Computing Fair and Efficient Allocations with Few Utility Values[⋆,⋆⋆]

Jugal Garg[a], Aniket Murhekar[b,*]

[a] *University of Illinois at Urbana-Champaign*
[b] *University of Illinois at Urbana-Champaign*

Abstract

We study the problem of allocating indivisible goods among agents with additive valuations in a fair and efficient manner, when agents have *few* utility values for the goods. We consider the compelling fairness notion of envy-freeness up to any good (EFX) in conjunction with Pareto-optimality (PO). Amanatidis et al. [1] showed that when there are at most two utility values, an EFX allocation can be computed in polynomial-time. We improve this result by showing that for such instances an allocation that is EFX and PO can be computed in polynomial-time. This is the first class apart from identical or binary valuations, for which EFX+PO allocations are shown to exist and are polynomial-time computable. In contrast, we show that when there are three utility values, EFX+PO allocations need not exist, and even deciding if EFX+PO allocations exist is NP-hard.

Our techniques allow us to obtain similar results for the fairness notion of equitability up to any good (EQX) together with PO. We show that for instances with two positive values an EQX+PO allocation can be computed in polynomial-time, and deciding if an EQX+PO allocation exists is NP-hard when there are three utility values.

We also study the problem of maximizing Nash welfare (MNW), and show that our EFX+PO algorithm returns an allocation that approximates the MNW to a factor of 1.061 for two valued instances, in addition to being EFX+PO. In contrast, we show that for three valued instances, computing an MNW allocation is APX-hard. Finally, we we give a polynomial-time algorithm for computing an MNW allocation for two-valued instances where the ratio of the two values is greater than a certain threshold.

Keywords: Fair and efficient allocation, EFX, Nash welfare, EQX

[⋆]Supported by NSF Grant CCF-1942321 (CAREER)
[⋆⋆]A preliminary conference version of this work has appeared in [2]
[*]Corresponding author
Email addresses: `jugal@illinois.edu` (Jugal Garg), `aniket2@illinois.edu` (Aniket Murhekar)

1. Introduction

The problem of fair division was formally introduced by Steinhaus [3], and has since been extensively studied in various fields, including economics and computer science [4, 5]. It concerns allocating resources (goods) to agents in a *fair* and *efficient* manner, and has various practical applications such as rent division, division of inheritance, course allocation, and government auctions.

Arguably, the most popular notion of fairness is *envy-freeness* (EF) [6, 7], which requires that every agent prefers their own bundle of goods to that of any other. However in the case of *indivisible* goods, EF allocations need not even exist (consider allocating 1 good among 2 agents). This motivated the study of its relaxations. One such relaxation is *envy-freeness up to one good* (EF1) allocation, defined by Budish [8], where every agent prefers their own bundle to the bundle of any other agent after removing *some* good from the other agent's bundle. It is well-known that an EF1 allocation always exists and is polynomial-time computable [9]. However, an EF1 allocation may be unsatisfactory because it allows the removal of the *most valuable* good from the other agent's bundle, which might be the main reason for huge envy to exist in the first place. Therefore, stronger fairness notions are desirable in many settings.

A stronger notion is called *envy-free up to any good* (EFX), defined by Caragiannis et al. [10], which requires every agent to prefer their bundle over the bundle of any other agent after removing *any* good from the other agent's bundle. Clearly, any allocation that is EFX is also EF1, but not vice-versa. The existence of EFX allocations is known for identical valuations [11], and was recently shown for 3 agents with additive valuations [12].[1] At the same time, we want the output allocation to be efficient because a fair allocation by itself may be highly inefficient. Consider for example two agents A_1 and A_2 and 2 goods g_1 and g_2 where A_i values only g_i and does not value the other good. The allocation in which g_1 is assigned to A_2 and g_2 is assigned to A_1 is clearly EFX. However both agents get zero utility, which is highly inefficient. The allocation in which g_i is assigned to A_i is more desirable since it is both fair as well as 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. A stronger notion called fractional Pareto optimality (fPO) requires that no other fractional allocation makes an agent better off without making someone else worse off. Every fPO allocation is therefore PO, but not vice-versa (see Appendix A for an example). Another reason to prefer fPO allocations over PO allocations is that the former admit efficient verification while the latter do not: given an allocation, it can be checked in polynomial time if it is fPO [15], whereas checking if an allocation is PO

[1]Settling the (non-)existence of EFX allocations is considered the biggest open question in fair division [13]; see [14] and references therein for recent progress on this problem.

is coNP-complete [16]. Hence if a centralized entity responsible for allocating resources claims the allocation is fPO, each agent can individually verify that this is indeed the case; in contrast such a check is not efficiently possible if the guarantee is only PO.

An important question is whether the notions of fairness (EF1 or EFX) can be achieved in conjunction with the efficiency notions (PO or fPO). Further, if yes, then whether they can be computed in polynomial-time. For this, the concept of *Nash welfare* provides a partial answer. The Nash welfare is defined as the geometric mean of the agents' utilities, and by maximizing it we achieve a tradeoff between efficiency and fairness. Caragiannis et al. [10] showed that the maximum Nash welfare (MNW) allocations are EF1 and PO under additive valuations. However, the problem of computing an MNW allocation is APX-hard [17] (hard to approximate). Bypassing this barrier, Barman et al. [15] devised a pseudo-polynomial-time algorithm that computes an EF1+PO allocation, and Garg and Murhekar [18] showed that an EF1+fPO allocation can be computed in pseudo-polynomial time. For the special case of binary additive valuations an MNW allocation is EFX+fPO, and is known to be polynomial-time computable [19, 20].

1.1. Our Contributions

In this work, we obtain several novel results on the notions of EFX, EQX, PO, and MNW, especially for instances in which agents have few values for the goods. A fair division instance is called k-valued if values that agents have for the goods belong a set of size k.

EFX. Amanatidis *et al.* [1] showed that for 2-valued instances any MNW allocation is EFX+PO, but left open the question of whether it can be computed in polynomial-time. They presented a polynomial-time algorithm which computes an EFX allocation for 2-valued instances, however, the outcome of their algorithm need not be PO (see Appendix A for an example). In this work, we show EFX+fPO allocations always exist for 2-valued instances and can be computed in polynomial-time.[2] Further, apart from the classes of identical valuations and binary valuations, this is the first class for which EFX+PO allocations exist and can be computed in polynomial-time.

In general, EFX+PO allocations are not guaranteed to exist [11]. We therefore ask the natural question: what is the complexity of *checking* if an instance admits an EFX+PO allocation? We show that this problem is NP-hard, somewhat surprisingly, even for 3-valued instances.

EQX. Our techniques allow us to obtain similar results for the fairness notion of *equitability up to any good* [21, 22]. An allocation is said to be EQX (resp. EQ1) if the utility an agent gets from her bundle is no less than the utility any

[2]Our results extend to the much broader class where there are two values $\{a_i, b_i\}$ per agent, but a_i/b_i is the same across agents.

other agent gets after removing *any* (resp. some) good from their bundle. We show that for positive 2-valued instances, an EQX+PO allocation can be computed in polynomial-time, and in contrast, even checking existence of EQX+PO allocations for 3-valued instances is NP-hard.

MNW. It is easy to see that an MNW allocation is PO. For 2-valued instances we show the stronger guarantee of fPO. Next, we show our EFX+PO algorithm returns an allocation that approximates the maximum Nash welfare to a factor of 1.061 in addition to being EFX and PO. This guarantee is better than the best known 1.45-approximation algorithm of [15] for the MNW problem.

Amanatidis *et al.* [1] showed that computing an MNW allocation is NP-hard for 3-values instances, which, as they remark "extends the hardness aspect, but not the inapproximability, of the result of Lee [17] for 5-valued instances", who had shown that MNW is NP-hard to approximate within a factor of 1.00008. In our work, we extend the inapproximability aspect too, and show that it is NP-hard to approximate the MNW to a factor of 1.00019, even for 3-valued instances, which is better than Lee's result. Finally, we present a polynomial-time algorithm for computing an MNW allocation for 2-valued instances where the ratio of the two values is larger than $m-n$, where m is the number of goods and n the number of agents.

Thus, for the problems of computing (i) EFX+PO, (ii) EQX+PO, and (iii) MNW allocations, our work improves the state-of-the-art and also crucially pinpoints the boundary between tractable and intractable cases.

1.2. Other Related Work

Barman *et al.* [15] showed that for n agents and m goods, an EF1+PO allocation can be computed in time $\mathsf{poly}(n, m, v_{max})$, where v_{max} is the maximum utility value. Their algorithm first perturbs the values to a desirable form, and then computes an EF1+fPO allocation for the perturbed instance, which for a small-enough perturbation is EF1+PO for the original instance. Their approach is via *integral market-equilibria*, which guarantees fPO at every step. Our algorithm uses a similar approach, with one main difference being that we do not need to consider any approximate instance and can work directly with the given values. The outcome of our algorithm is EFX+fPO, which beats the guarantee of EF1+PO.

Another key difference is the *run-time analysis*: our arguments show termination in $\mathsf{poly}(n, m)$ time for 2-valued instances, even when $v_{max} = 2^{\Omega(n+m)}$, whereas the analysis of Barman *et al.* only shows a $\mathsf{poly}(n, m, v_{max})$ time bound, even for 2-valued instances.

Garg and Murhekar [18] showed that an EF1+fPO allocation can be computed in $\mathsf{poly}(n, m, v_{max})$-time, by using integral market-equilibria. They also showed that an EF1+fPO allocation can be computed in $\mathsf{poly}(n, m)$-time for k-valued instances where k is a constant, however they do not show that the allocation returned by their algorithm is EFX for 2-valued instances.

Freeman *et al.* [22] showed that EQ1+PO allocations can be computed in pseudo-polynomial time for instances with positive values. They also show that the leximin solution, i.e., the allocation that maximizes the minimum utility, and subject to this, maximizes the second minimum utility, and so on; is EQX+PO. However, as remarked in [11], computing a leximin solution is intractable.

Barman *et al.* [20] showed that for identical valuations, any EFX allocation provides a 1.061-approximation to the MNW. Recently, [23] showed that for 2-valued instances where the two values are $\{p, q\}$ for $p, q \in \mathbb{N}$ and p divides q, an MNW allocation is polynomial-time computable. We note that the class we study for an exact algorithm (2-valued instances where the ratio of the two values is at least $m - n$) is orthogonal to their class. Moreover, for general 2-value instances, their algorithm achieves an approximation ratio of at most 1.0345, whereas we obtain a ratio of 1.061. They also showed 1.000015-factor APX-hardness when $p \geq 3$ and p, q are co-prime. Garg *et al.* [24] showed a 1.069-hardness of approximating MNW, although for 4-valued instances.

Instances with few values have been widely considered in the fair division literature: for instance Golovin [25] presents approximation algorithms and hardness results for computing max-min fair allocations in 3-valued instances; Aziz *et al.* [26] show PO is efficiently verifiable for 2-valued instances and coNP-hard for 3-valued instances; Aziz [27], and Vazirani and Yannakakis [28] study the Hylland-Zeckhauser scheme for probabilistic assignment of goods in 2-valued instances; Bogomolnaia and Moulin [29] study matching problems with 2-valued (dichotomous) preferences; Bliem *et al.* [30] study fixed-parameter tractability for computing EF+PO allocations with parameter $n + z$, where z is the number of values; recently, Garg *et al.* [31] showed EF1+PO allocations of 2-valued *chores* can be computed in polynomial time; and Garg *et al.* [32] study leximin assignments of papers ranked by reviewers on a small scale, in particular they present an efficient algorithm for 2 ranks, i.e., "high or low interest" and show NP-hardness for 3 ranks. More generally, such instances have been studied in resource allocation contexts, including makespan minimization with 2 or 3 job sizes [33, 34].

2. Preliminaries

Problem setting. 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 the 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 $v_i : M \to \mathbb{Z}_{\geq 0}$ is specified by m numbers $v_{ij} \in \mathbb{Z}_{\geq 0}$, one for each good $j \in M$, which denotes the value agent i has for good j. We assume that the valuation functions are additive, that is, for every agent $i \in N$, and for $S \subseteq M$, $v_i(S) = \sum_{j \in S} v_{ij}$. Further, we assume that for every good j, there is some agent i such that $v_{ij} > 0$. Note that we can in general work with rational values without loss of generality, since they can be scaled to make them integral, and the efficiency and fairness guarantees we consider are scale-invariant.[3]

[3]The properties of EFX, PO, and Nash welfare are invariant under scaling, while EQX is not scale-invariant in general. However, in our algorithms this is not an issue since we only

We call a fair division instance (N, M, V) a t-valued instance if $|\{v_{ij} : i \in N, j \in M\}| = t$. The class of 2-valued instances is made up of two disjoint fragments: binary instances, where all values $v_{ij} \in \{0, 1\}$; and $\{a, b\}$-instances, where all values $v_{ij} \in \{a, b\}$ for $a, b \in \mathbb{Z}_{>0}$. An important subclass of 3-valued instances is the $\{0, a, b\}$ class, wherein all values $v_{ij} \in \{0, a, b\}$ for $a, b \in \mathbb{Z}_{>0}$.

Allocation. An (integral) *allocation* $\mathbf{x}$ of goods to agents is a n-partition $(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ of the goods, where $\mathbf{x}_i \subseteq M$ is the bundle of goods allotted to agent i, who gets a total value of $v_i(\mathbf{x}_i)$. A *fractional allocation* $\mathbf{x} \in [0, 1]^{n \times m}$ is a fractional assignment of the goods to agents such that for each good $j \in M$, $\sum_{i \in N} x_{ij} = 1$. Here, $x_{ij} \in [0, 1]$ denotes the fraction of good j allotted to agent i. In a fractional allocation $\mathbf{x}$, an agent i receives a value of $v_i(\mathbf{x}_i) = \sum_{j \in M} v_{ij} x_{ij}$.

Fairness notions. An allocation $\mathbf{x}$ is said to be:

1. *Envy-free up to one good* (EF1) if for all $i, h \in N$, there exists a good $j \in \mathbf{x}_h$ s.t. $v_i(\mathbf{x}_i) \geq v_i(\mathbf{x}_h \setminus \{j\})$.
2. *Envy-free up to any good* (EFX) if for all $i, h \in N$ and for all goods $j \in \mathbf{x}_h$ we have $v_i(\mathbf{x}_i) \geq v_i(\mathbf{x}_h \setminus \{j\})$.
3. *Equitable up to one good* (EQ1) if for all $i, h \in N$, there exists a good $j \in \mathbf{x}_h$ s.t. $v_i(\mathbf{x}_i) \geq v_h(\mathbf{x}_h \setminus \{j\})$.
4. *Equitable up to any good* (EQX) if for all $i, h \in N$ and for all goods $j \in \mathbf{x}_h$ we have $v_i(\mathbf{x}_i) \geq v_h(\mathbf{x}_h \setminus \{j\})$.

