bundles according to C $\triangleright$ See text
  9:     Let i be a sink in $G_{\mathbf{x}}^*$
10:     $\mathbf{x}_i \leftarrow \mathbf{x}_i + j$
11: **return** $\mathbf{x}$

Since GOODS runs in polynomial time, Algorithm 4 runs in polynomial time as well. The following lemma about allocation $\mathbf{x}^{\geq 0}$ (Line 3) follows directly from [6].

Lemma 8. *For every agent $i \in N$,*

$$v_i(\mathbf{x}_i^{\geq 0}) + d_{\bar{i}}^{\geq 0}(\mathbf{x}^{\geq 0}) \geq \frac{1}{n} \cdot v_i(M^{\geq 0}),$$

where $d_{\bar{i}}^{\geq 0}(\mathbf{x}^{\geq 0}) = \max_{i' \neq i} \min_{j \in \mathbf{x}_{i'}^{\geq 0}} v_{ij}$.

Next, we show that Algorithm 4 maintains PropMX_0 with respect to the currently allocated set of items after allocating a pure bad.

Lemma 9. *Let $M_{\leq j}^- = \{1, 2, \ldots, j\} \subseteq M^-$ denote the set of bads allocated after the j-th step of the For-Loop in Lines $5 - 12$. Then, after the j-th step, for all agents $i \in N$, either*

$$v_i(\mathbf{x}_i) + d_i(\mathbf{x}) \geq \frac{1}{n} \cdot v_i\left(M^{\geq 0} \cup M_{\leq j}^-\right),$$

15

where $d_i(\mathbf{x}) = \max_{i' \neq i} \min_{\substack{j \in \mathbf{x}_{i'} \\ v_{ij} \geq 0}} v_{ij}$, or $\forall c \in \mathbf{x}_i \setminus \mathbf{x}_i^{\geq 0}$,

$$v_i(\mathbf{x}_i - c) \geq \frac{1}{n} \cdot v_i \left(M^{\geq 0} \cup M^-_{\leq j} \right).$$

Proof. We prove the lemma via induction on j. Notice that before the For-Loop in Lines $5 - 12$ of Algorithm 4, the lemma holds, because of Lemma 8. Assume that the lemma holds right after the $(j-1)$-th step of the For-Loop, and consider the allocation $\mathbf{x}$ immediately after the j-th step of the For-Loop. Let i be the agent that received bad j, and let $\mathbf{x}_i^{(b)}$ denote i's bundle right before step j.

First, consider an agent $i' \neq i$. Notice that if $\mathbf{x}_{i'}$ did not change during the j-th step of the For-Loop, then the lemma holds for i', by our induction hypothesis, as the left-hand side of both inequalities remained the same, while the right-hand side decreased. Furthermore, the only way $\mathbf{x}_{i'}$ could have changed during the j-th step of the For-Loop is via a top-envy cycle reallocation in $G^*_{\mathbf{x}}$. In this case, i' received her bust bundle among all bundles of $\mathbf{x}$, since $G^*_{\mathbf{x}}$ is the top-envy graph, which implies that i' does not envy any other agent after the j-th step, thus $v_{i'}(\mathbf{x}_h) \geq \frac{1}{n} \cdot v_{i'} \left(M^{\geq 0} \cup M^-_{\leq j} \right)$ and i' satisfies the lemma.

Next, consider agent i. Since i was a sink in the top-envy graph $G^*_{\mathbf{x}}$ prior to step j of the For-Loop, we know that i did not envy any other agent, and thus $v_i(\mathbf{x}_i) \geq \frac{1}{n} \cdot v_i \left(M^{\geq 0} \cup M^-_{\leq j-1} \right)$. This implies that, for every bad $c \in \left(\mathbf{x}_i^{(b)} \setminus \mathbf{x}_i^{\geq 0} \right) + j$, we have

