Maximizing Nash Social Welfare in 2-Value Instances

Hannaneh Akrami,[1] Bhaskar Ray Chaudhury,[2] Martin Hoefer,[3] Kurt Mehlhorn,[4] Marco Schmalhofer,[3] Golnoosh Shahkarami,[1] Giovanna Varricchio,[3] Quentin Vermande,[5] Ernest van Wijland[5]

[1] Max Planck Institute for Informatics, Graduiertenschule Informatik, Universität des Saarlandes
[2] University of Illinois, Urbana-Champaign, Department of Computer Science
[3] Goethe University Frankfurt, Institute for Computer Science
[4] Max Planck Institute for Informatics, Universität des Saarlandes
[5] École Normale Supérieure, Paris
{hakrami, mehlhorn, gshahkar}@mpi-inf.mpg.de, braycha@illinois.edu, {mhoefer, varricchio, schmalhofer}@em.uni-frankfurt.de , {quentin.vermande, ernest.van.wijland}@ens.fr

Abstract

We consider the problem of maximizing the Nash social welfare when allocating a set $\mathcal{G}$ of indivisible goods to a set $\mathcal{N}$ of agents. We study instances, in which all agents have 2-value additive valuations: The value of every agent $i \in \mathcal{N}$ for every good $j \in \mathcal{G}$ is $v_{ij} \in \{p, q\}$, for $p, q \in \mathbb{N}$, $p \le q$. In this work, we design an algorithm to compute an optimal allocation in polynomial time if p divides q, i.e., when $p = 1$ and $q \in \mathbb{N}$ after appropriate scaling. The problem is NP-hard whenever p and q are coprime and $p \ge 3$.

In terms of approximation, we present positive and negative results for general p and q. We show that our algorithm obtains an approximation ratio of at most 1.0345. Moreover, we prove that the problem is APX-hard, with a lower bound of 1.000015 achieved at $p/q = 4/5$.

Introduction

Fair division is an important area at the intersection of economics and computer science. While fair division with divisible goods is relatively well-understood in many contexts, the case of *indivisible* goods is significantly more challenging. Recent work in fair division has started to examine extensions of standard fairness concepts such as envy-freeness to notions such as EF1 (envy-free up to one good) (Lipton et al. 2004) or EFX (envy-free up to any good) (Caragiannis et al. 2016), most prominently in the case of non-negative, additive valuations of the agents. In this additive domain, notions of envy-freeness are closely related to the *Nash social welfare (NSW)*, which is defined by the geometric mean of the valuations. An allocation maximizing the Nash social welfare is Pareto-optimal, satisfies EF1 (Caragiannis et al. 2016) and in some cases even EFX (Amanatidis et al. 2020). An important question is, thus, if we can efficiently compute or approximate an allocation that maximizes NSW. This is the question we study in this paper.

More formally, we consider an allocation problem with a set $\mathcal{N}$ of n agents and a set $\mathcal{G}$ of m indivisible goods. Each agent $i \in \mathcal{N}$ has a valuation function $v_i : 2^{\mathcal{G}} \to \mathbb{Q}_{\ge 0}$. We assume all functions to be non-negative, monotone, and normalized to $v_i(\emptyset) = 0$. For convenience, we assume every v_i maps into the rational numbers. The goal is to find an allocation of the goods $\mathcal{A} = (A_1, \dots, A_n)$ to maximize the Nash social welfare, i.e., the geometric mean of the valuations

$$\text{NSW}(\mathcal{A}) = \left(\prod_{i=1}^{n} v_i(A_i)\right)^{1/n} .$$

Of particular interest are the instances admitting strictly positive NSW; clearly, in this case, an allocation that maximizes the NSW is Pareto-optimal. By maximizing the NSW, we strike a balance between maximizing the utilitarian social welfare $\sum_i v_i(A_i)$ and the egalitarian social welfare $\min_i v_i(A_i)$. Notably, optimality and approximation ratio for NSW are invariant to scaling each valuation $v_i(A_i)$ by an agent-specific parameter $c_i > 0$. This is yet another property that makes NSW an attractive objective function for allocation problems. It allows a further normalization – we can assume every $v_i : 2^{\mathcal{G}} \to \mathbb{N}_0$ maps into the *natural* numbers.

Finding desirable approximation algorithms for maximizing the NSW has become an active field of research only recently. For instances with additive valuations, where $v_i(A) = \sum_{j \in A} v_{ij}$ for every $i \in \mathcal{N}$, in a series of papers (Cole et al. 2017; Cole and Gkatzelis 2018; Anari et al. 2017; Barman, Krishnamurthy, and Vaish 2018a) several algorithms with constant approximation factors were obtained. The currently best factor is $e^{1/e} \approx 1.445$ (Barman, Krishnamurthy, and Vaish 2018a). The algorithm uses prices and techniques inspired by competitive equilibria, along with suitable rounding of valuations to guarantee polynomial running time.