Pareto-optimality. An allocation $\mathbf{y}$ dominates an allocation $\mathbf{x}$ if for all $i \in N$, $v_i(\mathbf{y}_i) \geq v_i(\mathbf{x}_i)$ 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 allocation dominates it. Further, an allocation is said to be *fractionally* Pareto-optimal (fPO) if no fractional allocation dominates it. Thus, any fPO allocation is PO, but not vice-versa (see Appendix A for an example).

Nash welfare. The Nash welfare of an allocation $\mathbf{x}$ is given by $\mathsf{NW}(\mathbf{x}) = \left(\Pi_{i \in N} v_i(\mathbf{x}_i)\right)^{1/n}$. An allocation that maximizes the NW is called an MNW allocation or Nash optimal allocation. An allocation $\mathbf{x}$ *approximates* the MNW to a factor α if $\alpha \cdot \mathsf{NW}(\mathbf{x}) \geq \mathsf{NW}(\mathbf{x}^*)$, where $\mathbf{x}^*$ is an MNW allocation.

Fisher markets. A Fisher market or a *market instance* is a tuple (N, M, V, e), where $N = [n]$ is a set of $n \in \mathbb{N}$ agents, $M = [m]$ is a set of $m \in \mathbb{N}$ divisible goods, $V = \{v_1, \ldots, v_n\}$ is a set of additive (linear) utility functions, and $e = \{e_1, \ldots, e_n\}$ is the set of agents' budgets, where each $e_i \geq 0$. In this model, agents can fractionally share goods. Each agent aims to obtain a bundle of goods that maximizes her total value subject to her budget constraint.

Given a (fractional) allocation $\mathbf{x}$ with a price vector $\mathbf{p}$, the *spending* of an agent i under $(\mathbf{x}, \mathbf{p})$ is given by $\mathbf{p}(\mathbf{x}_i) = \sum_{j \in M} p_j x_{ij}$. We define the *bang-per-buck* ratio α_{ij} of good j for an agent i as $\alpha_{ij} = v_{ij}/p_j$, and the *maximum*

uniformly scale the valuations of all agents, which preserves EQX.

bang-per-buck (MBB) ratio as $\alpha_i = \max_j \alpha_{ij}$. We define $\text{MBB}_i = \{j \in M : c_{ij}/p_j = \alpha_i\}$, called the *MBB-set*, to be the set of goods that give MBB to agent i at prices $\mathbf{p}$. We say $(\mathbf{x}, \mathbf{p})$ is *'on MBB'* if for all agents i and goods j, $x_{ij} > 0 \Rightarrow j \in \text{MBB}_i$. For integral $\mathbf{x}$, this means that $\mathbf{x}_i \subseteq \text{MBB}_i$ for all $i \in N$.

A *market equilibrium* or *market outcome* is a (fractional) allocation $\mathbf{x}$ of the goods to the agents and set of prices $\mathbf{p}$ of the goods satisfying the following:

- the market clears, i.e., all goods are fully allocated. Thus, $\forall j \in M$, $\sum_{i\in N} x_{ij} = 1$,
- each agent spends their budget, i.e., $\forall i \in N$, $\mathbf{p}(\mathbf{x}_i) = \sum_{j\in M} x_{ij}p_j = e_i$, and,
- agents only receive goods that give them maximum bang-per-buck, i.e., $(\mathbf{x}, \mathbf{p})$ is on MBB.

Market equilibria are an important tool in computing fair and efficient allocations because of their remarkable fairness and efficiency properties; see e.g., [35, 15, 36, 37, 38, 22, 39].

Theorem 1. *(First Welfare Theorem [40]) Let $(\mathbf{x}, \mathbf{p})$ be a equilibrium of a Fisher market $\mathcal{M}$. Then $\mathbf{x}$ is fractionally Pareto-optimal.*

Given an allocation $\mathbf{x}$ for a fair division instance (N, M, V) and a vector of prices $\mathbf{p}$ for the goods such that $(\mathbf{x}, \mathbf{p})$ is on MBB, one can define an associated Fisher market instance $\mathcal{M} = (N, M, V, e)$ by setting $e_i = \mathbf{p}(\mathbf{x}_i)$. It is easy to see that $(\mathbf{x}, \mathbf{p})$ is a market equilibrium of $\mathcal{M}$. Hence Theorem 1 implies:

Corollary 2. *Given an allocation $\mathbf{x}$ and prices $\mathbf{p}$ of the goods such that $(\mathbf{x}, \mathbf{p})$ is on MBB, the allocation $\mathbf{x}$ is fPO.*

3. Computing EFX+PO allocations

We first study the problem of computing an EFX+PO allocation for t-valued instances when $t \in \{2, 3\}$. We show that EFX+PO allocations can be computed in polynomial-time for 2-valued instances, and in contrast, computing such allocations for 3-valued instances is NP-hard.

3.1. EFX+PO allocations for 2-valued instances

We consider $\{a, b\}$-instances, as it is known EFX+PO allocations can be efficiently computed for binary instances. We remark that while the allocation returned by the algorithm Match&Freeze of Amanatidis *et al.*, [1] for $\{a, b\}$-instances is EFX, it need not be PO (see Appendix A). We improve this result by showing that:

Theorem 3. *Given a fair division $\{a, b\}$-instance $I = (N, M, V)$, an allocation that is EFX, fPO and approximates the maximum Nash welfare to a factor of 1.061 can be computed in polynomial-time.*

We prove this theorem by showing that Algorithm 1 computes such an allocation. We first define some relevant terms, including the concept of price envy-freeness introduced by Barman *et al.* [15]. A market outcome $(\mathbf{x}, \mathbf{p})$ is said to be *price envy-free up to one good* (pEF1) if for all agents $i, h \in N$ there is a good $j \in \mathbf{x}_h$ such that $\mathbf{p}(\mathbf{x}_i) \geq \mathbf{p}(\mathbf{x}_h \setminus \{j\})$. Similarly, we say it is pEFX if for all agents $i, h \in N$, and for all goods $j \in \mathbf{x}_h$ it holds that $\mathbf{p}(\mathbf{x}_i) \geq \mathbf{p}(\mathbf{x}_h \setminus \{j\})$. For market outcomes on MBB, the pEFX condition implies the EFX condition:

Lemma 1. *Let* $(\mathbf{x}, \mathbf{p})$ *be an integral market outcome on MBB. Then* $\mathbf{x}$ *is fPO. If* $(\mathbf{x}, \mathbf{p})$ *is pEFX, then* $\mathbf{x}$ *is EFX.*

Proof. The fact that $\mathbf{x}$ is fPO follows from Corollary 2. Since $(\mathbf{x}, \mathbf{p})$ is pEFX, for all pairs of agents $i, h \in N$, and all goods $j \in \mathbf{x}_h$ it holds that $\mathbf{p}(\mathbf{x}_i) \geq \mathbf{p}(\mathbf{x}_h \setminus \{j\})$. Since $(\mathbf{x}, \mathbf{p})$ is on MBB, $\mathbf{x}_i \subseteq \text{MBB}_i$. Let α_i be the MBB-ratio of i at the prices $\mathbf{p}$. By definition of MBB, $v_i(\mathbf{x}_i) = \alpha_i \mathbf{p}(\mathbf{x}_i)$, and $v_i(\mathbf{x}_h \setminus \{j\}) \leq \alpha_i \mathbf{p}(\mathbf{x}_h \setminus \{j\})$, for every $j \in \mathbf{x}_h$. Combining these, we get that $\mathbf{x}$ is EFX. □

Given a market outcome $(\mathbf{x}, \mathbf{p})$, we define the MBB graph to be a bipartite graph $G = (N, M, E)$ where for an agent i and good j, $(i, j) \in E$ iff $j \in \text{MBB}_i$. Further, an edge (i, j) is called an *allocation edge* if $j \in \mathbf{x}_i$, otherwise it is called an *MBB* edge.

For agents $i_0, \ldots, i_\ell$ and goods $j_1, \ldots, j_\ell$, a path $P = (i_0, j_1, i_1, j_2, \ldots, j_\ell, i_\ell)$ in the MBB graph, where for all $1 \leq \ell' \leq \ell$, $j_{\ell'} \in \mathbf{x}_{i_{\ell'}} \cap \text{MBB}_{i_{\ell'-1}}$, is called a *special* path. We define the *level* $\lambda(h; i_0)$ of an agent h w.r.t. i_0 to be half the length of the shortest special path from i_0 to h, and to be n if no such path exists. A path $P = (i_0, j_1, i_1, j_2, \ldots, j_\ell, i_\ell)$ is an *alternating path* if it is special, and if $\lambda(i_0; i_0) < \lambda(i_1; i_0) < \cdots < \lambda(i_\ell; i_0)$, i.e. the path visits agents in increasing order of their level w.r.t. i_0. Further, the edges in an alternating path alternate between allocation edges and MBB edges. Typically, we consider alternating paths starting from the *least spender* (LS) agent, who is an agent i with minimum $\mathbf{p}(\mathbf{x}_i)$ with ties broken lexicographically. We use alternating paths to reduce the pEF1-envy of agent i towards agent h by transferring goods along the alternating path $(i_0, j_1, i_1, j_2, \ldots, j_\ell, i_\ell)$, i.e., by transferring j_ℓ to $i_{\ell-1}$, etc. The structure of an alternating path ensures that transfers are done along MBB edges, implying that resulting allocations remain on MBB, and hence preserve the fPO property of the allocation. Figure 1 provides an example of an alternating path.

Definition 1 (Component C_i of a least spender i)**.** For a least spender i, define C_i^ℓ to be the set of all goods and agents which lie on alternating paths of length ℓ starting from i. Call $C_i = \bigcup_\ell C_i^\ell$ the *component* of i, i.e., the set of all goods and agents reachable from i through alternating paths.

We now describe Algorithm 1. Let $k = a/b > 1$. Let us first scale the valuations to $\{1, k\}$ since both properties EFX and fPO are scale-invariant. The algorithm starts with a welfare maximizing integral allocation $(\mathbf{x}, \mathbf{p})$, where $p_j = v_{ij}$ if $j \in \mathbf{x}_i$. The algorithm then explores if there is an alternating

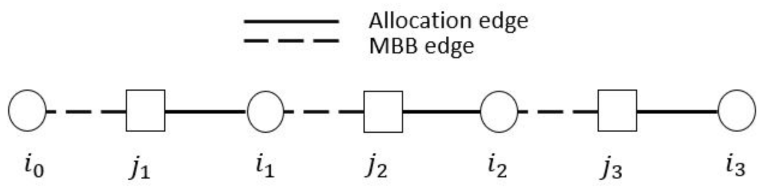

Figure 1: Example of an alternating path

Algorithm 1 EFX+fPO allocation for $\{a, b\}$-instances

```
Input: Fair division $\{a, b\}$-instance $(N, M, V)$
Output: An integral allocation $\mathbf{x}$
 1: Scale values to $\{1, k\}$, where $k = a/b > 1$.
 2: $(\mathbf{x}, \mathbf{p}) \leftarrow$ Integral welfare-maximizing market allocation, where $p_j = v_{ij}$ for $j \in \mathbf{x}_i$.
 3: Let $i \in \mathsf{argmin}_{h \in N} \mathbf{p}(\mathbf{x}_h)$ be the least spender
 4: while there is an alternating path $(i, j_1, i_1, \ldots, j_\ell, i_\ell)$, s.t. $\mathbf{p}(\mathbf{x}_{i_\ell} \setminus \{j_\ell\}) > \mathbf{p}(\mathbf{x}_i)$ do
 5:     Transfer $j_\ell$ from $i_\ell$ to $i_{\ell-1}$, update allocation $\mathbf{x}$
 6:     Update the LS $i \in \mathsf{argmin}_{h \in N} \mathbf{p}(\mathbf{x}_h)$
 7: if $\forall$ agents $h \notin C_i$, and $\forall j \in \mathbf{x}_h : \mathbf{p}(\mathbf{x}_h \setminus \{j\}) \leq \mathbf{p}(\mathbf{x}_i)$ then return $\mathbf{x}$    $\triangleright$ pEFX condition satisfied for all agents not in component of LS, defined in Def.1
 8: else
 9:     Raise prices of goods in $C_i$ by a multiplicative factor of $k$
10:     Repeat from Line 3
```

path $P = (i = i_0, j_1, i_1, \cdots, j_\ell, i_\ell = h)$, where i is the LS agent, such that $\mathbf{p}(\mathbf{x}_h \setminus \{j_\ell\}) > \mathbf{p}(\mathbf{x}_i)$, i.e., an alternating path along which the pEF1 condition is violated for the LS agent. We call any such agent h who owns some good j such that the pEF1 condition is not satisfied by the LS with respect to good j, a pEF1-violator. When such a path is encountered, the algorithm *transfers* j_ℓ from h to $i_{\ell-1}$. This process is repeated from Line 3 to account for a possible change in the LS, until there is no such path in the component C_i of the LS agent. Suppose there is some agent $h \notin C_i$ for which the pEFX condition is not satisfied with respect to the LS, then the algorithm *raises the prices* of all goods in the component of the LS agent by a factor of k, and the algorithm proceeds once again from Line 3.

The proof of Theorem 3 relies on Lemmas 1-6. We state and sketch some proofs below, and defer full proofs to Appendix B. We first show that we can re-scale prices to $\{1, k\}$.

Lemma 2. *For every outcome* $(\mathbf{x}, \mathbf{p})$ *constructed during the run of Algorithm 1,* *there exists a set of prices* $\mathbf{q}$ *such that* $(\mathbf{x}, \mathbf{q})$ *is also on MBB, and for every* $j \in M$*,* $q_j \in \{1, k\}$.