$$
\begin{aligned}
v_i \left(\mathbf{x}_i^{(b)} + j - c \right) &= v_i \left(\mathbf{x}_i^{(b)} + j \right) - v_{ic} \\
&= v_i \left(\mathbf{x}_i^{(b)} \right) + v_{ij} - v_{ic} \\
&\geq \frac{1}{n} \cdot v_i \left(M^{\geq 0} \cup M^-_{\leq j-1} \right) + v_{ij} - v_{ic} \\
&= \frac{1}{n} \cdot v_i \left(M^{\geq 0} \cup M^-_{\leq j} \right) - \frac{v_{ij}}{n} + v_{ij} - v_{ic} \\
&\geq \frac{1}{n} \cdot v_i \left(M^{\geq 0} \cup M^-_{\leq j} \right),
\end{aligned}
$$

where the last inequality follows from the fact that $v_{ic} \leq v_{ij} \leq \left(1 - \frac{1}{n} \right) v_{ij}$, since the instance is IDO for bads and $v_{ik} < 0$ for all $k \in M^-$. Therefore, i satisfies the lemma after the j-th step. $\square$

4.2 General Separable Instances

Consider now a general separable instance. Our idea is to again use the GOODS algorithm of [6], and then convert the subproblem of allocating M^- into an IDO instance via a reduction. This reduction appears in Li et al. [24], and is commonly used in designing approximation algorithms for MMS fair allocations [13, 8, 22]. We show here that it can be used in the mixed manna setting as well.

Lemma 10. *If there exists a polynomial time algorithm that given any separable mixed manna instance with IDO bads computes a PropMX (resp. PropMX$_0$) allocation, then there exists a polynomial time algorithm that given any separable mixed manna instance (with general bads) computes a PropMX (resp. PropMX$_0$) allocation.*

Proof. Given any separable mixed manna instance $I = (N, M, V)$, we start by creating $I' = (N, M, V')$, a new separable mixed manna instance with IDO bads. V' is defined as follows. For

16

all items that are not pure bads, the valuation stays the same for all agents, i.e. $v'_{ij} = v_{ij}$ for all $j \notin M^-$. Without loss of generality, let $M^- = \{1, 2, \ldots, m^-\}$. Next, let $\sigma_i(j) \in M$ be the j-th lowest-valued pure bad under value function v_i (i.e. $\sigma_i(1)$ is the "worst" bad for agent i), and let $v'_{ij} = v_{i\sigma_i(j)}$. It follows easily that instance I' has IDO bads, with $v'_{i1} \leq v'_{i2} \leq \ldots \leq v'_{im^-}$.

Afterwards, we run the algorithm for IDO instances on instance I', and get a PropMX (resp. PropMX$_0$) allocation $\mathbf{x}'$ with respect to I'. By definition, for every agent $i \in N$, we have either

$$v'_i(\mathbf{x}'_i) + d_i(\mathbf{x}') \geq \frac{1}{n} \cdot v'_i(M),$$

where $d_i(\mathbf{x}') = \max_{i' \neq i} \min_{\substack{j \in \mathbf{x}'_{i'} \\ v'_{ij} > 0}} v'_{ij}$ (resp. $d_i(\mathbf{x}') = \max_{i' \neq i} \min_{\substack{j \in \mathbf{x}'_{i'} \\ v'_{ij} \geq 0}} v'_{ij}$), or $\forall c \in \mathbf{x}'_i \cap M^-$,