Even for identical additive valuations, (i.e., $v_{ij} = v_j$ for all $i \in \mathcal{N}$, $j \in \mathcal{G}$,) the problem is NP-hard, and a greedy algorithm with factor of 1.061 (Barman, Krishnamurthy, and Vaish 2018b) as well as a PTAS (Nguyen and Rothe 2014) were obtained. In terms of inapproximability, the best known lower bound for additive valuations is $\sqrt{8/7} \approx 1.069$ (Garg, Hoefer, and Mehlhorn 2018). Notably, this lower bound applies even in the case when the additive valuation is composed of only three values with one of them being 0 (i.e.,

$v_{ij} \in \{0, p, q\}$ for all $i \in \mathcal{N}$, $j \in \mathcal{G}$, where $p, q \in \mathbb{N}$). For the case of two values with one 0 (i.e., $v_{ij} \in \{0, q\}$ for all $i \in \mathcal{N}$, $j \in \mathcal{G}$), an allocation maximizing the NSW can be computed in polynomial time (Barman, Krishnamurthy, and Vaish 2018b).

Contribution and Results. In this paper, we consider computing allocations with (near-)optimal NSW when every agent has a *2-value* valuation. In such an instance, $v_{ij} \in \{p, q\}$ for every $i \in \mathcal{N}$ and $j \in \mathcal{G}$, where $p, q \in \mathbb{N}_0$. Notably, in 2-value instances any optimal allocation satisfies EFX, which is not true when agents have 3 or more values (Amanatidis et al. 2020). The case $p = q$ is trivial. An optimal allocation can be computed in polynomial time when $p = 0 < q$ (Barman, Krishnamurthy, and Vaish 2018b). Hence, we concentrate on the case $1 \leq p < q$. We design a polynomial-time algorithm to find an optimal allocation when p divides q, i.e., after appropriate scaling, when $p = 1$ and $q \in \mathbb{N}$. Even if p does not divide q, the algorithm still guarantees an approximation factor of at most 1.0345. This is significantly lower than the constant factors obtained for general additive valuations (Cole et al. 2017; Cole and Gkatzelis 2018; Barman, Krishnamurthy, and Vaish 2018a). An approximation algorithm for 2-value instances with approximation factor 1.061 has been obtained in (Garg and Murhekar 2021). The algorithm is based on ideas from competitive equilibria. Our algorithm is a greedy procedure and improves this guarantee.

Complementing these positive results, we also prove new hardness results for 2-value instances. Maximizing the NSW is NP-hard whenever p and q are coprime and $p \geq 3$. Since for $p = 1$ we have a polynomial-time algorithm, $p = 2$ remains as an interesting open problem. Maximizing the NSW in 2-value instances is APX-hard. Our reduction from Gap-4D-Matching avoids the use of utilities $v_{ij} = 0$, which poses a substantial technical challenge over the more direct reduction for 3-value instances in (Garg, Hoefer, and Mehlhorn 2018). Our lower bound on the approximation factor is 1.000015 for $p/q = 4/5$. This answers an open problem from (Amanatidis et al. 2020).

Due to space constraints, all the missing proof can be found in the full version of the paper.

Related Work

Other than additive valuations, the design of approximation algorithms for maximizing NSW with *submodular valuations* has been subject to significant progress very recently. While small constant approximation factors have been obtained for special cases (Garg, Hoefer, and Mehlhorn 2018; Anari et al. 2018) (such as a factor $e^{1/e}$ for capped additive-separable concave (Chaudhury et al. 2018) valuations), (rather high) constants for the approximation of NSW with Rado valuations (Garg, Husic, and Végh 2021) and also general non-negative, monotone submodular valuations (Li and Vondrák 2021) have been obtained.

Interestingly, for *dichotomous* submodular valuations where the marginal valuation of every agent for every good j has only one possible non-negative value (i.e., $v_i(S \cup \{j\}) - v(S) \in \{0, p\}$ for $p \in \mathbb{N}$), an allocation maximizing the NSW can be computed in polynomial time (Babaioff, Ezra, and Feige 2021). In particular, in this case one can find in polynomial time an allocation that is Lorenz dominating, and simultaneously minimizes the lexicographic vector of valuations, and maximizes both utilitarian and Nash social welfare. Moreover, this allocation also has favorable incentive properties in terms of misreporting of agents.

More generally, approximation algorithms for maximizing NSW with subadditive valuations (Barman et al. 2020; Chaudhury, Garg, and Mehta 2021) and asymmetric agents (Garg, Kulkarni, and Kulkarni 2020) have been obtained, albeit so far not with constant approximation ratios.

Preliminaries

An instance $\mathcal{I}$ is given by the triple $(\mathcal{N}, \mathcal{G}, \{v_i\}_{i \in \mathcal{N}})$ where $\mathcal{N}$ is a set of n agents and $\mathcal{G}$ is a set of m indivisible goods. Every agent $i \in \mathcal{N}$ has an additive valuation function with $v_i(A) = \sum_{j \in A} v_{ij}$ for every $A \subseteq \mathcal{G}$. Here v_{ij} represents the value i assigns to the good $j \in \mathcal{G}$. We assume that all $v_{ij} \geq 0$. In this paper, we study *2-value* additive valuations, in which $v_{ij} \in \{p, q\}$ for $p, q \in \mathbb{N}_0$. To avoid trivialities, we assume $0 < p < q$. Note that for $p = 0$ we recover the dichotomous case studied in (Barman, Krishnamurthy, and Vaish 2018b; Babaioff, Ezra, and Feige 2021). We scale down the valuation of every agent by q such that $v_{ij} \in \{v, 1\}$ where $0 < v = p/q < 1$. Moreover, throughout the paper we assume p and q are coprime.

An allocation $A = (A_1, \ldots, A_n)$ is a partition of $\mathcal{G}$ among the agents, where $A_i \cap A_j = \emptyset$, for each $i \neq j$, and $\bigcup_{i \in \mathcal{N}} A_i = \mathcal{G}$. We evaluate an allocation using the Nash social welfare $\text{NSW}(A) = \left(\prod_{i \in N} v_i(A_i)\right)^{\frac{1}{n}}$.