Proof. Note that initially all prices are either 1 or k. Since all price rises are by a factor of k (Line 9), final prices are of the form $p_j = k^{s_j}$, for $s_j \in \mathbb{Z}_{\geq 0}$. Let j_0 be the smallest priced good with $p_{j_0} = k^s$, and let $j_0 \in \mathbf{x}_i$, for some agent $i \in N$. Then $\forall j \in \mathbf{x}_i : p_j \in \{k^s, k^{s+1}\}$. By the MBB condition for any agent $h \neq i$ for $j' \in \mathbf{x}_h$ and $j \in \mathbf{x}_i$:

$$\frac{v_{hj'}}{p_{j'}} \geq \frac{v_{hj}}{p_j},$$

which gives:

$$p_{j'} \leq \frac{v_{hj'}}{v_{hj}} p_j \leq k^{s+2} \quad .$$

Thus all $p_j \in \{k^s, k^{s+1}, k^{s+2}\}$. Either all $p_j \in \{k^s, k^{s+1}\}$, or $\exists j \in \mathbf{x}_h$ with $p_j = k^{s+2}$, for some agent $h \in N$. Then by the MBB condition for any good j':

$$\frac{v_{hj}}{p_j} \geq \frac{v_{hj'}}{p_{j'}},$$

which gives:

$$p_{j'} \geq \frac{v_{hj'}}{v_{hj}} p_j \geq k^{s+1} \quad .$$

Thus either all $p_j \in \{k^s, k^{s+1}\}$ or all $p_j \in \{k^{s+1}, k^{s+2}\}$. In either case we can scale the prices to belong to $\{1, k\}$. $\square$

This in fact shows that at any stage of Algorithm 1, the prices of goods are in $\{k^s, k^{s+1}\}$ for some $s \in \mathbb{Z}_{\geq 0}$. This, along with the fact that goods are always transferred along MBB edges, and the prices are raised only by factor of k, leads us to conclude that the MBB condition is never violated for any agent and the allocation is always on MBB throughout the run of the algorithm. Hence the allocation is fPO.

Lemma 3. *The allocation* $\mathbf{x}$ *returned by Algorithm 1 is on MBB w.r.t. prices* $\mathbf{p}$ *upon termination. Thus,* $\mathbf{x}$ *is fPO.*

The full proof of the above Lemma appears in Appendix B. We now show correctness:

Lemma 4. *The allocation* $\mathbf{x}$ *returned by Algorithm 1, together with the prices* $\mathbf{p}$ *on termination is pEFX.*

Proof. To see why $(\mathbf{x}, \mathbf{p})$ is pEFX, first note that by Lemma 2, we can assume the prices are in $\{1, k\}$. Suppose $(\mathbf{x}, \mathbf{p})$ is not pEFX. Then there must be an agent h and some good $j \in \mathbf{x}_h$ s.t. $\mathbf{p}(\mathbf{x}_h \setminus \{j\}) > \mathbf{p}(\mathbf{x}_i)$, where i is the least spender. If $h \notin C_i$, the algorithm would not have halted (negation of condition in line 8 holds). Therefore h is in C_i. Since the algorithm has halted, this means that along all alternating paths $(i, j_1, i_1, \ldots, h', j, h)$, it is the case that

$\mathbf{p}(\mathbf{x}_h \setminus \{j\}) \leq \mathbf{p}(\mathbf{x}_i)$. Suppose there is some alternating path s.t. $p_j = 1$. We know for all $j' \in \mathbf{x}_h$, $p_{j'} \geq 1$. Thus:

$$\mathbf{p}(\mathbf{x}_i) \geq \mathbf{p}(\mathbf{x}_h \setminus \{j\}) = \mathbf{p}(\mathbf{x}_h) - 1 \geq \mathbf{p}(\mathbf{x}_h \setminus \{j'\}),$$

which means that i is pEFX towards h. Now suppose along all alternating paths $(i, j_1, i_1, \ldots, h', j, h)$, it holds that $p_j = k$. Since $(\mathbf{x}, \mathbf{p})$ is not pEFX, it must be the case that there is some good $j' \in \mathbf{x}_h$ that is not reachable from i via any alternating path, with $p_{j'} = 1$. This means that $j' \notin mbb_{h'}$. Since $j \in mbb_{h'}$, comparing the bang-per-buck ratios gives $v_{h'j}/p_j > v_{h'j'}/p_{j'}$. This implies $v_{h'j} > kv_{h'j'}$ which is not possible when $v_{h'j}, v_{h'j'} \in \{1, k\}$, thus leading to a contradiction. Hence we conclude that $(\mathbf{x}, \mathbf{p})$ is pEFX. $\square$

Lemma 5. *Algorithm 1 terminates in polynomial-time.*

Proof. (Sketch) We first note that the number of alternating paths from an agent i to an agent h who owns a good j which is then transferred to an agent h' is at most $n \cdot n \cdot m$. Thus there are at most $\mathsf{poly}(n, m)$ transfers with the same LS and no price rise step.

Next, we argue that the number of identity changes of the least spender without a price rise step is $\mathsf{poly}(n, m)$. Suppose an agent i ceases to be the LS at iteration t, and subsequently (without price-rise steps) becomes the LS again for the first time at time t'. We show that the spending of i is strictly larger at t' than at t, and hence has strictly larger utility. Since all utility values are integers, the increase in i's utility is by at least 1. In any allocation $\mathbf{x}$, if s_i (resp. t_i) is the number of goods in $\mathbf{x}_i$ that are valued at b (resp. a) by i, the utility of i is $u_i = s_i b + t_i a$. Since $0 \leq s_i, t_i \leq m$, the number of different utility values i can get in any allocation is at most $O(m^2)$. Thus, for any agent i, the number of times her utility increases is at most $O(m^2)$. This is our key insight. It implies that without price rises, any agent can become the least spender only $O(m^2)$ times. Hence, the number of identity changes of the LS in the absence of price rise steps is at most $O(nm^2)$.

For polynomial run-time, it remains to be shown that the number of price-rises is $\mathsf{poly}(n, m)$. We do this via a potential function argument similar to [22]. The full proof is present in 1. $\square$

Finally, we show that the allocation returned by our algorithm also provides a good approximation to the MNW, and defer the proof to the Appendix B.

Lemma 6. *Let $\mathbf{x}$ be the allocation output by Algorithm 1. If $\mathbf{x}^*$ is an MNW allocation, then* $\mathsf{NW}(\mathbf{x}) \geq \frac{1}{1.061}\mathsf{NW}(\mathbf{x}^*)$.

Proof. (Sketch) Let $\mathbf{p}$ be the price vector on termination. Consider a scaled fair division instance $I' = (N, M, V')$ with identical valuations, where $v'_{ij} = p_j$ for each $i \in N, j \in M$. Since $(\mathbf{x}, \mathbf{p})$ is pEFX for the instance I (Lemma 4), $\mathbf{x}$ is EFX for the instance I'. Barman et al., [20] showed that for identical valuations, any EFX allocation provides a 1.061-approximation to the maximum Nash welfare.

Hence $\mathbf{x}$ provides a 1.061-approximation to the MNW of I', and we can show that because $(\mathbf{x}, \mathbf{p})$ is on MBB (from Lemma 3), $\mathbf{x}$ gives the same guarantee for the MNW of the instance I. □

Lemmas 1, 3, 4, 5, and 6 together prove Theorem 3.

3.2. EFX+PO for 3-valued instances

On generalizing the class of valuations slightly to $\{0, a, b\}$, EFX+PO allocations are no longer guaranteed to exist [11] (see Appendix A for an example).

Therefore we investigate the complexity of checking if an EFX and PO allocation exists or not, and show that this problem is NP-hard.

Theorem 4. *Given a fair division instance $I = (N, M, V)$, checking if I admits an allocation that is both EFX and PO is NP-hard, even when I is a $\{0, a, b\}$-instance.*

We reduce from 2P2N3SAT, an instance of which consists of a 3SAT formula over n variables $\{x_i\}_{i\in[n]}$ in conjunctive normal form, and m distinct clauses $\{C_j\}_{j\in[m]}$, with three literals per clause. Additionally, each variable x_i appears exactly twice negated and exactly twice unnegated in the formula. Deciding if there exists a satisfying assignment for such a formula is known to be NP-complete [41]. Given a 2P2N3SAT-instance $(\{x_i\}_{i\in[n]}, \{C_j\}_{j\in[m]})$, we construct a fair division instance with $2n + m$ agents and $7n + m$ goods, with all values in $\{0, 1, 3\}$ as follows:

1. For every variable x_i, create two agents T_i and F_i. Also create 7 goods: $d_i^T, d_i^F, g_i, y_i^T, z_i^T, y_i^F, z_i^F$. Both T_i and F_i value g_i at 3. T_i values d_i^T, y_i^T, z_i^T at 1, and F_i values d_i^F, y_i^F, z_i^F at 1. T_i and F_i value all other goods at 0.
2. For every clause $C_j = \ell_1 \vee \ell_2 \vee \ell_3$, create one agent D_j and a good e_j. D_j values e_j at 1. If for any $k \in [3]$, $\ell_k = x_i$ for some $i \in [n]$ then D_j values y_i^T, z_i^T at 1; and if for any $k \in [3]$, $\ell_k = \neg x_i$ for some $i \in [n]$ then D_j values y_i^F, z_i^F at 1. D_j values all other goods at 0.

We show that this instance admits an EFX+PO allocation iff the formula has a satisfying assignment. We illustrate the correspondence between PO allocations and assignments, and how our construction enforces EFX allocations to give rise to satisfying assignments (and vice versa). In any PO allocation, for every $i \in [n]$, d_i^A must be assigned to A_i, for $A \in \{T, F\}$; and g_i must be assigned to T_i or F_i. Consider the assignment $x_i = A$, if g_i is allotted to A_i, for $A \in \{T, F\}$, for all $i \in [n]$. Suppose for some $i \in [n]$, g_i is allocated to T_i. Then in an EFX allocation, because F_i must not envy T_i after removing d_i^T from the bundle of T_i, F_i must get utility at least 3. This is only possible if both y_i^F and z_i^F are allocated to F_i. This leaves y_i^T, z_i^T for the clause agents, when x_i is True. Thus if there is a satisfying assignment, the remaining goods can be allocated to the clause agents in an EFX+PO manner. Also, if all assignments are unsatisfying, some clause agent will end up not being EFX towards another agent in any PO allocation.

We defer the full proof to Appendix B, and also show:

Algorithm 2 EQX+fPO allocation for $\{a, b\}$-instances

```
Input: Fair division {a,b}-instance $(N, M, V)$
Output: An integral allocation $\mathbf{x}$
1: $(\mathbf{x}, \mathbf{p}) \leftarrow$ Integral welfare-maximizing market allocation, where $p_j = v_{ij}$ for
   $j \in \mathbf{x}_i$.
2: Let $i \in \mathsf{argmin}_{h \in N} v_h(\mathbf{x}_h)$ be the least utility (LU) agent
3: while there is an alternating path $(i, j_1, i_1, \dots, j_\ell, i_\ell)$, s.t. $v_{i_\ell}(\mathbf{x}_{i_\ell} \setminus \{j_\ell\}) >$
   $v_i(\mathbf{x}_i)$ do
4:     Transfer $j_\ell$ from $i_\ell$ to $i_{\ell-1}$, update allocation $\mathbf{x}$
5:     Update the LU agent $i \in \mathsf{argmin}_{h \in N} v_h(\mathbf{x}_h)$
6: if $\forall$ agents $h \notin C_i$, and $\forall j \in \mathbf{x}_h : v_h(\mathbf{x}_h \setminus \{j\}) \leq v_i(\mathbf{x}_i)$ then return $\mathbf{x}$
   ▷ EQX condition satisfied for all agents not reachable through alt. paths
   from LU agent; $C_i$ is defined in Def. 1
7: else
8:     Raise prices of goods in $C_i$ by a multiplicative factor of $a/b$
9:     Repeat from Line 2
```

Lemma 7. *Given a fair division instance $I = (N, M, V)$, checking if I admits an allocation that is both EFX and fPO is NP-complete, even when I is a $\{0, a, b\}$-instance.*

4. Computing EQX+PO allocations

We now consider relaxations of the fairness notion of equitability, which demands that all agents receive roughly the same utility. An allocation is said to be equitable up to any good (EQX) if for all $i, h \in N$, for all $j \in \mathbf{x}_h$ we have $v_i(\mathbf{x}_i) \geq v_h(\mathbf{x}_h \setminus \{j\})$. It is known that for binary instances, EQX+PO allocations can be computed in polynomial-time, whenever they exist [22]. Hence we first consider $\{a, b\}$-instances. We show that:

Theorem 5. *Given a fair division $\{a, b\}$-instance $I = (N, M, V)$, an allocation that is both EQX and fPO exists and can be computed in polynomial-time.*

We prove this by showing that Algorithm 2 terminates in polynomial-time with an EQX+fPO allocation. Since we are interested in EQX as opposed to EFX, we need not construct a pEFX allocation and can instead work directly with the values. Since the techniques used in the analysis of Algorithm 2 are similar to the analysis of Algorithm 1, we defer the full proof to Appendix C. We remark here that our techniques also enable us to show that EQ1+fPO allocations can be computed in polynomial-time for $\{a, b\}$-instances of *chores*.

For $\{0, a, b\}$-instances, EQX+PO allocations need not exist (example in Appendix A). Therefore, we study the complexity of checking if an EQX+PO allocation exists or not, and show that this problem too is NP-hard. The full proof is deferred to Appendix C.

Theorem 6. *Given a fair division instance $I = (N, M, V)$, checking if I admits an allocation that is both EQX and PO is NP-hard, even when I is a $\{0, a, b\}$-instance.*

5. Maximizing Nash Welfare

We turn to the problem of maximizing Nash welfare for t-valued instances when $t \in \{2, 3\}$. We state our results and sketch the proofs, and defer full proofs to Appendix D.

5.1. MNW for 2-valued instances

Recall that for $\{a, b\}$-instances, we showed in Lemma 6 that Algorithm 1 approximates the MNW to a 1.061-factor. Our main result of this section is a polynomial-time algorithm for computing the MNW allocation for a subclass of 2-valued instances.

Theorem 7. *Given a fair division $\{a, b\}$-instance $I = (N, M, V)$, where $a/b > m - n$, an MNW allocation can be computed in polynomial-time.*

We present a high-level idea of our Algorithm 3 and technique. We scale all values to $\{1, k\}$. We assume that for every agent $i \in N$, there is some good $j \in M$, such that $v_{ij} = k$, since if for some agent i, $v_{ij} = 1$, $\forall j \in M$, then we could rescale her values and set $v_{ij} = k$, $\forall j \in M$. Further we assume w.l.o.g. that $m > n$, since otherwise an MNW allocation $\mathbf{x}$ can be found by computing a weighted maximum matching. Under these assumptions, every agent receives at least one good in $\mathbf{x}$, and hence $\mathsf{NW}(\mathbf{x}) > 0$. We also use the fact that there is a polynomial-time algorithm $\mathsf{BinaryMNW}$ that computes an MNW allocation for binary instances.