$$v'_i(\mathbf{x}'_i - c) \geq \frac{1}{n} \cdot v'_i(M).$$

Next, we construct a PropMX (resp. PropMX$_0$) allocation for instance I, using $\mathbf{x}'$. For every item $j \notin M^-$, assign it to the agent that gets it in $\mathbf{x}'$, i.e. if $j \in \mathbf{x}'_i$, then assign j to $\mathbf{x}_i$. For every pure bad $j \in M^-$, in sequential order from $j = m^-$ to 1, let i_j denote the agent that gets bad j in $\mathbf{x}'$. We let i_j select her favourite unallocated bad, $c = \arg\max_{1 \leq \ell \leq j} v_{i\ell}$ and assign it to $\mathbf{x}_i$. We show that there is a bijection $f_i : \mathbf{x}_i \to \mathbf{x}'_i$ such that for any bad $e \in \mathbf{x}_i$, we have $v_{ie} \geq v'_{ij}$, where $j = f_i(e)$. Recall that $v'_{ij} \geq v'_{i\ell}$ for all $\ell \leq j$, and that j is the j-th lowest valued pure bad for agent i. Notice that there must exist an unallocated bad e' with value $v_{ie'} \geq v'_{ij}$; if $\sigma_i(j)$ is unallocated at the beginning of step j, then $e' = \sigma_i(j)$. If $\sigma_i(j)$ was allocated, by the pigeonhole principle and because at most $j - 1$ bads have been allocated at the beginning of step j, there must exist an unallocated bad $\sigma_i(k)$, where $k < j$, and we know that $v_{i\sigma_i(k)} \geq v_{i\sigma_i(j)}$. Thus, there must exist an unallocated bad e' with value $v_{ie'} \geq v'_{ij}$. Since e has maximum value for i among unallocated bads under value function v_i, we have $v_{ie} \geq v'_{ij}$. Note that this process also ensures $|\mathbf{x}'_i| = |\mathbf{x}_i|$, $v'_i(M) = v_i(M)$ and $d_i(\mathbf{x}') = d_i(\mathbf{x})$ for all agents $i \in N$.

For every agent $i \in N$, there are two cases.

- Assume i is PropMX (resp. PropMX$_0$) satisfied in $\mathbf{x}'$ because

$$v'_i(\mathbf{x}'_i) + d_i(\mathbf{x}') \geq \frac{1}{n} \cdot v'_i(M).$$

  Notice that $\mathbf{x}'$ and $\mathbf{x}$ are identical with respect to items that are not pure bads, and also, $v_i(\mathbf{x}_i \cap M^-) \geq v'_i(\mathbf{x}'_i \cap M^-)$, which implies $v_i(\mathbf{x}_i) \geq v'_i(\mathbf{x}'_i)$. Since $v'_i(M) = v_i(M)$ and $d_i(\mathbf{x}') = d_i(\mathbf{x})$, we get:

$$v_i(\mathbf{x}_i) + d_i(\mathbf{x}) \geq \frac{1}{n} \cdot v_i(M),$$

  and i is PropMX (resp. PropMX$_0$) satisfied in $\mathbf{x}$.

- Assume now that i is PropMX (resp. PropMX$_0$) satisfied in $\mathbf{x}'$ because $\forall c \in \mathbf{x}'_i \cap M^-$,

$$v'_i(\mathbf{x}'_i - c) \geq \frac{1}{n} \cdot v'_i(M).$$

  We know that $\forall c \in \mathbf{x}_i \cap M^-$, $v_{ic} \geq v'_{if_i(c)}$. Thus, for every $c \in \mathbf{x}_i \cap M^-$:

$$v_i(\mathbf{x}_i - c) = \sum_{\substack{\ell \in \mathbf{x}_i \\ \ell \neq c}} v_{i\ell} \geq \sum_{\substack{\ell' \in \mathbf{x}'_i \\ \ell' \neq f_i(c)}} v'_{i\ell'} = v_i(\mathbf{x}' - f_i(c)) \geq \frac{1}{n} \cdot v_i(M),$$

17

and i is PropMX (resp. PropMX$_0$) satisfied in $\mathbf{x}$.

We conclude that $\mathbf{x}$ is a PropMX (resp. PropMX$_0$) allocation with respect to I. $\qquad \square$

Lemmas 9 and 10 together prove Theorem 4.

5   Revisiting Restricted Mixed Goods

In this section, we revisit the restricted mixed good setting of Section 3 and use the reduction of Lemma 10.

5.1   EFX for Restricted Mixed Goods with IDO Bads

We show that we can obtain EFX allocations for the restricted mixed goods setting with IDO bads.