Notice that there might exist allocations with NSW $= 0$, however, not all such allocations have to be considered equal. In particular, among these allocations, the ones maximizing the number of agents with non-empty bundles are more preferable. Hence, for same number agents with non-empty bundles, the higher is NSW restricted on these agents the better is the allocation. In this scenario, to do not overload our notation, while comparing two allocations with 0 NSW, we say one has better NSW than the other if satisfies the aforementioned conditions. Further, when $m < n$, it is always possible to compute an optimal allocation in this sense; hence, in the rest of this paper, we assume $m \geq n$.

We represent every valuation by distinguishing between *big* and *small* goods for agents. We use sets $B_i = \{j \mid v_{ij} = 1\}$ and $S_i = \mathcal{G} \setminus B_i$ to denote the subsets of goods that agent i considers as big and small, respectively. If not differently specified, whenever we say agent i has a small (resp. big) good it means that it is a small (resp. big) for i. Globally, we use $B = \bigcup_i B_i$ and $S = \bigcap_i S_i = \mathcal{G} \setminus B$ for the sets of goods that are big for at least one agent or small for all agents, respectively. As such, an instance $\mathcal{I}$ with 2-value additive valuations can be fully described by $(\mathcal{N}, \mathcal{G}, (B_i)_{i \in \mathcal{N}}, v)$.

Of particular interest will be *non-wasteful* allocations (c.f. (Babaioff, Ezra, and Feige 2021)), in which we only assign the goods from B and give them to agents that value them as big goods. Formally, a non-wasteful allocation $A^b = (A^b_1, \ldots, A^b_n)$ has $A^b_i \subseteq B_i$ and $\bigcup A^b_i = B$.

Comparing Optimal Allocations. In our analysis, we often compare optimal allocations for 2-value valuations to optimal allocations for the same 2-value valuations with v replaced by 0. Given a fair allocation instance $\mathcal{I}$, we denote by O^* an optimal allocation and $\text{NSW}(O^*)$ the NSW of an optimal allocation. Similarly, for any $\mathcal{I}$ we consider a corresponding *dichotomous* instance $\mathcal{I}^{(d)} = (\mathcal{N}, B, \{v_i^{(d)}\}_{i\in\mathcal{N}})$ obtained by setting $v_{ij}^{(d)} = 0$ for all $i \in \mathcal{N}, j \in S_i$. We use O to denote an optimal allocation in the dichotomous instance $\mathcal{I}^{(d)}$. In particular, if we cannot assign a big good to every agent in $\mathcal{I}^{(d)}$, we assume O assigns a big good to as many agents as possible, and it maximizes the NSW among this set of agents. Note that we assume the goods in S that are small for all agents are never assigned in O, and as such we exclude them from consideration in $\mathcal{I}^{(d)}$. Clearly, O will be a non-wasteful allocation. In O^*, by Pareto-optimality, each good must be assigned to an agent. However, for any agent i, the set of big goods in O_i might not be a superset of the set of big goods in O_i^*.

We denote by $b_i = |B_i \cap O_i|$ and by $b_i^* = |B_i \cap O_i^*|$ the number of big goods agent i is receiving in O and O^*, respectively. Also, b and b^* are used to represent the vectors of b_i and b_i^*, respectively.

Example 1. Let $\mathcal{I}$ be a fair allocation instance with $n = 2$ agents, $m = 5$ goods, and $v = 2/3$. All agents have identical valuations. There are two big goods and three small goods. Then only two optimal allocations exist (with $\text{NSW} = 2$) obtained by assigning all big goods to one agent and all small goods to the other. However, for $v = 0$, every optimal allocation assigns each agent one big good.

In general, there is no simple direct connection between O and O^*, not even between vectors b and b^*. Nonetheless, it is possible to choose O^* in such a way it is, to some extent, the "closest" to a given O. Hence, in order to simplify our proofs, we will assume that $\mathcal{N}$ is numbered in non-increasing order of b_i's and subject to that in non-increasing order of b_i^*'s, i.e., for $i, j \in \mathcal{N}$, if $b_i < b_j$, or $b_i = b_j$ and $b_i^* < b_j^*$, then $j < i$. There can be many optimal solutions O^*. For a rigorous reasoning we pick O^* based on a hierarchy of three criteria based on a fixed O: (1) O^* maximizes the NSW (i.e., it is *optimal*); among all these solutions it (2) maximizes the overlap in big goods $\sum_{i\in\mathcal{N}}|O_i \cap O_i^*|$ (i.e., it is *sum-closest* to O); among all these solutions it (3) maximizes lexicographically $(|O_i \cap O_i^*|)_{i\in\mathcal{N}}$ (i.e., it is *sum-lex-closest* to O). Condition (3) is tied to the ordering of the agents, for which the tie-breaking in turn depends on O^*. Tie-breaking and lexicographic maximization allow a consistent choice of O^*, since both aim to maximize the number of big goods in O^* for agents with small index.

Given this choice of O^*, we capture the relation to O in a structured way using the notion of a *transformation graph*.