The main idea is that when $k > m - n$, any agent prefers a single good that they value at k over any number of goods they value at 1, that they can get. We first try to give every agent one good that they value at k. We do this by computing a maximum matching $\mathcal{M}$ in G and checking if its size is n. If yes, then we argue that an MNW allocation can be computed as follows. Let $M_{high} = \{j \in M : \exists i \in N \text{ s.t. } v_{ij} = k\}$ be the set of goods valued at k by at least one agent, and let $M_{low} = M \setminus M_{high}$ be the set of goods valued at 1 by all agents. We consider the binary instance (N, M_{high}, V') where for any agent i and good $j \in M_{high}$, $v'_{ij} = \lfloor v_{ij}/k \rfloor \in \{0, 1\}$, and compute an MNW allocation $\mathbf{x}'$ using the algorithm $\mathsf{BinaryMNW}$. Next, we construct $\mathbf{x}$, an MNW allocation of I, from the allocation $\mathbf{x}'$ by allocating the remaining goods M_{low} in a round-robin fashion to the set N_1 of agents who get the least utility in $\mathbf{x}'$. This corresponds to the procedure $\mathsf{RoundRobin}$ on Line 10, which we now describe. Let $|M_{low}| = |N_1|q + r$, where $q, r \in \mathbb{Z}_{\geq 0}$, and $0 \leq r < |N_1|$. Then exactly r agents in N_1 are given $q + 1$ goods from M_{low}, and $|N_1| - r$ agents are given q goods from M_{low}. This corresponds to lines 5-11 of Algorithm 3.

Suppose on the other hand $\mathcal{M}$ has size less than n. Let P be the set of agents that are unmatched under $\mathcal{M}$. We iteratively compute the sets $P^{t+1} = \{h \in N :$

Algorithm 3 MNW for $\{a, b\}$-instances with $a/b > m - n$

```
Input: Fair division $\{a, b\}$-instance $(N, M, V)$
Output: An integral allocation $\mathbf{x}$
1: $k \leftarrow a/b$
2: Scale valuations so that $v_{ij} \leftarrow v_{ij}/k$, and s.t. for every $i \in N$, there is some
   $j \in M$ s.t. $v_{ij} = k$.
3: Let $G = (N, M, E)$ be a bipartite graph with $(i, j) \in E$ iff $v_{ij} = k$ for $i \in N$
   and $j \in M$.
4: Compute a maximum matching $\mathcal{M}$ in $G$.
5: if $|\mathcal{M}| = n$ then
6:     $M_{high} \leftarrow \{j \in M : \exists i \in N$ s.t. $v_{ij} = k\}$
7:     $v'_{ij} \leftarrow \lfloor v_{ij}/k \rfloor$ for every $i \in N$, $j \in M_{high}$
8:     $\mathbf{x}' \leftarrow \mathsf{BinaryMNW}(N, M_{high}, V')$
9:     $N_1 \leftarrow \{i \in N : v_i(\mathbf{x}'_i) = \min_{h \in N} v_h(\mathbf{x}'_h)\}$
10:    $\mathbf{x} \leftarrow \mathsf{RoundRobin}(\mathbf{x}', N_1, M \setminus M_{high})$        ▷ see text
11:    return $\mathbf{x}$
12: else    // $|\mathcal{M}| < n$
13:    $P \leftarrow$ Set of agents unmatched under $\mathcal{M}$
14:    Compute the set $Q$ as described in text
15:    $T \leftarrow$ Set of goods matched to $Q$ under $\mathcal{M}$
16:    $v'_{ij} = \lfloor v_{ij}/k \rfloor$ for $i \in N \setminus (P \cup Q)$, $j \in M \setminus T$
17:    $\mathbf{x}' \leftarrow \mathsf{BinaryMNW}(N \setminus Q, M \setminus T, V')$
18:    $\mathbf{x}_i \leftarrow \mathbf{x}'_i$ for $i \in N \setminus Q$
19:    $\mathbf{x}_i = \{j\}$ for $i \in Q$ s.t. $(i, j) \in \mathcal{M}$
20:    return $\mathbf{x}$
```

$\exists i \in P^t, \exists j \in \mathbf{x}_h$ s.t. $v_{ij} = k\}$ for $t \geq 0$, where $P^0 = P$, and let $Q = \bigcup_{t \geq 1} P^t$. Then $S = P \cup Q$ is a Hall's violator set of agents. Let T be the set of neighbors of S (and Q) in G. Algorithm 3 then computes an MNW allocation $\mathbf{x}'$ for the binary instance $(N \setminus Q, M \setminus T, V')$, where $v'_{ij} = \lfloor v_{ij}/k \rfloor$ for $i \in N \setminus S$, and 1 if $i \in P$, in polynomial-time using BinaryMNW, and then computes an MNW allocation $\mathbf{x}$ of I from $\mathbf{x}'$ by further allocating to agents in Q the good they are matched to in $\mathcal{M}$. This corresponds to lines 12-20 of Algorithm 3. The formal proof of Theorem 7 lies on several non-trivial arguments and is deferred to Appendix D.

It is known that an MNW allocation is PO [10]. Our next result improves the efficiency guarantee offered by an MNW allocation of a 2-valued instance to fractional Pareto-optimality.

Theorem 8. *Given a fair division $\{a, b\}$-instance $I = (N, M, V)$, any MNW allocation is fPO. This result is sharp; an MNW allocation of a 3-valued instance need not be fPO.*

To show this, we present a price-setting algorithm in Appendix D that assigns a price p_j to each good j s.t. $(\mathbf{x}, \mathbf{p})$ is on MBB, where $\mathbf{x}$ is an MNW allocation. Together with Corollary 2, this shows $\mathbf{x}$ is fPO. We include an example in Appendix D showing sharpness.

5.2. MNW for 3-valued instances

Our final result shows APX-hardness for the MNW problem with we slighly generalize the class of allowed values to $\{0, a, b\}$. This rules out the existence of a polynomial-time approximation scheme (PTAS) for the MNW problem even $\{0, a, b\}$-instances, thus strengthening the result of [1], who showed NP-hardness for the same.

Theorem 9. *Given a fair division instance $I = (N, M, V)$, it is NP-hard to approximate the MNW to a factor better than 1.00019, even for $\{0, a, b\}$-instances.*

We present the reduction and defer the full proof to Appendix D. We consider a 2P2N3SAT-instance: $\{x_i\}_{i\in[n]}, \{C_j\}_{j\in[m]}$, where $3m = 4n$. For each variable x_i, we create two agents T_i, F_i and one good g_i which is valued at 2 by both T_i, F_i. For each clause C_j, we create a good h_j which is valued at 1 by agent A_i if setting $x_i = A$ makes C_j true, for $A \in \{T, F\}$. We also create $2n - m$ dummy goods $\{d_j\}_{j\in[2n-m]}$ which are valued at 1 by all agents. All other values are 0. We show that if we can approximate the MNW to a factor better than 1.00019, we can decide if there is an assignment with $\geq \rho_1 m$ clauses, or all assignments satisfy at most $\leq \rho_2 m$ clauses, for specific constants ρ_1, ρ_2. The latter problem is known to be NP-complete [41].

6. Conclusion

In this paper, we push the boundary between tractable and intractable cases for the problems of fair and efficient allocations. We presented positive algorithmic results for computing EFX+PO, EQX+PO, and 1.061-approximate MNW allocations for 2-valued instances. In contrast, we showed that for 3-valued instances, checking existence of EFX+PO (or EQX+PO) allocations is NP-complete, and computing MNW is APX-hard. Our techniques can be adapted to compute EQ1+PO allocations for 2-valued instances of *chores*. An interesting direction for future work is to develop approximation algorithms for the MNW problem for k-valued instances where k is a small constant at least 3. Another challenging problem is investigating if EFX (or EFX+PO) allocations exist (and can be efficiently computed) for 2-valued instances of chores.

References

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

[2] J. Garg, A. Murhekar, Computing fair and efficient allocations with few utility values, in: In Proc. 14th Symp. Algorithmic Game Theory (SAGT), 2021.

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

[4] S. Brams, A. Taylor, Fair Division: From Cake-Cutting to Dispute Resolution, Cambridge University Press, 1996.

[5] H. Moulin, Fair Division and Collective Welfare, Mit Press, MIT Press, 2004.

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

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

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

[9] R. J. Lipton, E. Markakis, E. Mossel, A. Saberi, On approximately fair allocations of indivisible goods, in: Proceedings of the 5th ACM Conference on Electronic Commerce (EC), 2004, pp. 125–131.

[10] I. Caragiannis, D. Kurokawa, H. Moulin, A. D. Procaccia, N. Shah, J. Wang, The unreasonable fairness of maximum Nash welfare, ACM Trans. Econ. Comput. 7 (3).

[11] B. Plaut, T. Roughgarden, Almost envy-freeness with general valuations, SIAM Journal on Discrete Mathematics 34 (2) (2020) 1039–1068.

[12] B. R. Chaudhury, J. Garg, K. Mehlhorn, EFX exists for three agents, in: Proceedings of the 21st ACM Conference on Economics and Computation (EC), 2020, p. 1–19.

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

[14] B. R. Chaudhury, J. Garg, K. Mehlhorn, R. Mehta, P. Misra, Improving EFX guarantees through rainbow cycle number, in: Proceedings of the 22nd ACM Conference on Economics and Computation, EC '21, Association for Computing Machinery, New York, NY, USA, 2021, p. 310–311.

[15] S. Barman, S. K. Krishnamurthy, R. Vaish, Finding fair and efficient allocations, in: Proceedings of the 19th ACM Conference on Economics and Computation (EC), 2018, pp. 557–574.

[16] B. de Keijzer, S. Bouveret, T. Klos, Y. Zhang, On the complexity of efficiency and envy-freeness in fair division of indivisible goods with additive preferences, in: F. Rossi, A. Tsoukias (Eds.), Algorithmic Decision Theory, 2009, pp. 98–110.

[17] E. Lee, APX-hardness of maximizing Nash social welfare with indivisible items, Information Processing Letters 122.

[18] A. Murhekar, J. Garg, On fair and efficient allocations of indivisible goods, Proceedings of the AAAI Conference on Artificial Intelligence 35 (6) (2021) 5595–5602.

[19] A. Darmann, J. Schauer, Maximizing Nash product social welfare in allocating indivisible goods, SSRN Electronic Journal 247.

[20] S. Barman, S. K. Krishnamurthy, R. Vaish, Greedy algorithms for maximizing Nash social welfare, in: Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS), 2018, p. 7–13.

[21] L. Gourvès, J. Monnot, L. Tlilane, Near fairness in matroids, in: Proceedings of the 21st European Conference on Artificial Intelligence (ECAI), 2014, p. 393–398.

[22] R. Freeman, S. Sikdar, R. Vaish, L. Xia, Equitable allocations of indivisible goods, in: Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI), 2019, pp. 280–286.

[23] H. Akrami, B. R. Chaudhury, M. Hoefer, K. Mehlhorn, M. Schmalhofer, G. Shahkarami, G. Varricchio, Q. Vermande, E. v. Wijland, Maximizing Nash social welfare in 2-value instances, Proceedings of the AAAI Conference on Artificial Intelligence 36 (5) (2022) 4760–4767.

[24] J. Garg, M. Hoefer, K. Mehlhorn, Satiation in Fisher markets and approximation of Nash social welfare, CoRR abs/1707.04428.

[25] D. Golovin, Max-min fair allocation of indivisible goods, Technical Report, CMU-CS-05-144.

[26] H. Aziz, P. Biró, J. Lang, J. Lesca, J. Monnot, Efficient reallocation under additive and responsive preferences, Theoretical Computer Science 790 (2019) 1 – 15.

[27] H. Aziz, The Hylland-Zeckhauser rule under bi-valued utilities, CoRR abs/2006.15747.

[28] V. V. Vazirani, M. Yannakakis, Computational complexity of the Hylland-Zeckhauser scheme for one-sided matching markets, in: Proceedings of the 12th Innovations in Theoretical Computer Science Conference, (ITCS), 2021.

[29] A. Bogomolnaia, H. Moulin, Random matching under dichotomous preferences, Econometrica 72 (1) (2004) 257–279.

[30] B. Bliem, R. Bredereck, R. Niedermeier, Complexity of efficient and envy-free resource allocation: Few agents, resources, or utility levels, in: Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI), 2016, p. 102–108.

[31] J. Garg, A. Murhekar, J. Qin, Fair and efficient allocations of chores under bivalued preferences, Proceedings of the AAAI Conference on Artificial Intelligence 36 (5) (2022) 5043–5050.

[32] N. Garg, T. Kavitha, A. Kumar, K. Mehlhorn, J. Mestre, Assigning papers to referees, Algorithmica, v.x, x-x (2009) 58.

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

[34] D. Chakrabarty, S. Khanna, S. Li, On (1, e)-restricted assignment makespan minimization, in: Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), 2015, p. 1087–1101.

[35] R. Cole, V. Gkatzelis, Approximating the Nash social welfare with indivisible items, in: Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing (STOC), 2015, p. 371–380.

[36] B. R. Chaudhury, Y. K. Cheung, J. Garg, N. Garg, M. Hoefer, K. Mehlhorn, On fair division for indivisible items, in: 38th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS), 2018, pp. 25:1–25:17.

[37] J. Garg, M. Hoefer, K. Mehlhorn, Approximating the Nash social welfare with budget-additive valuations, in: Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), 2018, pp. 2326–2340.

[38] S. Barman, S. Krishnamurthy, On the proximity of markets with integral equilibria, Proceedings of the 33rd AAAI Conference on Artificial Intelligence (2019) 1748–1755.

[39] P. McGlaughlin, J. Garg, Improving Nash social welfare approximations, J. Artif. Intell. Res. 68 (2020) 225–245.

[40] A. Mas-Colell, P. Mas-Colell, W. D, M. Whinston, J. Green, C. Hara, P. Green, I. Segal, O. U. Press, S. Tadelis, Microeconomic Theory, Oxford University Press, 1995.

[41] P. Berman, M. Karpinski, A. Scott, Approximation hardness of short symmetric instances of MAX-3SAT, Electronic Colloquium on Computational Complexity (ECCC).

Appendix A. Examples

Appendix A.1. PO allocations are not necessarily fPO

Consider the following instance with two agents 1 and 2; and three goods g_1, g_2 and g_3.