Theorem 5. *Given a fair division instance of restricted mixed goods and IDO bads, an EFX allocation can be computed in polynomial-time. Furthermore, for the case of binary mixed goods, a EFX$_0$ allocation can be computed in polynomial-time.*

The algorithm is almost identical to Algorithm 2, with the only Phase 3 changed to allocate the IDO bads in the order 1 to m^-, where $v_{i1} \leq v_{i2} \leq \dots v_{im^-}$, for every $i \in N$, like in Algorithm 4. We replace Phase 3 of Algorithm 2 with the following Algorithm 5.

Algorithm 5 EFX for Restricted Mixed Goods & IDO Bads

Input: Instance (N, M, V) with binary mixed goods and identical bads
Output: Allocation $\mathbf{x}$

 Phase 3: Allocating M^- . . . .
1: Order bads in M^- in IDO order $v_{i1} \leq v_{i2} \leq \dots \leq v_{im^-}$
2: **while** $M^- \neq \emptyset$ **do**
3:     Pick smallest $j \in M^-$ according to $\prec$
4:     Let $G^*_{\mathbf{x}} = (N, E)$ be the top-envy-graph defined by $\mathbf{x}$ $\triangleright$ *Def. 2
5:     **while** $\nexists$ any sinks in $G^*_{\mathbf{x}}$ **do**
6:         Let C be a cycle in $G^*_{\mathbf{x}}$
7:         Reallocate bundles according to C $\triangleright$ See text
8:     Let i be a sink in $G^*_{\mathbf{x}}$ $\triangleright$ Def. 2
9:     $\mathbf{x}_i \leftarrow \mathbf{x}_i + j$ $\triangleright$ *Assign j to i
10:     $M^- \leftarrow M^- - j$
11: **return** $\mathbf{x}$

Proof of Theorem 5. Suppose $\mathbf{x}$ is a partial EFX allocation prior to allocating an item $j \in M^-$. Suppose we allocate j to a sink agent i in the top-envy-graph $G^*_{\mathbf{x}}$ to obtain an allocation $\mathbf{x}'$. We will show later that resolving cycles in $G^*_{\mathbf{x}}$ preserves EFX. For sake of contradiction assume $\mathbf{x}'$ is not EFX. Since $\mathbf{x}_h = \mathbf{x}'_h$ for every $h \neq i$, any possible violation of the EFX condition must involve agent i. We claim that i is EFX towards any other agent $h \neq i$ in $\mathbf{x}'$. To see this, note that for any bad $j' \in \mathbf{x}'_i$, $v_i(\mathbf{x}'_i - j') = v_i(\mathbf{x}_i) + v_{ij} - v_{ij'} \geq v_i(\mathbf{x}_i)$ since $j' < j$. Now since i is a sink, $v_i(\mathbf{x}_i) \geq v_i(\mathbf{x}_h)$, thus implying that $v_i(\mathbf{x}'_i - j') \geq v_i(\mathbf{x}_h)$ for every bad $j' \in \mathbf{x}'_i$ and any agent $h \neq i$, showing that i does not EFX-envy any other agent.

18

Now suppose some agent $h \neq i$ EFX-envies i in $\mathbf{x}'$. Since $\mathbf{x}$ was EFX, either (1) $v_h(\mathbf{x}_h) \geq v_h(\mathbf{x}_i - g)$ for every $g \in \mathbf{x}_i$ with $v_{hg} > 0$, or (2) $v_h(\mathbf{x}_h - j') \geq v_h(\mathbf{x}_i)$ for every $j' \in \mathbf{x}_h$ with $v_{hj'} < 0$. If Condition (1) holds, then we also obtain $v_h(\mathbf{x}'_h) = v_h(\mathbf{x}_h) \geq v_h(\mathbf{x}'_i - g)$ for every $g \in \mathbf{x}_i$ with $v_{hg} > 0$, since $\mathbf{x}'_h = \mathbf{x}_h$ and $\mathbf{x}'_i = \mathbf{x}_i + j$ but $v_{hj} \leq 0$. Thus h continues to be EFX towards i in $\mathbf{x}'$. If Condition (2) holds, then, using $v_{hj} \leq 0$ we obtain $v_h(\mathbf{x}'_h - j') = v_h(\mathbf{x}_h - j') \geq v_h(\mathbf{x}_i) \geq v_h(\mathbf{x}'_i)$ for every $j' \in \mathbf{x}'_h$ with $v_{hj'} < 0$, once again showing that h does not EFX-envy i in $\mathbf{x}'$.