Let A and A' be two possible allocations. We denote by $G_{A\to A'}$ the *transformation graph* from allocation A to allocation A'. More formally, $G_{A\to A'} = (\mathcal{N}, E_{A\to A'})$ is a directed multigraph, where $\mathcal{N}$ is the set of the vertices. Each edge $e = (i, j) \in E_{A\to A'}$ corresponds to some good $k \in A_i \cap A'_j$ and vice versa. Observe that a transformation graph is well defined when the allocations A and A' are partial allocations, i.e., not all the goods have been allocated. We use the notation $g(e) = k$. Observe that $G_{A'\to A}$ can be obtained by simply reversing all the directed edges in $G_{A\to A'}$. A path in $G_{A\to A'}$ can be seen as a sequence of goods $(g(e_1), g(e_2), \ldots, g(e_{k-1}))$ such that $e_j = (i_j, i_{j+1})$ and $g(e_j) \in A_{i_j} \cap A'_{i_{j+1}}$ for all $j = 1, \ldots, k-1$. We say we *trade (goods along) a path* if we remove $g(e_j)$ from A_{i_j} and add it in $A_{i_{j+1}}$, for each $j = 1, \ldots, k-1$. Moreover, we say that a path is a *balancing path* if after trade the utilities of the interior agents remain unchanged, i.e., $v_{i_j}(g(e_j)) = v_{i_j}(g(e_{j-1}))$, for each $j = 2, \ldots, k-1$. Observe that every edge in the transformation graph is a balancing path; moreover, every path contained in a balancing path is a balancing path as well. In general, there exist four types of balancing paths. A *small-to-big* or *SB-balancing path* is a balancing path $(g(e_1), g(e_2), \ldots, g(e_{k-1}))$, where $g(e_1) \in S_{i_1}$ and $g(e_{k-1}) \in B_{i_k}$. BS/SS/BB-balancing paths are defined accordingly. Finally, we will briefly pay attention to BB-balancing paths starting and ending at the same agent – we term them *balancing cycles* (and omit the prefix BB, since clear from context).

Preliminaries on O, O^*, and $G_{O^*\to O}$. Given the pair of allocations O and O^* with vectors b and b^* for the numbers of big goods, the next lemmas reveal some interesting structure of $G_{O^*\to O}$. Notice that by the properties of O and O^* described above, the graph $G_{O^*\to O}$ neither has SS- nor BS-balancing paths. Moreover, it has no balancing cycles, since O^* is optimal and sum-closest to O. We are particularly interested in all agents, for which the number of big goods assigned in O and O^* differ. These agents are inherently connected to each other in the transformation graph.

Lemma 1. *For every agent i with $b_i^* > b_i$ there is an agent j with $b_j^* < b_j$ such that in $G_{O^*\to O}$ there is a BB-balancing path from i to j.*

Lemma 2. *For every agent j with $b_j^* < b_j$ there is an agent i*

1. *such that in $G_{O^*\to O}$ there is an SB-balancing path from i to j, or*
2. *with $b_i^* > b_i$ such that in $G_{O^*\to O}$ there is a BB-balancing path from i to j.*

An Optimal Algorithm when p Divides q

Consider algorithm TWOVALUEAPPROX. In phase 1, it computes O, an optimal allocation in the corresponding dichotomous instance $\mathcal{I}^{(d)}$. This can be done in polynomial time (Barman, Krishnamurthy, and Vaish 2018b). Note that after phase 1, there can be agents with empty bundles. Then we assume O maximizes the number of agents receiving at least one good. Moreover, restricting attention to the set of agents with nonempty bundles, O maximizes the NSW among them. It is easy to see that an allocation O with this property is computed both by the algorithm for dichotomous additive instances in (Barman, Krishnamurthy, and Vaish 2018b) and its generalization to dichotomous submodular ones in (Babaioff, Ezra, and Feige 2021).

For phases 2 and 3, the algorithm calls procedure BALANCE. In phase 2, if there exist unassigned goods (i.e., goods

Algorithm 1: Algorithm TWOVALUEAPPROX

Input: A fair allocation instance $\mathcal{I} = (\mathcal{N}, \mathcal{G}, (B_i)_{i\in\mathcal{N}}, v)$
Output: An allocation $A = (A_1, \ldots, A_n)$
/* Phase 1: Find optimal allocation for dichotomous instance */
Compute an optimal allocation $O = (O_1, \ldots, O_n)$ for $\mathcal{I}^{(d)}$ for goods in B
/* Phases 2 and 3 */
$A \leftarrow$ BALANCE$(\mathcal{I}, O)$
return A

Algorithm 2: Algorithm BALANCE

Input: A fair allocation instance $\mathcal{I} = (\mathcal{N}, \mathcal{G}, (B_i)_{i\in\mathcal{N}}, v)$ and a non-wasteful allocation $A^b = (A^b_1, \ldots, A^b_n)$ (i.e. with $A^b_i \subseteq B_i$ and $\bigcup A^b_i = B$)
Output: An allocation $A = (A_1, \ldots, A_n)$ of all goods in $\mathcal{G}$
/* Phase 2: Adding only-small valued goods */
Let $A_i = A^b_i$ for all $i \in \mathcal{N}$;
while *there exists* $g \in S$ **do**
  $i = \arg\min_j v_j(A_j)$
  $A_i \leftarrow A_i \cup \{g\}$ and $S \leftarrow S \setminus \{g\}$
/* Phase 3: Local search */
$i_1 = \arg\max_j v_j(A_j)$ and $i_2 = \arg\min_j v_j(A_j)$
while *moving a good* $g \in A_{i_1}$ *to* A_{i_2} *strictly increases NSW*(A) **do**
  $A_{i_1} \leftarrow A_{i_1} \setminus \{g\}$ and $A_{i_2} \leftarrow A_{i_2} \cup \{g\}$
  $i_1 = \arg\max_j v_j(A_j)$ and $i_2 = \arg\min_j v_j(A_j)$
return A

that are small for all agents), they get assigned sequentially to an agent with the currently smallest valuation.