Table A.1: Instance for which a PO allocation is not fPO

	g_1	g_2	g_3
A_1	4	2	1
A_2	4	1	2

Consider the allocation $\mathbf{x}$ given by $\mathbf{x}_1 = \{g_1\}, \mathbf{x}_2 = \{g_2, g_3\}$. Agent 1 gets a utility of $u_1 = 4$ and agent 2 gets $u_2 = 3$. Clearly $\mathbf{x}$ is PO, because it maximizes the utilitarian social welfare. Now consider the fractional allocation $\mathbf{y}$ given by $\mathbf{y}_1 = \{0.75g_1, 0.5g_2\}, \mathbf{y}_2 = \{0.25g_1, 0.5g_2, g_3\}$. Under $\mathbf{y}$, agent 1 gets a utility of 4 but agent 2 agents a utility of 3.5, which is better than the utility under $\mathbf{x}$. Hence $\mathbf{x}$ is PO but not fPO.

Appendix A.2. Match&Freeze need not return a PO allocation

We show that the EFX allocation returned by the algorithm of [1] need not be PO. Consider the following $\{a, b\}$-instance with three agents (A_1, A_2, A_3) and 8 goods $(g_1, \ldots, g_8)$. Let $a > b > 0$.

Table A.2: Instance for which Match&Freeze does not return a PO allocation

	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8
A_1	a	a	a	a	b	b	b	b
A_2	a	a	a	a	b	b	b	b
A_3	a	a	a	b	a	a	a	a

In the first iteration of Match&Freeze, suppose g_i is matched to A_i for $i \in [3]$. In the second round, suppose the matching assigns g_4 to A_1 and g_5 to A_3, after which g_6 is assigned to A_2. Since g_6 is valued at b by A_2, and since A_2 values g_4 at a, agent A_1 gets frozen. In the next and final round, A_3 gets g_8 and A_2 gets g_7. In the resulting allocation $\mathbf{x}$, the vector of utilities the agents receive is $(2a, a + 2b, 3a)$. However, consider the allocation $\mathbf{y}$ given by $\mathbf{y}_1 = \{g_1, g_3\}$, $\mathbf{y}_2 = \{g_2, g_4\}$ and $\mathbf{y}_3 = \{g_5, g_7, g_8\}$. The vector of utilities under $\mathbf{y}$ is $(2a, 2a, 3a)$. Thus when $a > 2b$, $\mathbf{y}$ dominates $\mathbf{x}$, showing that $\mathbf{x}$ is not PO.

Appendix A.3. EFX+PO allocations need not exist

Consider the instance in Table A.3 from [11] with 2 agents A_1, A_2 and 3 goods $\{g_1, g_2, g_3\}$.

In any PO allocation, g_2 has to be allocated to A_1, and g_3 has to be allocated to A_2. Thus, the only PO allocations are $(\{g_1, g_2\}, \{g_3\})$, and $(\{g_2\}, \{g_1, g_3\})$. However neither of them are EFX, since in the first allocation A_2 envies A_1 after removing g_2, and in the second allocation A_1 envies A_2 after removing g_3. In fact, this example also shows that EQX+PO allocations need not exist.

Table A.3: Instance for which EFX+PO allocations do not exist

	g_1	g_2	g_3
A_1	2	1	0
A_2	2	0	1

Appendix B. Missing Proofs from Section 3

Lemma 2. *For every outcome* $(\mathbf{x}, \mathbf{p})$ *constructed during the run of Algorithm 1, there exists a set of prices* $\mathbf{q}$ *such that* $(\mathbf{x}, \mathbf{q})$ *is also on MBB, and for every* $j \in M$*,* $q_j \in \{1, k\}$*.*

Proof. Note that initially all prices are either 1 or k. Since all price rises are by a factor of k (Line 9), final prices are of the form $p_j = k^{s_j}$, for $s_j \in \mathbb{Z}_{\geq 0}$. Let j_0 be the smallest priced good with $p_{j_0} = k^s$, and let $j_0 \in \mathbf{x}_i$, for some agent $i \in N$. Then $\forall j \in \mathbf{x}_i : p_j \in \{k^s, k^{s+1}\}$. By the MBB condition for any agent $h \neq i$ for $j' \in \mathbf{x}_h$ and $j \in \mathbf{x}_i$:

$$\frac{v_{hj'}}{p_{j'}} \geq \frac{v_{hj}}{p_j},$$

which gives:

$$p_{j'} \leq \frac{v_{hj'}}{v_{hj}} p_j \leq k^{s+2} \ .$$

Thus all $p_j \in \{k^s, k^{s+1}, k^{s+2}\}$. Either all $p_j \in \{k^s, k^{s+1}\}$, or $\exists j \in \mathbf{x}_h$ with $p_j = k^{s+2}$, for some agent $h \in N$. Then by the MBB condition for any good j':

$$\frac{v_{hj}}{p_j} \geq \frac{v_{hj'}}{p_{j'}},$$

which gives:

$$p_{j'} \geq \frac{v_{hj'}}{v_{hj}} p_j \geq k^{s+1} \ .$$

Thus either all $p_j \in \{k^s, k^{s+1}\}$ or all $p_j \in \{k^{s+1}, k^{s+2}\}$. In either case we can scale the prices to belong to $\{1, k\}$. $\square$

Lemma 3. *The allocation* $\mathbf{x}$ *returned by Algorithm 1 is on MBB w.r.t. prices* $\mathbf{p}$ *upon termination. Thus,* $\mathbf{x}$ *is fPO.*

Proof. We first show that $(\mathbf{x}, \mathbf{p})$ forms a market equilibrium for the Fisher market instance (N, M, V, e), where for every $i \in N$, $e_i = \mathbf{p}(\mathbf{x}_i)$. It is easy to see that the market clears and the budget of every agent is exhausted. It remains to be shown that every agent buys goods from their MBB-set. To see this, notice that this is the case in the initial allocation in Line 2 of the algorithm. Suppose we assume inductively that at iteration t, in the corresponding allocation $\mathbf{x}$, every agent buys MBB goods. We ensure that the goods are transferred along MBB edges, and thus no transfer step causes the MBB condition of any agent to be violated. Consider a price rise step, in which the prices of all goods in C_i, the component of the LS i, are increased by a factor of k. Notice that since prices of these goods are raised, no agent $h \notin C_i$ prefers these goods over their own goods, and hence they continue to own goods from their MBB set. For any agent $h \in C_i$ and a good $j \notin C_i$, the bang-per-buck that j gets from h before the price rise is strictly less than her MBB ratio, since j is not in the MBB set of h. Further, since all prices are powers of k, the bang-per-buck values are also (non-positive) powers of k. Thus after the price rise, it cannot happen that a good $j \notin C_i$ gives a higher bang-per-buck than the goods h already owns, since her MBB ratio drops only by a factor of k. This shows that the MBB condition is never violated for any agent throughout the algorithm. Thus, the final allocation with the final price vector $(\mathbf{x}, \mathbf{p})$ forms a market equilibrium. The First Welfare Theorem implies that $\mathbf{x}$ is fPO. □

Lemma 5. *Algorithm 1 terminates in polynomial-time.*

Proof. We first argue that the number of iterations with the same least spender i in the absence of any price rise steps is $\mathsf{poly}(n, m)$. To see this, we count the number of alternating paths from i to an agent h which owns a good j which is then transferred to an agent h'. The number of such paths is at most n^2m, thus there are at most $\mathsf{poly}(n, m)$ transfers with the same LS and no price rise step.

Next we argue that the number of identity changes of the least spender without a price rise step is $\mathsf{poly}(n, m)$. First note that the spending of the least spender never decreases in the execution of the algorithm. Next, observe that in the absence of price rise steps an agent stops being a least spender only if she gets a good. Suppose at an iteration t, when the allocation is $\mathbf{x}$, the LS i gets a good j and ceases to be the LS. Suppose she becomes the LS again for the first time after t at the iteration t'', when the allocation is $\mathbf{x}''$. If she did not lose any good between t and t'', then her spending has strictly increased. Suppose on the other hand, let t' be the iteration after t and before t'' where i loses a good j' for the last time. Let $\mathbf{x}'$ be the corresponding allocation and let h be the LS at t'. Let $\mathbf{p}$ be the price vector. Now observe that:

$$\mathbf{p}(\mathbf{x}''_i) \geq \mathbf{p}(\mathbf{x}'_i \setminus \{j'\}) > \mathbf{p}(\mathbf{x}'_h) \geq \mathbf{p}(\mathbf{x}_i). \tag{B.1}$$

The first inequality holds because j' is the last good that i loses. The second inequality holds because j' is removed from i only because i violated the pEF1 condition w.r.t. the LS h at $\mathbf{x}'$. The third inequality holds because the spending of the LS does not decrease.

Since we are assuming that the prices remain unchanged, the MBB-ratio of the agents do not change. Hence the agents' utilities are proportional to their spendings. This means that between the events when a LS i ceases to be a LS, and subsequently becomes the LS again, her utility has strictly increased, since her spending strictly increases from (B.1). Since all utility values are integers, the increase in i's utility is by at least 1. In any allocation $\mathbf{x}$, if s_i (resp. t_i) is the number of goods in $\mathbf{x}_i$ that are valued at b (resp. a) by i, the utility of i is $u_i = s_ib + t_ia$. Since $0 \leq s_i, t_i \leq m$, the number of different utility values i can get in any allocation is at most $O(m^2)$. Thus, for any agent i, the number of times her utility increases is at most $O(m^2)$. This means that without price rises, any agent can become the least spender only $O(m^2)$ times. Thus the number of identity changes of the LS in the absence of price rise steps is at most $O(nm^2)$.

Thus, for polynomial run time, all that remains to be shown is that the number of price rise steps is polynomial in n, m. For this, we observe that the set E_t of pEF1-violators just before price rise step number t does not increase with t. This is because if a new agent becomes an pEF1-violator after a transfer step, she will cease to be an pEF1-violator in subsequent iterations before the next price rise event, since she will lose one or more goods in a transfer step. Further, no new agent can become an pEF1-violator due to a price rise step, since the spending of the non-pEF1-violator agents only increases in a price rise step. The above is argued more formally in Lemma 16 of [15].

Thus, the goods belonging to the set of pEF1-violators have not experienced any price rise step, and their prices are the prices at the beginning, i.e., either 1 or k. Just before the price rise step number t, let $\alpha_1^t, \ldots, \alpha_n^t$ be the MBB-ratios of the agents. For any good $j \notin C_i$, where i is the LS agent, for every agent h, the MBB-ratio of h must at least be the bang-per buck h gets from j. Thus before any price rise step t, for every agent h and any good $j \notin C_i$:

$$\alpha_h^t \geq \frac{v_{hj}}{p_j} \geq \frac{1}{k}. \tag{B.2}$$

Consider the potential function given by $\phi(t) = \sum_{h \in N} \log_k \alpha_h^t$. Before the first price rise step, all the MBB-ratios are 1. Hence $\alpha_h^1 = 1$ for every $h \in N$. So, $\phi(1) = 0$. From (B.2), we have for all price rise steps, $\phi(t) \geq n \log_k(1/k) = -n$. After a price rise step, the price of every good in C_i is raised by k. Hence the MBB-ratio of every agent experiencing price-rise decreases by a factor k. Since at least one agent experiences price-rise, $\phi(t+1) \leq \phi(t) - 1$. Thus, the number of price rise steps is at most n. Hence the algorithm terminates in polynomial-time. □

Lemma 6. *Let* $\mathbf{x}$ *be the allocation output by Algorithm 1. If* $\mathbf{x}^*$ *is an MNW allocation, then* $\mathsf{NW}(\mathbf{x}) \geq \frac{1}{1.061}\mathsf{NW}(\mathbf{x}^*)$.

Proof. Recall from Lemmas 3 and 4 that $(\mathbf{x}, \mathbf{p})$ is a pEFX market outcome on MBB, where $\mathbf{p}$ is the price vector on termination. Let α_i be the MBB-ratio of agent i in $(\mathbf{x}, \mathbf{p})$. Consider a scaled fair division instance $I' = (N, M, V')$, where $v'_{ij} = \frac{1}{\alpha_i} v_{ij}$. Since the Nash welfare function is scale-invariant, if $\mathbf{x}^*$ is an MNW allocation for I, $\mathbf{x}^*$ is also an MNW allocation for I'; further if $\mathbf{x}$ is a β-approximation to the MNW value in I', then $\mathbf{x}$ is also a β-approximation to the MNW value in I. We also have by the MBB-condition that for every agent i, $v'_i(\mathbf{x}_i) = \mathbf{p}(\mathbf{x}_i)$, and that $v'_i(\mathbf{x}^*_i) \leq \mathbf{p}(\mathbf{x}^*_i)$. Thus:

$$\mathsf{NW}(\mathbf{x}) = \big(\prod_{i \in [n]} v'_i(\mathbf{x}_i)\big)^{1/n} = \big(\prod_{i \in [n]} \mathbf{p}(\mathbf{x}_i)\big)^{1/n}, \tag{B.3}$$

and

$$\mathsf{NW}(\mathbf{x}^*) = \Big(\prod_{i\in[n]} v'_i(\mathbf{x}^*_i)\Big)^{1/n} \leq \Big(\prod_{i\in[n]} \mathbf{p}(\mathbf{x}^*_i)\Big)^{1/n}. \tag{B.4}$$

Next we consider an instance $I'' = (N, M, V'')$ with identical valuations, that is: $v''_{ij} = p_j$ for all $i \in N, j \in M$. Since $(\mathbf{x}, \mathbf{p})$ is pEFX for the instance I, $\mathbf{x}$ is EFX for the instance I''. Let $\mathcal{X}$ be the set of all integral allocations of the goods to agents. [20] showed that for a fair division instance with identical valuations, any EFX allocation provides a 1.061-approximation to the maximum Nash welfare. Thus, we have:

$$\Big(\prod_{i\in[n]} v''_i(\mathbf{x}_i)\Big)^{1/n} \geq \frac{1}{1.061}\max_{\mathbf{y}\in\mathcal{X}}\Big(\prod_{i\in[n]} v''_i(\mathbf{y}_i)\Big)^{1/n},$$

which gives:

$$\begin{aligned}\Big(\prod_{i\in[n]} \mathbf{p}(\mathbf{x}_i)\Big)^{1/n} &\geq \frac{1}{1.061}\max_{\mathbf{y}\in\mathcal{X}}\Big(\prod_{i\in[n]} \mathbf{p}(\mathbf{y}_i)\Big)^{1/n} \\ &\geq \frac{1}{1.061}\Big(\prod_{i\in[n]} \mathbf{p}(\mathbf{x}^*_i)\Big)^{1/n}.\end{aligned} \tag{B.5}$$

Equations (B.3), (B.4) and (B.5) together give:

$$\mathsf{NW}(\mathbf{x}) \geq \frac{1}{1.061}\mathsf{NW}(\mathbf{x}^*),$$

as claimed. □

Theorem 4. *Given a fair division instance $I = (N, M, V)$, checking if I admits an allocation that is both EFX and PO is NP-hard, even when I is a $\{0, a, b\}$-instance.*

Proof. We reduce from 2P2N3SAT. An instance of 2P2N3SAT consists of a 3SAT formula over n variables $x_1, \dots, x_n$ in conjunctive normal form. There are m distinct clauses $C_1, C_2, \dots, C_m$, with three literals per clause. Additionally, each variable x_i appears exactly twice negated and exactly twice unnegated in the formula. Given an instance of 2P2N3SAT, deciding if there exists a satisfying assignment is known to be NP-complete [41].