Next we argue that resolving any cycles in $G^*_{\mathbf{x}}$ preserves EFX. Suppose we reallocate the bundles according to a top-envy cycle C in $G^*_{\mathbf{x}}$. For any agent i who is not in the cycle, her bundle is not changed by the reallocation. Although other bundles are reallocated, the items in each bundle do not change and thus i does not EFX-envy any other agent. For any agent $i \in C$, she will obtain her best bundle in this partial allocation $\mathbf{x}$, since $G^*_{\mathbf{x}}$ is the top-envy graph, and hence the cycle-swapped allocation is envy-free for agent i. Therefore, resolving any cycles in $G^*_{\mathbf{x}}$ preserves EFX. Thus $\mathbf{x}'$ is EFX.

The case of binary mixed goods follows by the same analysis and the fact that the partial allocation of Algorithm 3, after the end of Phase 2 is EFX_0+PO. □

5.2 PropMX for Restricted Mixed Goods with General Bads

Our final result shows that we can obtain PropMX allocations for the restricted mixed goods setting with general bads. To achieve this, we utilize our reduction in Lemma 10 from general bads instances to IDO bads instances. Note that the allocation is not guaranteed to be Pareto Optimal.

Theorem 6. *Given a fair division instance of restricted mixed goods, a PropMX allocation can be computed in polynomial-time. Furthermore, for the case of binary mixed goods, a PropMX_0 allocation can be computed in polynomial-time.*

Proof. Given any fair division instance of mixed goods with restricted valuations $I = (N, M, V)$, we first turn it into a new fair division instance of mixed goods with restricted valuations and IDO bads $I' = (N, M, V')$, via the reduction of Lemma 10 on the set of pure bads M^-. Notice that, as long as the algorithm for IDO bads that is executed on the I' returns an allocation $\mathbf{x}$ that is PropMX+PO, we know that any item $j \notin M^-$ has been allocated in $\mathbf{x}$ to an agent that has non-negative value for it. Thus, since the reduction of Lemma 10 does not distinguish between M^+ and M^0, it still holds in this setting as well.

Next, notice that the partial allocation of Algorithm 2 after the end of Phase 2 has allocated all items $j \notin M^-$. Also, it is EFX+PO, and thus by Lemma 1, it is PropMX+PO. Therefore, one can use the partial allocation of pure bads of Phase 3 for I' as a guide on how to allocate the items in M^- and maintain PropMX, as Lemma 10 describes.

The case of binary mixed goods follows by the same analysis and the fact that the partial allocation of Algorithm 3, after the end of Phase 2 is $\text{PropMX}_0+\text{PO}$. □

6 Discussion

In this paper, we study the fair and efficient allocation of an indivisible mixed manna. We measured efficiency through Pareto-optimality, and fairness through EFX and PropMX, where PropMX combines PropM for goods and PropX for bads. We obtained polynomial time algorithms to find allocations that satisfy a mix of these guarantees for several classes of instances, namely separable, restricted mixed goods, and binary mixed goods.

Much like EFX+PO, settling the existence of PropM+PO allocations is a challenging open problem even for goods manna. One can also ask whether allocations satisfying more demanding

fairness notions than PropM exist, both for goods and the more general mixed manna setting. Lastly, the existence of weighted-PropMX allocations for agents with unequal entitlements is an interesting direction for future work.

References

[1] Martin Aleksandrov and Toby Walsh. Greedy algorithms for fair division of mixed manna, 2019. `arXiv:1911.11005`.

[2] Martin Aleksandrov and Toby Walsh. Two algorithms for additive and fair division of mixed manna, 2020. `arXiv:2007.04129`.