Finally, in phase 3, big goods received by the agents may be reallocated and turned into small ones. In particular, we greedily move a big good from the agent with the highest valuation to an agent with the smallest valuation if and only if this move increases the NSW. Observe that if the current allocation has NSW $= 0$, moving one good to an empty bundle is considered profitable for the NSW as long as the number of agents with non-empty bundle increases.

Running Time. To bound the running time, we start by proving a lemma about properties of phases 2 and 3 of the algorithm. We denote by i^t_1 and i^t_2 the agents i_1 and i_2 in round t of phase 3.

Lemma 3. *The following properties hold during the execution of* BALANCE$(\mathcal{I}, O)$*:*

- *Every agent i with small goods has a valuation of at most $v_i(A_i) \le \min_{j\in\mathcal{N}} v_j(A_j) + v$.*
- *If a move in round t of phase 3 strictly increases the NSW, then (1) i^t_1 only has big goods, (2) we never moved a good away from agent i^t_2 during earlier rounds $1, \ldots, t-1$ of phase 3, and (3) none of the goods $g \in A_{i^t_1}$ is big for i^t_2.*

Algorithm TWOVALUEAPPROX runs in polynomial time: Phase 1 runs in polynomial time (Barman, Krishnamurthy, and Vaish 2018b), and Lemma 3 shows that BALANCE$(\mathcal{I}, O)$ (re-)allocates each good at most once.

Optimality. Let us now focus on the Nash social welfare of the final allocation. We show that the algorithm computes an optimal allocation when p divides q, i.e., when $p = 1$ and $q \in \mathbb{N}$ (after scaling valuations). In this case, an integer number of small goods are exactly as valuable as a big one. This fact will be key to show the main result in this section.

Theorem 1. *If $p = 1$ and $q \in \mathbb{N}$, then* TWOVALUEAPPROX *computes an optimal allocation in polynomial time.*

Proposition 1 is the first step toward proving the theorem. It implies that BALANCE$(\mathcal{I}, O)$ maintains an optimal assignment for a fixed number of big goods assigned to each of the agents. Towards this end, consider a *partial big-allocation* A^P such that $A^P_i \subseteq B_i$ for all $i \in \mathcal{N}$, i.e., in A^P all agents only receive big goods. Since A^P is partial, there might be unassigned goods $\mathcal{G}^U = \mathcal{G} \setminus \bigcup_i A^P_i$. Now consider a *small-extension* A of A^P obtained by assigning each good $g \in \mathcal{G}^U$ to some agent i with $g \in S_i$. Note that if a good $g \in \mathcal{G}^U$ is big for all agents, then A^P does not have any small-extension. We use the notation $s_i = |A_i \cap S_i|$.

Proposition 1. *If for all pairs i, j that satisfy $v_i(A_i) + v < v_j(A_j)$ we have $s_j = 0$, then A is a small-extension of A^P with maximum NSW.*

Proof. Assume by contradiction that A is not the best small-extension of A^P. Let A^* be a small-extension of A^P with largest NSW that is sum-lex-closest to A (i.e., maximizes $\sum_{i\in\mathcal{N}}|A_i \cap A^*_i|$). We define $s^*_i = |A^*_i \cap S_i|$. If A is not optimal, then there exists $i \in \mathcal{N}$ such that $s_i < s^*_i$. Thus, there must be an SS-balancing path in $G_{A^*\to A}$ from i to j with $s^*_j < s_j$. Observe that $s_j > 0$. Hence, there exists a way to trade along the path without changing the valuation of interior agents. Since A^* is an optimal small-extension that is sum-lex-closest to A, this must be strictly profitable, so $v_i(A^*_i)\cdot v_j(A^*_j) > (v_i(A^*_i) - v)\cdot\left(v_j(A^*_j) + v\right)$. Then, since $v > 0$, this is equivalent to $v_j(A^*_j) + v > v_i(A^*_i)$. Since $A_i \cap B_i = A^*_i \cap B_i = A^P_i$ and $A_j \cap B_j = A^*_j \cap B_j = A^P_j$, we see that $v_j(A^*_j) \le v_j(A_j) - v$ and $v_i(A^*_i) \ge v_i(A_i) + v$. Putting it all together we get $v_j(A_j) \ge v^*_j(A^*_j) + v > v_i(A^*_i) \ge v_i(A_i) + v$. However, we have $s_j > 0$, a contradiction to the assumption of A in the lemma. ☐

We observed in Lemma 3 that throughout BALANCE$(\mathcal{I}, O)$, all agents receiving small goods differ in valuation by at most v. This implies that when $v_i(A_i) + v < v_j(A_j)$ at any point during the algorithm, then $s_j = 0$, i.e., j has no small goods.

For the next proposition, we assume BALANCE$(\mathcal{I}, \cdot)$ is applied to a particular form of non-wasteful allocation, which will eventually result in an optimal allocation. Recall that numbers b_i and b^*_i refer to the number of big goods that agent i receives in allocations O and O^*, respectively, and that agents are numbered in non-increasing order of their valuation in O and then in O^*, i.e., if $i \le j$, then $(b_i > b_j)$ or ($b_i = b_j$ and $b^*_i \ge b^*_j$).