Given a 2P2N3SAT-instance: $\{x_i\}_{i\in[n]}, \{C_j\}_{j\in[m]}$, we construct a fair division instance with $2n+m$ agents and $7n+m$ goods, with all values in $\{0,1,3\}$ as follows:

1. For every variable x_i, create two agents T_i and F_i. Also create 7 goods: $d_i^T, d_i^F, g_i, y_i^T, z_i^T, y_i^F, z_i^F$. Both T_i and F_i value g_i at 3. T_i values d_i^T, y_i^T, z_i^T at 1, and F_i values d_i^F, y_i^F, z_i^F at 1. T_i and F_i value all other goods at 0.
2. For every clause $C_j = \ell_1 \vee \ell_2 \vee \ell_3$, create one agent D_j and a good e_j. D_j values e_j at 1. If for any $k \in [3]$, $\ell_k = x_i$ for some $i \in [n]$ then D_j values y_i^T, z_i^T at 1; and if for any $k \in [3]$, $\ell_k = \neg x_i$ for some $i \in [n]$ then D_j values y_i^F, z_i^F at 1. D_j values all other goods at 0.

Suppose the formula has a satisfying assignment. Using this satisfying assignment, we create an allocation of goods as follows: For every x_i set to True, we assign g_i, d_i^T to T_i and d_i^F, y_i^F, z_i^F to F_i. For every x_i set to False, we assign g_i, d_i^F to F_i and d_i^T, y_i^T, z_i^T to T_i. We also assign e_j to D_j for every $j \in [m]$. Since the assignment satisfies all clauses, there exists one literal which is True in every clause C_j. If this literal is an unnegated variable, say x_i, then we assign one of y_i^T or z_i^T (whichever is left) to D_j. If this literal is a negated variable, say $\neg x_i$, then we assign one of y_i^F or z_i^F (whichever is left) to D_j. Notice that every item has been assigned to an agent that values it at the highest among all agents. Thus this allocation maximizes the utilitarian social Welfare, and is PO. Further, it is also EFX. This is because under this assignment for every $i \in [n]$, if x_i is True, T_i gets the highest utility of 4 and does not envy any agent, and in this case F_i gets a value of 3 and thus does not envy T_i even after removing d_i^T from T_i's bundle. An analogous argument holds when x_i is False. Next, for each $j \in [m]$, D_j gets a value of at least 2, and the number of goods belonging to an agent $D_{j'}$ valued at 1 by D_j, for $j \neq j'$ is at most 2 since the clauses are distinct. Thus the maximum utility D_j has for goods allocated to $D_{j'}$ is 2, and thus D_j does not envy $D_{j'}$ for any $j' \in [m]$. Finally note that D_j can have a value of at most 2 for the goods owned by T_i or F_i for any i, and hence D_j does not envy either. Thus the allocation is EFX.

Suppose on the other hand every assignment is unsatisfying, i.e., under any assignment there is always some clause C_j all of whose literals are False. Suppose there exists an EFX and PO allocation of goods to agents. Since the allocation is PO, for every $i \in [n]$, d_i^T must be assigned to T_i, d_i^F must be assigned to F_i, and g_i must be assigned to either T_i or F_i. Also, for each $j \in [m]$, e_j must be assigned to D_j. Suppose for some $i \in [n]$, g_i is allocated to T_i. Then, because F_i must not envy T_i after removing d_i^T from the bundle of T_i, F_i must have utility at least 3. This is only possible if both y_i^F and z_i^F are allocated to F_i. Similarly, if g_i is allocated to F_i, then for the allocation to be EFX, y_i^T and z_i^T both must be allocated to T_i. Now consider the assignment defined from the allocation in the following manner: if g_i is allocated to T_i, set x_i to True. If g_i is allocated to F_i, set x_i to False. By our assumption, this assignment leaves one at least one clause C_j unsatisfied, which means all literals in that clause evaluate to False. If x_i is a literal appearing in C_j, then D_j values y_i^T, z_i^T, but since x_i is set to False, both these goods are owned by T_i. Similarly, if $\neg x_i$ is a literal appearing in C_j, then D_j values y_i^F, z_i^F, but since x_i is set to True, both these goods are owned by F_i. Thus, all goods except e_j that D_j values belong to other agents, with there being at least one agent (some T_i or F_i for some $i \in [n]$) who owns two such goods. Clearly D_j will envy this agent (T_i or F_i) after removing one good (d_i^T or d_i^F) from their bundle. Thus, the allocation is not EFX, which contradicts our assumption.

This shows that checking if a fair division instance admits an EFX and PO allocation is NP-hard. □

Lemma 7. *Given a fair division instance $I = (N, M, V)$, checking if I admits an allocation that is both EFX and fPO is NP-complete, even when I is a*

$\{0, a, b\}$*-instance.*

Proof. Notice that our reduction in the proof of Theorem 4 constructs an instance for which every PO allocation is also fPO, since every good is assigned to an agent who values it at the highest among all agents. This shows that checking if an allocation is EFX+fPO is NP-hard, even for $\{0, a, b\}$-instances. Using the fact that checking whether an allocation is fPO or not can be decided in polynomial-time (see for example [15]), we conclude that checking if an allocation is EFX+fPO is in NP. This shows that checking if an allocation is EFX+fPO is NP-complete, for $\{0, a, b\}$-instances as well. □

Appendix C. Missing Proofs from Section 4

Theorem 5. *Given a fair division* $\{a, b\}$*-instance* $I = (N, M, V)$*, an allocation that is both EQX and fPO exists and can be computed in polynomial-time.*

Proof. The proof relies on Lemmas 8 and 9 below.

Lemma 8. *Let* $\mathbf{x}$ *be the allocation returned by Algorithm 2. Then* $\mathbf{x}$ *is EQX and fPO.*

Proof. Similar to Lemma 4, we can show that $\mathbf{x}$ together with the price vector $\mathbf{p}$ on termination forms a market equilibrium. This shows via the First Welfare Theorem that $\mathbf{x}$ is fPO. We can use Lemma 2 to scale all prices to $\{1, k\}$. Suppose $\mathbf{x}$ is not EQX. Then there is an agent h such that for all $j \in x_h$, $v_h(\mathbf{x}_h \setminus \{j\}) > v_i(\mathbf{x}_i)$. If $h \notin C_i$, then the algorithm would not have halted. This means that $h \in C_i$.

Since the algorithm has halted, along all alternating paths $P = (i, j_1, i_1, \ldots, h', j, h)$, it is the case that $v_h(\mathbf{x}_h \setminus \{j\}) \le v_i(\mathbf{x}_i)$. One of two of the following cases must hold:

1. There is some alternating path $P = (i, j_1, i_1, \ldots, h', j, h)$, with $v_{hj} = b$. Then for all $j' \in \mathbf{x}_h$:

$$v_i(\mathbf{x}_i) \ge v_h(\mathbf{x}_h \setminus \{j\}) = v_h(\mathbf{x}_h) - b = v_h(\mathbf{x}_h \setminus \{j'\}),$$

   and for all $j' \in \mathbf{x}_h$ with $v_{hj'} = a$:

$$v_i(\mathbf{x}_i) \ge v_h(\mathbf{x}_h \setminus \{j\}) > v_h(\mathbf{x}_h) - a = v_h(\mathbf{x}_h \setminus \{j'\}),$$

   which means that the EQX condition is satisfied for h.
2. Suppose along all alternating paths $P = (i, j_1, i_1, \ldots, h', j, h)$, it holds that $v_{hj} = a$. If for all goods $j' \in \mathbf{x}_h$ not reachable along any alternating path, it holds that $v_{hj'} = a$, then the EQX condition is satisfied for h:

$$v_i(\mathbf{x}_i) \ge v_h(\mathbf{x}_h \setminus \{j\}) = v_h(\mathbf{x}_h) - a = v_h(\mathbf{x}_h \setminus \{j'\}).$$

   On the other hand suppose there is a good $j' \in \mathbf{x}_h$ that is not reachable from i via any alternating path, with $v_{hj'} = b$. Note that this means there are two goods $j, j' \in \mathbf{x}_h$ s.t. $v_{hj} = a$, and $v_{hj'} = b$. By the MBB condition, we must have $p_j = k$ and $p_{j'} = 1$. Consider any alternating path $P = (i, j_1, i_1, \ldots, h', j, h)$. Then it must be the case that $j' \notin mbb_{h'}$. If $\alpha_{h'}$ is the MBB-ratio of h', then this means:

$$\alpha_{h'} = \frac{v_{h'j}}{p_j} > \frac{v_{h'j'}}{p_{j'}},$$

which gives $v_{h'j} > k v_{h'j'}$ which is not possible when $v_{h'j}, v_{h'j'} \in \{1, k\}$.

This shows that the returned allocation $\mathbf{x}$ is EQX. ☐

Lemma 9. *Algorithm 2 terminates in polynomial-time.*

Proof. Using similar arguments as made in Lemma 5, we can conclude that the number of transfers without a change in the identity of the LU agent is $\mathsf{poly}(n, m)$, and also that the number of identity changes of the LU agent without a price rise step is $\mathsf{poly}(n, m)$. Thus for polynomial run time, we only need to bound the number of price rise steps by a polynomial in n, m.

For this, we observe that the set of EQ1-violators before a price rise step does not increase during the execution of the algorithm. This is because if a new agent becomes an EQ1-violator after a transfer step, she will cease to be an EQ1-violator in subsequent iterations before a price rise event. Also, due to a price-rise step, no new agent can become an EQ1-violator since the utility of all agents remains the same. Thus, the goods belonging to the set of EQ1-violators have not experienced any price rise step, and their prices are either a or b. Just before the price rise step number t, let $\alpha_1^t, \ldots, \alpha_n^t$ be the MBB-ratios of the agents. For any good $j \notin C_{i_0}$, where i_0 is the LU agent, for every agent i, the MBB-ratio of i must at least be the bang-per buck i gets from j. Thus, just before any price rise step t, for every agent i and any good $j \notin C_{i_0}$:

$$\alpha_i^t \geq \frac{v_{ij}}{p_j} \geq \frac{b}{a} \quad . \tag{C.1}$$

Consider the potential function given by:

$$\phi(t) = \sum_{i \in [n]} \log_k \alpha_i^t \quad .$$

Before the first price rise step, all the MBB-ratios are 1. Hence $\alpha_i^1 = 1$ for every $i \in N$. So, $\phi(1) = 0$. From (C.1), we have for all price rise steps, $\phi(t) \geq n \log_k(b/a) = -n$. Also after a price rise step, at least one agent's MBB-ratio value decreases by a factor k, thus $\phi(t+1) \leq \phi(t) - 1$. Thus, the number of price rise steps is at most n. Hence, the algorithm terminates in polynomial-time. ☐

This concludes the proof of Theorem 5. ☐

Theorem 6. *Given a fair division instance $I = (N, M, V)$, checking if I admits an allocation that is both EQX and PO is NP-hard, even when I is a $\{0, a, b\}$-instance.*

Proof. Once again, we reduce from 2P2N3SAT. Let the 2P2N3SAT instance contain n variables $x_1, \ldots, x_n$ and m distinct clauses $C_1, C_2, \ldots, C_m$. Each clause C_j is of the form $\ell_j^1 \vee \ell_j^2 \vee \ell_j^3$, where for each $t \in [3]$, the literal ℓ_j^t is either x_i or $\neg x_i$ for some $i \in [n]$.

Given a 2P2N3SAT-instance: $\{x_i\}_{i \in [n]}, \{C_j\}_{j \in [m]}$, we construct a fair division instance with $2n + 3m$ agents and $7n + 2m$ goods as follows, where $2b < a \leq 3b$. In fact our reduction works for the case of $\{0, 1, 3\}$-instances too.

1. For every variable x_i, create two agents T_i and F_i. Also create 7 goods: $d_i^T, d_i^F, g_i, y_i^T, z_i^T, y_i^F, z_i^F$. Both T_i and F_i value g_i at a. T_i values d_i^T, y_i^T, z_i^T at b, and F_i values d_i^F, y_i^F, z_i^F at b. T_i and F_i value all other goods at 0.
2. For every clause $C_j = \ell_j^1 \vee \ell_j^2 \vee \ell_j^3$, create three agents D_j^1, D_j^2, D_j^3 and two goods p_j, q_j. For $t \in [3]$, each D_j^t values both p_j and q_j at a. If $\ell_j^t = x_i$ for some $i \in [n]$ then D_j^t values y_i^T, z_i^T at a; and if $\ell_j^t = \neg x_i$ then D_j^t values y_i^F, z_i^F at a. D_j^t values all other goods at 0.

Suppose the formula has a satisfying assignment. Using this satisfying assignment, we create an allocation of goods as follows: For every x_i set to True, we assign g_i, d_i^T to T_i and d_i^F, y_i^F, z_i^F to F_i. Additionally, we assign y_i^T and z_i^T to the agents D_j^t for which $\ell_j^t = x_i$. Analogously, for every x_i set to False, we assign g_i, d_i^F to F_i and d_i^T, y_i^T, z_i^T to T_i. Further, we assign y_i^F and z_i^F to the agents D_j^t for which $\ell_j^t = \neg x_i$. We also assign p_j, q_j to agents in $\{D_j^1, D_j^2, D_j^3\}$ for every $j \in [m]$, so that every agent D_j^t gets utility of at least a. This is possible since the assignment satisfies all clauses, meaning that there exists at least one True literal in every clause. Notice that every item has been assigned to an agent that values it at the highest among all agents. Thus this allocation maximizes the utilitarian social Welfare, and is PO. Further, it is also EQX. This is because under this assignment for every $i \in [n]$, if x_i is True, T_i gets a utility of $a + b$, and F_i gets a utility of $3b$. If x_i is False, T_i gets a utility of $3b$, and F_i gets a utility of $a + b$. The utility of any agent D_j^t is either a or $2a$. Since $2b < a \leq 3b$, we see that the allocation is EQX.