[3] Georgios Amanatidis, Georgios Birmpas, Aris Filos-Ratsikas, Alexandros Hollender, and Alexandros A. Voudouris. Maximum Nash welfare and other stories about EFX. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, (IJCAI)*, pages 24–30, 2020.

[4] Haris Aziz, Ioannis Caragiannis, Ayumi Igarashi, and Toby Walsh. Fair allocation of indivisible goods and chores. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, IJCAI'19, page 53–59. AAAI Press, 2019.

[5] Haris Aziz, Hervé Moulin, and Fedor Sandomirskiy. A polynomial-time algorithm for computing a pareto optimal and almost proportional allocation. *CoRR*, abs/1909.00740, 2019. URL: `http://arxiv.org/abs/1909.00740`, `arXiv:1909.00740`.

[6] Artem Baklanov, Pranav Garimidi, Vasilis Gkatzelis, and Daniel Schoepflin. Propm allocations of indivisible goods to multiple agents. *CoRR*, abs/2105.11348, 2021. URL: `https://arxiv.org/abs/2105.11348`, `arXiv:2105.11348`.

[7] Nikhil Bansal and Maxim Sviridenko. The santa claus problem. In *Proceedings of the Thirty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '06, page 31–40, New York, NY, USA, 2006. Association for Computing Machinery. `doi:10.1145/1132516.1132522`.

[8] Siddharth Barman and Sanath Kumar Krishnamurthy. Approximation algorithms for maximin fair division. *ACM Trans. Econ. Comput.*, 8(1), March 2020. `doi:10.1145/3381525`.

[9] Siddharth Barman, Sanath Kumar Krishnamurthy, and Rohit Vaish. Finding fair and efficient allocations. In *Proceedings of the 19th ACM Conference on Economics and Computation (EC)*, pages 557–574, 2018.

[10] Siddharth Barman, Sanath Kumar Krishnamurthy, and Rohit Vaish. Greedy algorithms for maximizing Nash social welfare. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, page 7–13, 2018.

[11] Umang Bhaskar, A. R. Sricharan, and Rohit Vaish. On approximate envy-freeness for indivisible chores and mixed resources. *CoRR*, abs/2012.06788, 2020. URL: `https://arxiv.org/abs/2012.06788`, `arXiv:2012.06788`.

[12] Anna Bogomolnaia, Hervé Moulin, Fedor Sandomirskiy, and Elena Yanovskaya. Competitive division of a mixed manna. *Econometrica*, 85(6):1847–1871, 2017. URL: `http://www.jstor.org/stable/44955184`.

[13] 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, Mar 2016. `doi:10.1007/s10458-015-9287-3`.

[14] Ioannis Caragiannis, David Kurokawa, Hervé Moulin, Ariel D. Procaccia, Nisarg Shah, and Junxing Wang. The unreasonable fairness of maximum Nash welfare. In *Proceedings of the 17th ACM Conference on Economics and Computation (EC)*, page 305–322, 2016.

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

[16] Andreas Darmann and Joachim Schauer. Maximizing Nash product social welfare in allocating indivisible goods. *SSRN Electronic Journal*, 247, 01 2014.

[17] D.K. Foley. Resource allocation and the public sector. *Yale Economic Essays*, 7(1):45–98, 1967.

[18] Jugal Garg and Peter McGlaughlin. Computing competitive equilibria with mixed manna. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '20, page 420–428, Richland, SC, 2020. International Foundation for Autonomous Agents and Multiagent Systems.

[19] Jugal Garg and Aniket Murhekar. Computing fair and efficient allocations with few utility values. *In Proc. 14th Symp. Algorithmic Game Theory (SAGT)*, 2021. To appear.

[20] Jugal Garg, Aniket Murhekar, and John Qin. Fair and efficient allocations of chores under bivalued preferences. *CoRR*, abs/2110.09601, 2021. URL: `https://arxiv.org/abs/2110.09601`, `arXiv:2110.09601`.