Definition 1. *An allocation $\tilde{O}$ is said to be* well-structured *if it is non-wasteful and there is some value $0 \le K \le n$ s.t.*

- $\tilde{b} = (b_1, \dots, b_K, b^*_{K+1}, \dots, b^*_n)$,
- *for each $i \le K$ either $b_i > b^*_i$, or $b_i = b^*_i$ and there is $j \le K$ with $b_j > b^*_j$ and $b^*_j < b^*_i$,*
- *for each $i \le K$ and $j > K$, $b^*_i \ge b^*_j$.*

Proposition 2. *Let $\tilde{O}$ be any well-structured allocation. Then* BALANCE$(\mathcal{I}, \tilde{O})$ *computes an optimal allocation.*

Proof. We denote by $m' = \sum_{i=1}^K b_i - \sum_{i=1}^K b^*_i$ the number of goods from B assigned as small in O^*.

We start with some structural observations. Suppose we remove an arbitrary set G of m' goods from $\bigcup_{i \le K} \tilde{O}_i$ in such a way that the numbers of the remaining big goods for agents $i \le K$ are a permutation of $(b^*_1, \dots, b^*_K)$. We then assign the goods in $S \cup G$ sequentially to an agent with the currently lowest valuation. Moreover, let us pretend for the moment that the goods in $S \cup G$ are small for all the agents, that means, we increase the valuation of the agents receiving them by v. By Proposition 1, this will lead to an optimal small-extension and, since we start from a partial allocation inducing a permutation of $(b^*_1, \dots, b^*_K)$, this must be an allocation with maximum NSW. This has several implications:

1. The goods in this process are indeed small for any agent receiving it. Otherwise, the allocation could be Pareto-improved, contradicting the optimality of O^*.
2. All small goods in O^* are allocated to agents i with $i > K$. For contradiction, suppose agent $i \le K$ receives a small good. As the small goods were allocated in turn to an agent with minimum valuation, we can assume that i has $\min_{k \le K} b^*_k$ big goods. Thus i must have given some big good away. Then exchanging this good with the small one Pareto-improves the allocation, contradicting the optimality of the allocation.

We now show that BALANCE indeed removes a set G of m' goods as described above.

If $m' = 0$, the statement is trivial and the proposition follows. We denote by $\tilde{O}^t$ the allocation after the t-th round in BALANCE (counting both phases 2 and 3) and $\tilde{b}^t$ the vector of big goods. We will show inductively that (1) in every round the number of big goods remain "above" O^*, i.e., there is a permutation σ of $\{1, \dots, K\}$ such that $b^*_{\sigma(i)} \le \tilde{b}^t_i$ for all $i \le K$; and (2) in phase 3 the agent with highest valuation is an $i \le K$. As the base case, consider $t = 0$ before the start of phase 2. Clearly, (1) and (2) hold by assumption.

Suppose both properties hold until the end of some round $t < m' + |S| - 1$. Consider round $t + 1$. By hypothesis there is a permutation σ such that $b^*_{\sigma(i)} \le \tilde{b}^t_i$ for all $i \le K$ and $\sum_{i \in \mathcal{N}} \tilde{b}^t_i > \sum_{i \in \mathcal{N}} b^*_i$. This implies that $\tilde{O}^t$ is sum-closer to $\tilde{O}$ than O^*, hence cannot be optimal. Moreover, there is $i < K$ such that $\tilde{b}^t_i > b^*_{\sigma(i)}$. If for all $j \le K$ we remove $\tilde{b}^t_j - b^*_{\sigma(j)}$ goods and assign them iteratively to the least-valuation agents $k > K$, the NSW becomes optimal and thus strictly improves. This implies that after round t there is a move improving the NSW, so BALANCE will not terminate since it would execute another round of phase 3.

Now consider i as the highest-valuation agent at the end of round t. By (2) this is an agent $i \le K$.

Suppose round $t + 1$ is in phase 2. Then σ still fits, and (1) holds after round $t + 1$. Suppose (2) does not hold, i.e., after round $t+1$ an agent $j > K$ has highest valuation. This agent must have received the small good in round $t+1$, so the valuations of all agents differ by at most v. Hence, phase 3 would not start if phase 2 ended after round $t + 1$. However, since there is at least one agent $k \le K$ with $b^*_{\sigma(k)} < \tilde{b}^t_k$, we proved above phase 3 would start after round $t + 1$, a contradiction.

Now suppose round $t + 1$ is in phase 3. If $b^*_{\sigma(i)} < \tilde{b}^t_i$, then σ still fits, so let us assume that $b^*_{\sigma(i)} = \tilde{b}^t_i$. If there is $j \le K$ such that $\tilde{b}^t_j = \tilde{b}^t_i$ and $b^*_{\sigma(j)} < \tilde{b}^t_j$, then $(i, j) \circ \sigma$ works. Let us assume that all agents with maximum valuation in $\tilde{O}^t$ have as many goods as in O^*. We have $\tilde{b}^{t+1}_i < b^*_{\sigma(i)}$ and $\sum_{j \le K} b^*_j \le \sum_{j \le K} \tilde{b}^{t+1}_j$ (because $t + 1 \le m'$), so there is $j \le K$ such that $b^*_{\sigma(j)} < \tilde{b}^{t+1}_j$. Since j cannot have maximum valuation in $\tilde{O}^t$, so $\tilde{b}^{t+1}_j \le \tilde{b}^{t+1}_i$. Consider the allocation where every $k \le K$ gives away $\max(0, \tilde{b}^{t+1}_k - b^*_{\sigma(k)})$ goods, except j that gives $\tilde{b}^{t+1}_j - b^*_{\sigma(j)} - 1$ goods. This allocation differs in valuation profile from O^* only by agents i and j (up to a permutation) and we have $b^*_{\sigma(j)} < \tilde{b}^{t+1}_j \le \tilde{b}^{t+1}_i \le b^*_{\sigma(i)}$, so this new allocation has higher NSW than O^*, a contradiction to the optimality of O^*. This proves that (1) holds after round $t + 1$.