Suppose on the other hand every assignment is unsatisfying, i.e., under any assignment there is always some clause C_j all of whose literals are False. Suppose there exists an EQX and PO allocation of goods to agents. Since the allocation is PO, for every $i \in [n]$, d_i^T must be assigned to T_i, d_i^F must be assigned to F_i, and g_i must be assigned to either T_i or F_i. Also, for each $j \in [m]$, p_j and q_j must be assigned to agents in $\{D_j^1, D_j^2, D_j^3\}$. Suppose for some $i \in [n]$, g_i is allocated to T_i. The utility of T_i is at least $a + b$. Since the allocation is EQX, F_i must have a utility of at least a, which equals T_i's utility after removing d_i^T of value b. Since $a > 2b$, this is only possible if F_i receives d_i^F, y_i^F and z_i^F, thus obtaining a total value of $3b$. Similarly, if g_i is allocated to F_i, then for the allocation to be EQX, y_i^T and z_i^T must be allocated to T_i. Now consider the assignment defined from the allocation in the following manner: if g_i is allocated to T_i, set x_i to True, and if g_i is allocated to F_i, set x_i to False. By our assumption, this assignment leaves at least one clause C_j unsatisfied, which means all literals in that clause evaluate to False. If for some $t \in [3]$, $\ell_j^t = x_i$, then D_j^t values y_i^T, z_i^T. However since x_i is False, these goods have been assigned to T_i. Similarly if $\ell_j^t = \neg x_i$, then D_j^t values y_i^F, z_i^F. However since x_i is True, these goods have been assigned to F_i. We conclude that the three agents D_j^1, D_j^2, D_j^3 can receive only the goods p_j and q_j, implying that some agent will get a utility of 0. Thus the allocation cannot be EQX, which contradicts our assumption.

This shows that checking if a fair division instance admits an EQX and PO allocation is NP-hard. □

Similar to Lemma 7, we can show that:

Lemma 10. *Given a fair division instance $I = (N, M, V)$, checking if I admits an allocation that is both EQX and fPO is NP-complete, even when I is a $\{0, a, b\}$-instance.*

Appendix D. Missing Proofs from Section 5

Theorem 7. *Given a fair division $\{a, b\}$-instance $I = (N, M, V)$, where $a/b > m - n$, an MNW allocation can be computed in polynomial-time.*

Proof. Suppose we are given a $\{a, b\}$-instance (N, M, V) with $k = a/b > m - n$. Since Nash welfare is scale-invariant we can scale the values to $\{1, k\}$. We can also assume without loss of generality that $m > n$, since if $m \leq n$, then an MNW allocation $\mathbf{x}$ can be computed by solving a weighted maximum matching problem. Since all agents value all goods positively, $\mathsf{NW}(\mathbf{x}) > 0$. Further we can assume that for every agent $i \in N$, there is some good $j \in M$, such that $v_{ij} = k$, since if for some agent i, $v_{ij} = 1$ for all $j \in M$, then we could rescale her values and set $v_{ij} = k$ for all $j \in M$. For a set of agents $S \subseteq N$, let $\mathbf{x}_S = \bigcup_{i \in S} \mathbf{x}_i$ be the set of goods owned by agents in S.

The main idea in computing an MNW allocation is that when $k > m - n$, any agent prefers a single good that they value at k over any number of goods they value at 1. We use this idea to show several properties of an MNW allocation $\mathbf{x}$. Partition the set of agents into L sets depending on the number of goods of value k they own in $\mathbf{x}$. Specifically, let $N = N_1 \sqcup \cdots \sqcup N_L$, such that there exist numbers $m_1 < \cdots < m_L \in \mathbb{Z}_{\geq 0}$, such that for any $1 \leq \ell \leq L$, for all agents $i \in N_\ell$ it holds that $|\{j \in \mathbf{x}_i : v_{ij} = k\}| = m_\ell$. We show:

Lemma 11. *There exists $r \in \mathbb{Z}_{\geq 0}$ s.t. for all $i \in N_1$, $v_i(\mathbf{x}_i) = m_1 k + r_i$, where $r_i \in \{r, r+1\}$.*

Proof. Let $i \in \mathsf{argmin}_{i \in N} v_i(\mathbf{x}_i)$, and let $r = v_i(\mathbf{x}_i) - m_1 k$. Suppose there exists $h \in N_1$ s.t. $v_h(\mathbf{x}_h) - m_1 k \geq r + 2$. Then on transferring one good $j \in \mathbf{x}_h$ with $v_{hj} = 1$, we can observe that the NW strictly increases, contradicting the fact that $\mathbf{x}$ is Nash optimal. □

Lemma 12. *For all $i \notin N_1$, and for all $j \in \mathbf{x}_i$, $v_{ij} = k$.*

Proof. Suppose for some $\ell \geq 2$ there is some agent $h \in N_\ell$ and some good $j \in \mathbf{x}_h$ with $v_{hj} = 1$. Consider an agent $i \in \mathsf{argmin}_{i' \in N_1} v_{i'}(\mathbf{x}_{i'})$, and let $v_i(\mathbf{x}_i) = m_1 k + r$ for some $r \in \mathbb{Z}_{\geq 0}$. Since $\ell \geq 2$, $m_\ell \geq m_1 + 1$, and hence

$$v_h(\mathbf{x}_h) \geq m_\ell k + 1 \geq (m_1 + 1)k + 1 \ .$$

Note that each agent other than i owns at least one good, and h owns at least two goods. Hence i can own at most $m - n$ goods. Thus $r \leq m - n < k$, and $v_i(\mathbf{x}_i) + 1 < m_1 k + k + 1 \leq v_h(\mathbf{x}_h)$. Now consider the allocation $\mathbf{x}'$ obtained on transferring j from h to i. Then $v_h(\mathbf{x}'_h) = v_h(\mathbf{x}_h) - 1$ and $v_i(\mathbf{x}'_i) \geq v_i(\mathbf{x}_i) + 1$. We have:

$$\begin{aligned} &v_i(\mathbf{x}'_i) v_h(\mathbf{x}'_h) > v_i(\mathbf{x}_i) v_h(\mathbf{x}_h) \\ &\iff v_h(\mathbf{x}_h) > v_i(\mathbf{x}_i) + 1 \ , \end{aligned}$$

which was shown to be true, contradicting the fact that $\mathbf{x}$ is Nash optimal. □

Lemma 13. *Suppose $m_1 \geq 1$. Consider any good $j \in \mathbf{x}_i$ for an agent $i \in N_1$ with $v_{ij} = 1$. Then for every agent $h \in N_1$ it must be the case that $v_{hj} = 1$.*

Proof. Suppose for sake of contradiction there is some $j \in \mathbf{x}_i$ for $i \in N_1$ with $v_{ij} = 1$, and some $h \in N$ with $v_{hj} = k$. First let us assume $h \in N_\ell$ for $\ell \geq 2$. Since every agent owns at least one good and i owns at least two (since $m_1 \geq 1$), h can own at most $m - n$ goods. Thus $m_\ell \leq m - n < k$. This means that $v_h(\mathbf{x}_h) + k = m_\ell k + k \leq k^2 + k \leq m_1 k^2 + k = k(m_1 k + 1) \leq k v_i(\mathbf{x}_i)$. Consider the allocation $\mathbf{x}'$ obtained on transferring j from i to h. Then $v_i(\mathbf{x}'_i) = v_i(\mathbf{x}_i) - 1$ and $v_h(\mathbf{x}'_h) = v_h(\mathbf{x}_h) + k$. We have:

$$\begin{aligned} &v_i(\mathbf{x}'_i) v_h(\mathbf{x}'_h) > v_i(\mathbf{x}_i) v_h(\mathbf{x}_h) \\ &\iff k v_i(\mathbf{x}_i) > v_h(\mathbf{x}_h) + k, \end{aligned}$$

which was shown to be true, contradicting the fact that $\mathbf{x}$ is Nash optimal.

Now suppose $h \in N_1$. If there is some good $j' \in \mathbf{x}_h$ with $v_{hj'} = 1$, then on swapping j and j' among the agents i and h, observe that the NW strictly increases. If there is no such j', then transferring j from i to h strictly increases the NW, again contradicting the fact that $\mathbf{x}$ is Nash optimal. □

Consider a bipartite graph $G = (N, M, E)$ with edges between the agents N on one side and M on the other, where $(i, j) \in E$ iff $v_{ij} = k$. Then:

Lemma 14. *G has a matching of size n iff $m_1 \geq 1$.*

Proof. The reverse direction is obvious. If $m_1 \geq 1$, then in the allocation every agent gets a good that she values at k. This constitutes a matching in G of size n.

On the other hand, suppose G has a matching of size n and yet $m_1 = 0$ in an MNW allocation $\mathbf{x}$. Now consider walks starting from an agent $i_0 \in N_1$ of the form $(i_0, j_1, i_1, \ldots, j_s, i_s)$, where for every $1 \le r \le s$, $v_{i_{r-1}j_r} = k$ and $j_r \in \mathbf{x}_{i_r}$. Note that such a walk exists because for we ensure the valuations are such that for any agent there is some good she values at k. Suppose there is some such walk in which there is some agent $i_s \notin N_1$ who owns two goods j_s and j_{s+1} which she values at k. Then on transferring good j_r to i_{r-1} for each $r \in [s]$, we observe that the NW strictly increases as $v_{i_0}(\mathbf{x}_{i_0}) < k$, and $v_{i_r}(\mathbf{x}_{i_r}) \ge 2k$. Similarly, if there is a walk that is actually a cycle, i.e., i_s values a good $j_0 \in \mathbf{x}_{i_0}$ at k, then upon transferring goods along the cycle the NW strictly improves. Finally suppose for every such walk starting with an agent $i_0 \in N_1$, terminates at an agent i_s who is either in N_1 or is in N_2 and owns only one good j_s that she values at k. Consider the set S of agents which are part of such walks. Then by the property of such walks, agents in S collectively like at most $|S| - 1$ goods at value k. That is, Hall's condition is violated for the set S in the graph G. However this contradicts the fact that G has a matching of size n, hence disproving the assumption that $m_1 = 0$. □

Let $M_{high} = \{j \in M : \exists i \in N \text{ s.t. } v_{ij} = k\}$ be the set of goods valued at k by at least one agent, and let $M_{low} = \{j \in M : \forall i \in N : v_{ij} = 1\} = M \setminus M_{high}$ be the set of goods valued at 1 by all agents. Suppose G has a matching of size n. From Lemma 13 it is clear that all goods owned by agents in N_1 that are valued at 1, are also valued at 1 by any other agent, i.e., they belong to M_{low}. Consider the allocation $\mathbf{x}'$ given by $\mathbf{x}'_i = \mathbf{x}_i \cap M_{high}$ for all agents i. Then from Lemma 14, every agent gets at least one good they value at k in $\mathbf{x}'$. From Lemma 12, every agent owns only goods they value at k in $\mathbf{x}'$. Thus, it is clear that $\mathbf{x}'$ maximizes the NW for the instance $I' = (N, M_{high}, V')$, where for any agent $i \in N$ and good $j \in M_{high}$, $v'_{ij} = \lfloor v_{ij}/k \rfloor \in \{0, 1\}$. Since I' is a binary-valued instance, using the algorithm BinaryMNW of [20] we can compute $\mathbf{x}'$. Then using Lemma 11, $\mathbf{x}$ can be obtained from $\mathbf{x}'$ by allocating M_{low} to agents in N_1 in a round-robin fashion, which is done by the procedure RoundRobin. This corresponds to lines 5-11 of Algorithm 3.

We now consider the case of $m_1 = 0$. By Lemma 14 we know that there is no matching of size n in G. Let us now define the sets: $N_2^{t+1} = \{h \in N : \exists i \in N_2^t, \exists j \in \mathbf{x}_h \text{ s.t. } v_{ij} = k\}$ for $t \ge 0$, where $N_2^0 = N_1$. We show:

Lemma 15. *Suppose $m_1 = 0$. Then $N_2^t \subseteq N_2$ for $t \ge 1$ and $m_2 = 1$.*

Proof. Suppose for some $T \ge 1$ and $i_t \in N_2^T$ there is some agent $h \in N$, s.t. for some good $j_{T+1} \in \mathbf{x}_h$ s.t. $v_{hj_{T+1}} = k$. By definition for $0 \le t \le T$ there exist agents $i_t \in N_2^t$ and goods $j_{t+1} \in \mathbf{x}_{i_{t+1}}$ such that $j_{t+1} \in \mathbf{x}_{t+1}$ and $v_{i_t j_{t+1}} = k$ for all $0 \le t \le T$, where $h = i_{T+1}$. Let $\mathbf{x}'$ be the allocation obtained from $\mathbf{x}$ by transferring good j_t to i_{t-1} for each $t \in [T+1]$. We have:

$$v_{i_0}(\mathbf{x}'_{i_0}) v_h(\mathbf{x}'_h) > v_{i_0}(\mathbf{x}_{i_0}) v_h(\mathbf{x}_h)$$
$$\iff v_h(\mathbf{x}_h) > v_{i_0}(\mathbf{x}_{i_0}) + k \ .$$

As shown before, $v_{i_0}(\mathbf{x}_{i_0}) \le m - n < k$. Hence $v_{i_0}(\mathbf{x}_{i_0}) + k < 2k$. Now see that $v_h(\mathbf{x}_h) = m_\ell k$. If either $\ell \ge 3$, or $\ell = 2$ and $m_2 = 2$, then $v_h(\mathbf{x}_h) \ge 2k >$

$v_{i_0}(\mathbf{x}_{i_0}) + k$, which means $\mathbf{x}$ is not Nash optimal. Now suppose $\ell = 1$. Then performing the transfer of goods as before, and additionally transferring a good $j_0 \in \mathbf{x}_{i_0}$ to i_{T+1} improves the NW since the valuation of i_0 improves. This shows that $N_2^t \subseteq N_2$ for all $t \geq 1$, and $m_2 = 1$. $\square$

Let $\bigcup_{r\geq 1} N_2^r = N_2^*$, which is a union of at most n sets since $N_2^* \subseteq N_2$. Further, Lemma 15 also shows that set of goods valued at k by some agent in N_1 or N_2^* is owned by agents in N_2^* only. That is, $\{j : v_{ij} = k, i \in N_1 \cup N_2^*\} = \mathbf{x}_{N_2^*}$. From Lemma 12, all agents not in N_1 own at least one good they value at k. This shows that $N_1 \cup N_2^*$ is a Hall's violator set S of agents in G. If $\mathcal{M}$ is the maximum matching in G, then N_1 comprises of agents who do not get matched under $\mathcal{M}$, and N_2^* is the set of agents in S who are matched under $\mathcal{M}$.