[21] Daniel Halpern, Ariel D. Procaccia, Alexandros Psomas, and Nisarg Shah. Fair division with binary valuations: One rule to rule them all. In Xujin Chen, Nikolai Gravin, Martin Hoefer, and Ruta Mehta, editors, *Web and Internet Economics*, pages 370–383, Cham, 2020. Springer International Publishing.

[22] Xin Huang and Pinyan Lu. An algorithmic framework for approximating maximin share allocation of chores. In *Proceedings of the 22nd ACM Conference on Economics and Computation*, EC '21, page 630–631, New York, NY, USA, 2021. Association for Computing Machinery. `doi:10.1145/3465456.3467555`.

[23] 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*, EC '21, page 683–684, New York, NY, USA, 2021. Association for Computing Machinery. `doi:10.1145/3465456.3467553`.

[24] Bo Li, Yingkai Li, and Xiaowei Wu. Almost proportional allocations for indivisible chores. *CoRR*, abs/2103.11849, 2021. URL: `https://arxiv.org/abs/2103.11849`, `arXiv:2103.11849`.

[25] R. J. Lipton, E. Markakis, E. Mossel, and A. Saberi. On approximately fair allocations of indivisible goods. In *Proceedings of the 5th ACM Conference on Electronic Commerce (EC)*, pages 125–131, 2004.

[26] Hervé Moulin. Fair division in the internet age. *Annual Review of Economics*, 11(1):407–441, 2019. arXiv:https://doi.org/10.1146/annurev-economics-080218-025559, doi:10.1146/annurev-economics-080218-025559.

[27] Aniket Murhekar and Jugal Garg. On fair and efficient allocations of indivisible goods. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(6):5595–5602, May 2021.

[28] Benjamin Plaut and Tim Roughgarden. Almost envy-freeness with general valuations. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, page 2584–2603, 2018.

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

[30] H. Steinhaus. Sur la division pragmatique. *Econometrica*, 17(1):315–319, 1949.

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

A Counterexamples

A.1 Non-Existence of PropX for Goods

In this section, we present an example that shows PropX allocations need not exist for goods, even for the simpler setting of identical valuations. The example is due to Aziz, Moulin and Sandomirskiy [26, 5]. We present it here as well for completeness.

Consider three agents a and b, and five goods g_1, g_2, g_3, g_4 and g_5. The agents' valuations for the goods are the following:

	g_1	g_2	g_3	g_4	g_5
a	3	3	3	3	1
b	3	3	3	3	1
c	3	3	3	3	1

Notice that the proportional share for every agent is $\frac{13}{3}$ and that in any balanced allocation one agent gets two items of value 3, one agent gets one item of value 3 along with g_5, which has value 1 and one agent gets only one item of value 3. Without loss of generality, let $\mathbf{x}_a = \{g_1, g_4\}, \mathbf{x}_b = \{g_2, g_5\}$ and $\mathbf{x}_c = \{g_3\}$. Notice that $v_c(\mathbf{x}_c) = 3$, and even if c receives g_5, we have $v_c(\mathbf{x}_c) + v_{c5} = 4 < \frac{13}{3}$. Thus, there exists no PropX allocation for this instance.

A.2 Non-Existence of PropM$_0$+PO

In this section, we present an example that shows PropM$_0$+PO allocations need not always exist, even for the simpler setting of restricted goods and no bads. Consider two agents a and b, and three goods g_1, g_2 and g_3. The agents' valuations for the goods are the following:

	g_1	g_2	g_3
a	1	0	2
b	0	1	2

Notice that for any PO allocation $\mathbf{x}$, we have $g_1 \in \mathbf{x}_a$ and $g_2 \in \mathbf{x}_b$. Also notice that $v_a(M) = v_b(M) = \frac{3}{2}$. However, either a or b will not receive g_3. Assume without loss of generality that $g_3 \in \mathbf{x}_a$. We have $d_b(\mathbf{x}) = 0$, since the maximin item for agent b is g_1, and thus $v_b(\mathbf{x}_b) + d_b(\mathbf{x}) = 1 + 0 < 3/2$. Therefore, there exists no PropM$_0$+PO allocation for this instance.