Suppose (2) does not hold, i.e., there is an agent $j > K$ with highest valuation. This agent must have a small good, since $b^*_i \ge b^*_j$ for all $i \le K$, $j > K$. Hence, at the end of round $t + 1$, the valuations of all agents differ by at most v, and there is no improving move left for round $t + 2$. If $t + 1 < m' + |S|$ we have an agent $k \le K$ with $b^*_{\sigma(k)} < \tilde{b}^t_k$, and BALANCE will execute another round in phase 3, a contradiction.

Note that the good moved in round $t + 1$ must be given to an agent $j > K$ – even if we expanded the set of goods removed from agents $1, \dots, K$ from the ones in rounds $1, \dots, t + 1$ to a set G of goods considered above, all goods would be given only to agents $j > K$.

Finally, we consider the case $t = m' + |S| - 1$. Then after round $t + 1$, we obtain a permutation σ of $\{1, \dots, K\}$ such that $b^*_{\sigma(i)} \le \tilde{b}^{m'}_i$ for all $i \le K$. We also have $\sum_{i \le K} \tilde{b}^{m'+|S|}_i = \sum_{i \le K} b_i - m' = \sum_{i \le K} b^*_i$. Hence, $\tilde{b}^{m'+|S|}_i = b^*_i$ for all $i \le K$. Thus, the set of removed goods is a set G considered above, and as such the resulting allocation $\tilde{O}^{m'-|S|}$ is optimal. As a consequence, BALANCE stops after this iteration and returns an optimal allocation. $\square$

The proposition shows that if the allocation computed in phase 1 has suitable properties, then the allocation computed by BALANCE is an optimal one. We now further compare O and O^* to better understand why the hypothesis of Proposition 2 is not always satisfied by O and which conditions on $v = p/q$ are sufficient for it.

In O the big goods are as evenly balanced as possible. When $v \neq 0$, an optimal allocation O^* might require to make the big goods more unbalanced. In the next proposition, we examine the details of this observation. In case $\frac{1}{v} \in \mathbb{N}$, we observe that Proposition 2 holds, and thus Algorithm 1 computes an optimal allocation. Recall that we assume agents to be numbered in non-increasing order of b_i. The following proposition holds even when O^* is optimal and sum-closest to O (but not necessarily sum-lex-closest).

Proposition 3. *Suppose O^* is optimal and sum-closest to O and there is an agent i such that $b_i < b_i^*$. Consider an agent j such that $b_j^* < b_j$ and there is a BB-balancing path in $G_{O^* \to O}$ from i to j. Then $v_i(O_i^*) - 1 + v \cdot \lfloor \frac{1}{v} \rfloor < v_j(O_j^*) < v_i(O_i^*) - 1 + v \cdot \lceil \frac{1}{v} \rceil$, $b_j \leq b_i + 1$, and $b_j \leq b_i^*$.*

Proof. For $k \in \mathcal{N}$, we denote by $s_k^* = |O_k^* \cap S_k|$ the number of goods of k that are small to k.

As O^* is optimal, trading along a BB-balancing path in $G_{O^* \to O}$ from i to j cannot increase the NSW, i.e. $v_i(O_i^*) \cdot v_j(O_j^*) \geq (v_i(O_i^*) - 1) \cdot (v_j(O_j^*) + 1)$ and, hence, $v_i(O_i^*) \leq v_j(O_j^*) + 1$, leading to the optimality condition $b_i^* + vs_i^* \leq b_j^* + vs_j^* + 1$. Besides, if j has a good that is big to i, then either there is a balancing cycle, which contradicts the fact that O^* is closest to O, or the good is small for j and trading along the cycle gives a new allocation that Pareto-dominates O^*. So none of the goods of j is considered big by i.

We first show that $b_j \leq b_i + 1$. Suppose for contradiction that this is not the case. Then by reversing the path between i and j and trading goods, we see that O is not optimal in the dichotomous instance.

Next we show $b_i^* - b_j^* \geq 2$ and $b_j \leq b_i^*$. If $b_j \leq b_i$, then $b_j^* < b_j \leq b_i < b_i^*$ and since these numbers are integers we obtain $b_i^* - b_j^* \geq 2$, as well as $b_j \leq b_i^*$. Thus, we are left with the case $b_j = b_i + 1$. We have $b_j^* \leq b_i$ and $b_j \leq b_i^*$, and thus the following inequalities: $b_j^* \leq b_i = b_j - 1 \leq b_i^* - 1$. If one of the inequalities is strict, then we obtain $b_j^* \leq b_i^* - 2$ and $b_j \leq b_i^*$. Otherwise, $b_j^* = b_i$ and $b_j = b_i^*$. Then the optimality condition gives $s_i^* \leq s_j^*$. Now we trade along the path. Thereby we assign a big good to j. In exchange, agent i receives $s_j^* - s_i^*$ many small goods from j's bundle. This exchanges $v_i(O_i^*)$ and $v_j(O_j^*)$, and thus does not impact the NSW. This contradicts the fact that O^* is closest to O.