We now show how an MNW allocation $\mathbf{x}$ can be computed. Let $\mathcal{M}$ be the maximum matching and let P be the set of agents that are unmatched under $\mathcal{M}$. The set P corresponds to the set N_1 in the preceding analysis. Analogous to how N_2^* was computed, compute the set Q, and define $S = P \cup Q$ to be a Hall's violator set of agents. Let T be the set of neighbors of S in G. Notice that in any MNW allocation, the agents in Q get only a single good, which they value at k. Further agents in S value only goods in T at k. This means that $\mathbf{x}'$ actually maximizes the NW for the instance I', where $I' = (N \setminus Q, M \setminus T, V)$, and $\mathbf{x}'$ is the allocation $\mathbf{x}$ but restricted to agents in $N \setminus Q$. Notice from Lemma 12, all agents apart from P get only goods that they value at k. Also since agents in P value all goods not in T at 1, in fact $\mathbf{x}'$ maximizes the NW for the instance $I'' = (N \setminus Q, M \setminus T, V')$, where:

$$v'_{ij} = \begin{cases} \lfloor v_{ij}/k \rfloor \text{ if } i \in N \setminus S \\ 1 \text{ if } i \in P \end{cases},$$

for all $i \in N \setminus Q$ and $j \in M \setminus T$. Since I'' is now a binary instance, we can compute an MNW allocation $\mathbf{x}'$ in polynomial-time. Then, $\mathbf{x}$ can be obtained by $\mathbf{x}'$ by simply assigning to the agents in Q the goods in T they were matched to under $\mathcal{M}$. This corresponds to lines 12-20 of Algorithm 3.

To conclude, Algorithm 3 computes an MNW allocation for $\{a, b\}$-instances with $a/b > m - n$ in polynomial-time. $\square$

Theorem 8. *Given a fair division $\{a, b\}$-instance $I = (N, M, V)$, any MNW allocation is fPO. This result is sharp; an MNW allocation of a 3-valued instance need not be fPO.*

Proof. We do this by showing that there are prices $\mathbf{p}$ for the goods s.t. $(\mathbf{x}, \mathbf{p})$ is on MBB. We further require that all $p_j \in \{1, k\}$ by invoking Lemma 2. For a set of agents $S \subseteq N$, let $\mathbf{x}_S = \bigcup_{i\in S} \mathbf{x}_i$ be the set of goods owned by agents in S.

For this we present the price-setting algorithm, Algorithm 4.

Algorithm 4 terminates in polynomial-time since it sets the price of each good exactly once, and in every iteration of the loops the price of at least one good is set. Let $\mathbf{p}$ be the prices of the goods as set by Algorithm 4. Note that

Algorithm 4 Price-setting algorithm

Input: MNW allocation $\mathbf{x}$ for an $\{a, b\}$-instance (N, M, V)

Output: Prices $\mathbf{p}$ of goods s.t. $(\mathbf{x}, \mathbf{p})$ is on MBB

```
1: $k \leftarrow a/b$
2: $N_1 = \{i \in N : \exists j, j' \in \mathbf{x}_i$ s.t. $v_{ij} = k, v_{ij'} = 1\}$
3: $N_2 = \{i \in N : \forall j \in \mathbf{x}_i, v_{ij} = k\}$
4: $N_3 = \{i \in N : \forall j \in \mathbf{x}_i, v_{ij} = 1\}$.
5: for all $j \in \mathbf{x}_{N_1}$ do
6:     $p_j \leftarrow v_{ij}$, if $j \in \mathbf{x}_i$ for $i \in N_1$
7: for all $j \in x_{N_3}$ do
8:     $p_j \leftarrow 1$
9: repeat
10:     $N_2' \leftarrow \{i \in N_2 : \exists j \notin \mathbf{x}_i$ s.t. $p_j = 1, v_{ij} = k\}$.
11:     for all $j \in \mathbf{x}_{N_2'}$ do
12:         $p_j \leftarrow 1$
13:     $N_2 \leftarrow N_2 \setminus N_2'$
14: until $N_2' \neq \emptyset$
15: for all $j \in \mathbf{x}_{N_2}$ do                    ▷ At this point, $N_2' = \emptyset$
16:     $p_j \leftarrow k$
```

the MBB ratios of agents in N_1 or N_3 are 1. The MBB ratio of agents in N_2 can be either 1 or k.

Now we claim that $(\mathbf{x}, \mathbf{p})$ is on MBB. Suppose for the sake of contradiction for some agent i the allocation is not on MBB. We consider two cases:

1. Agent $i \in N_1 \cup N_3$ is not on MBB. Since the MBB ratio of agent i, $\alpha_i = 1$, there must be some good $j \in \mathbf{x}_h \setminus \mathbf{x}_i$ for some agent h s.t. $v_{ij}/p_j > 1$, which means $v_{ij} = k$ and $p_j = 1$. Suppose $h \in N_2$. Since $p_j = 1$, by our Algorithm 4 this happens only if there is some $j_1 \in \mathbf{x}_{h_1} \setminus \mathbf{x}_h$ s.t. $p_{j_1} = 1$, $v_{hj_1} = k$. Further, suppose $h_1 \in N_2$. We continue the same reasoning and arrive at a sequence of goods and agents $(i, j = j_0, h = h_0, j_1, h_1, \ldots, j_\ell, h_\ell)$, s.t. $j_{\ell'} \in \mathbf{x}_{h_{\ell'}}$, $p_{j_{\ell'}} = 1$, for all $0 \leq \ell' \leq \ell$. Further $v_{h_{\ell'} j_{\ell'}} = k$ and $h_{\ell'} \in N_2$, for all $0 \leq \ell' \leq \ell - 1$. Also $v_{h_\ell j_\ell} = 1$ and $h_\ell \in N_1 \cup N_3$. Such a path exists by the price-setting algorithm and because the number of agents in N_2 is bounded by n. Now transferring $j_{\ell'}$ to $h_{\ell'-1}$ for all $1 \leq \ell' \leq \ell$, j to i, and some good $j' \in \mathbf{x}_i$ with $v_{ij'} = 1$ to h_ℓ improves the Nash welfare, leading to a contradiction. Note that this analysis also subsumes the cases of $h \in N_1$ and $h \in N_3$. Therefore for no agent $i \in N_1 \cup N_3$ the MBB condition is violated.
2. Agent $i \in N_2$ is not on MBB. If $\alpha_i = k$ then the MBB condition cannot be violated for i. Then it must mean that $\alpha_i = 1$, and that there is a good $j \notin \mathbf{x}_i$ with $p_j = 1$ and $v_{ij} = k$ which causes the MBB condition of i to be violated. However, in this case, Algorithm 4 would have set the prices of goods in $\mathbf{x}_i$ to 1 instead of k (from Line 12, since i would belong to N_2' by

Line 10), causing α_i to equal k instead of 1. This means that the MBB condition of no agent $i \in N_2$ is violated.

In other words, the allocation $(\mathbf{x}, \mathbf{p})$ is on MBB. Thus, any MNW allocation for $\{a, b\}$-instances is fPO.

For sharpness, consider the 3-valued instance of Table D.4.

Table D.4: A 3-valued instance for which MNW allocation is not fPO

	g_0	g_1	g_2
A_1	5	2	1
A_2	5	1	2

The MNW allocations are $\mathbf{x} = (\{g_0\}, \{g_1, g_2\})$ and $\mathbf{y} = (\{g_1, g_2\}, \{g_0\})$ with Nash welfare $\sqrt{5 \cdot 3} = \sqrt{15}$. However it is easy to see that in any fPO allocation $\mathbf{z}$, $g_i \in \mathbf{z}_i$ for $i \in [2]$. Such an allocation has Nash welfare only $\sqrt{7 \cdot 2} = \sqrt{14}$. Hence MNW allocations need not be fPO for k-valued instances for $k \geq 3$. $\square$

Theorem 9. *Given a fair division instance $I = (N, M, V)$, it is NP-hard to approximate the MNW to a factor better than 1.00019, even for $\{0, a, b\}$-instances.*

Proof. We consider a 2P2N3SAT-instance: $\{x_i\}_{i \in [n]}, \{C_j\}_{j \in [m]}$, where $3m = 4n$. There exist constants ρ_1, ρ_2 such that it is NP-hard to decide if there is an assignment with $\geq \rho_1 m$ clauses satisfied, or if in all assignments at most $\leq \rho_2 m$ clauses are satisfied [41]. Here, $\rho_1 = (1016 - \epsilon)/1016, \rho_2 = (1015 + \epsilon)/1016$, for small $\epsilon > 0$.

For each variable x_i, we create two agents T_i, F_i and one good g_i which is valued at 2 by both T_i, F_i. For each clause C_j, we create a good h_j which is valued at 1 by agent A_i if setting $x_i = A$ makes C_j true, for $A \in \{T, F\}$. We also create $2n - m$ dummy goods $\{d_j\}_{j \in [2n-m]}$ which are valued at 1 by all agents. All other values are 0. Let G be the collection of g_i's, H of h_j's and D of d_j's. Overall, there are $2n$ agents and $3n$ goods.

Suppose that there is an assignment with $\ell \geq \rho_1 m$ satisfied clauses. Then we construct an allocation as follows. For every variable x_i set to True, assign g_i to F_i, and for every variable x_i set to False, assign g_i to T_i. Let P be the set of n agents who get a good from G, and Q be the set of agents who don't. For each satisfied clause C_j, give h_j to an agent in Q who has positive value for it. Thus ℓ goods from H are allocated. Suppose p agents get utility 2 from goods in H. Then $\ell - 2p$ agents get utility of 1 from H, and $n - \ell + p$ agents get 0 utility from H (since each h_j gives utility 1). Corresponding to each unsatisfied clause C_j, the remaining $m - \ell$ goods from H must be assigned to an agent in P who already has some good from G. We now allocate goods from D to agents in Q to try and "balance" the agents' utilities so that agents get a utility of 2. The "demand" for goods of value 1 is $2(n - \ell + p) + \ell - 2p = 2n - \ell$, and the "supply" is $2n - m$. The shortage is $m - \ell$, hence $n - m + \ell$ agents in Q get a utility of 2, and $m - \ell$ agents in Q get a utility of 1. Let $H' \subseteq H$ be the goods h_j corresponding to unsatisfied clauses C_j must be allocated to agents in P. Consider a bipartite graph with goods H' on one side and agents P on the other side, with an edge from a good to an agent if the agent has value 1 for the good. Here, H' corresponds to unsatisfied clauses, and P corresponds to False literals. Since each unsatisfied clause contains three literals which are False under the given assignment, in the graph each good in H' will have edges to exactly three agents in P in this graph. Further we know that all clauses contain distinct literals, and all clauses are distinct. Thus any $W \subseteq H'$ has at least $|W|$ outgoing edges. By Hall's Marriage Theorem, this graph has a matching saturating H', further $|H'| = m - \ell \leq (1 - \rho_1)m \leq n = |P|$, since $\rho_1 \geq 1/4$. We then allocate goods in H' to their matched agents. This completes the allocation, and we have exactly $m - \ell$ agents in P getting a utility of 3. This implies a lower bound MNW^L on the maximum Nash welfare:

$$\begin{aligned} MNW^L &= \left(3^{m-\ell} \cdot 2^{2n-2m+2\ell} \cdot 1^{m-\ell}\right)^{\frac{1}{2n}} \\ &= \left(3^m \cdot 4^{n-m} \cdot \left(\frac{4}{3}\right)^{\ell}\right)^{\frac{1}{2n}} \\ &\geq \left(3^m \cdot 4^{n-m}\right)^{\frac{1}{2n}} \cdot \left(\frac{4}{3}\right)^{\frac{\rho_1 m}{2n}} . \end{aligned} \tag{D.1}$$

Now suppose under all assignments at most $\rho_2 m$ clauses are satisfied. Consider the Nash optimal allocation. Since it is PO, every g_i is assigned to either T_i or F_i. Let P be the set of n agents owning a good from G, and Q the set of n agents who don't get goods from G. Agents in P get utility of at least 2. Consider an assignment defined as follows: for every g_i assigned to T_i, set x_i to False; and for every g_i assigned to F_i, set x_i to True. Let ℓ be the number of satisfied clauses. Consider the set H' of goods h_j corresponding to unsatisfied clauses C_j. Any good $h_j \notin H'$ must be allocated to an agent A_i s.t. setting $x_i = A$ makes C_j true, for $A \in \{T, F\}$. Clearly, this agent A_i belongs to Q. Let p be the number of agents in Q which get utility of 2 from goods in $H \setminus H'$. Then $\ell - 2p$ agents from Q get a utility of 1, and $n - \ell + p$ agents from Q get 0 utility from goods in $H \setminus H'$. The goods from D must be allocated to agents only in Q, since it is sub-optimal (for maximizing NW) to give a good d_j to an agent in P who then gets a utility of 3, while some agent in Q has a utility of at most 1. Hence by a similar argument as the previous part, exactly $m - \ell$ agents in Q get utility of 1 in the optimal allocation. Now consider goods in H'. These goods must be allocated to agents in P, who also own some good from G. To get an upper bound on Nash welfare, we can consider the best case where each good in H' is given to exactly one agent in P. Thus $m - \ell$ agents in P get a utility of 3, and rest of the agents in P get utility 2. This gives an upper bound

MNW^U on the maximum Nash welfare:

$$\begin{aligned} MNW^U &= \left(3^{m-\ell} \cdot 2^{2n-2m+2\ell} \cdot 1^{m-\ell}\right)^{\frac{1}{2n}} \\ &= \left(3^m \cdot 4^{n-m} \cdot \left(\frac{4}{3}\right)^{\ell}\right)^{\frac{1}{2n}} \\ &\leq \left(3^m \cdot 4^{n-m}\right)^{\frac{1}{2n}} \cdot \left(\frac{4}{3}\right)^{\frac{\rho_2 m}{2n}} . \end{aligned} \tag{D.2}$$

Thus from Equations D.1 and D.2, we cannot approximate the Nash welfare to a factor better than:

$$\alpha = \frac{MNW^L}{MNW^U} = \left(\frac{4}{3}\right)^{\frac{(\rho_1-\rho_2)m}{2n}} ,$$

unless P = NP. With the best known values for ρ_1, ρ_2 from [41], we get $\alpha \approx 1.00019$, which is better than the factor 1.00008 of [17], which applies for 5-valued instances. An APX-hardness result for 3-valued instances was not known until now. The best hardness factor for the general MNW problem is 1.069 due to [24], however this only applies to 4-valued instances. □