Having shown that $b_i^* - b_j^* \geq 2$, we see with the optimality condition that $s_j^* - s_i^* \geq \frac{1}{v}$. We prove by contradiction that the relation between $v_j(O_j^*)$ and $v_i(O_i^*)$ holds.

Assume $v_j(O_j^*) \leq v_i(O_i^*) - 1 + v\lfloor \frac{1}{v} \rfloor$. Then $v_i(O_i^*) \cdot v_j(O_j^*) \leq (v_i(O_i^*) - 1 + v\lfloor \frac{1}{v} \rfloor)(v_j(O_j^*) + 1 - v\lfloor \frac{1}{v} \rfloor)$, which means that trading along the path from i to j and transferring $\lfloor \frac{1}{v} \rfloor$ small goods from O_j^* to O_i does not decrease the NSW of the allocation. This is impossible because O^* was taken as close to O as possible.

Now, if $v_j(O_j^*) \geq v_i(O_i^*) - 1 + v \cdot \lceil \frac{1}{v} \rceil$, then $v_i(O_i^*) \cdot v_j(O_j^*) \leq (v_i(O_i^*) - 1 + v\lceil \frac{1}{v} \rceil)(v_j(O_j^*) + 1 - v\lceil \frac{1}{v} \rceil)$ and same reasoning applies by using $\lceil \frac{1}{v} \rceil$ small goods. □

We can now prove Theorem 1.

Proof of Theorem 1. We show that BALANCE$(\mathcal{I}, O)$ is an optimal allocation. To this aim we show that O satisfies the assumptions of Proposition 2.

We first observe that if $\frac{1}{v} \in \mathbb{N}$, then there exists no agent i such that $b_i < b_i^*$. Otherwise, by Lemma 1 and Proposition 3, $v_i(O_i^*) - 1 + v \cdot \lfloor \frac{1}{v} \rfloor < v_j(O_j^*) < v_i(O_i^*) - 1 + v \cdot \lceil \frac{1}{v} \rceil$ for some j. Since, $\frac{1}{v} \in \mathbb{N}$, we have $\lfloor \frac{1}{v} \rfloor = \lceil \frac{1}{v} \rceil = \frac{1}{v}$ implying $v_i(O_i^*) < v_j(O_j^*) + 1 - v \cdot \lfloor \frac{1}{v} \rfloor < v_i(O_i^*)$ which is impossible. Thus, for each $i \in \mathcal{N}$, $b_i \geq b_i^*$. Moreover, the entries of b are sorted in non-increasing order. By selecting K as the maximum index $i \in \{0, \dots, n\}$ for which $b_i > b_i^*$, we see that O is well-structured. Therefore, by Proposition 2, BALANCE$(\mathcal{I}, O)$ returns an optimal allocation. □

Approximation

In this section we study the case $\frac{1}{v} \notin \mathbb{N}$ and prove a small approximation ratio for our algorithm. The idea is to compare the behavior of BALANCE$(\mathcal{I}, O)$ to BALANCE$(\mathcal{I}, \tilde{O})$ for a suitably chosen allocation $\tilde{O}$ such that the final allocation of the latter procedure is optimal. In the following, we discuss a high-level description of the arguments.

We transform O^* into an allocation $\tilde{O}$ by moving each good from B that is assigned as small in O^* to the agent that owns it in O and removing all goods of S from the agents' bundles. The obtained allocation $\tilde{O}$ has the corresponding vector of big goods $\tilde{b}$ such that for each $i \in \mathcal{N}$ either $\tilde{b}_i = b_i$ or $\tilde{b}_i = b_i^*$. In particular, the vector $\tilde{b}$ can be written as $(b_1, \dots, b_K, b_{K+1}^*, \dots, b_n^*)$, for some index $0 \leq K \leq n$.

We set K to the largest index such that $\forall i \leq K$ we have $\tilde{b}_i = b_i > b_i^*$, or $b_i = b_i^*$ and there is $j \leq K$ such that $\tilde{b}_j = b_j > b_j^*$ and $b_j^* < b_i^*$. If there is no such index, we simply set $K = 0$. Intuitively, we choose K as the largest index such that $\tilde{O}$ qualifies as a well-structured allocation in the sense of Definition 1. Hence, by Proposition 2, we have that BALANCE$(\mathcal{I}, \tilde{O})$ returns an optimal allocation.

Suppose we run BALANCE$(\mathcal{I}, \tilde{O})$. Let $\tilde{O}^t$ denote the allocation after t time steps, and let $\tilde{T}$ be the last step before BALANCE$(\mathcal{I}, \tilde{O})$ terminates. By the choice of $\tilde{O}$, the allocation $\tilde{O}^{\tilde{T}}$ is an optimal allocation (possibly different from O^*). BALANCE moves big goods from agents $i \leq K$ and assigns them as small to agents $j > K$ as long as it is strictly profitable for the NSW. Hence, for every agent $j > K$ the number of big goods stays the same during the procedure. In $\tilde{O}^{\tilde{T}}$, every agent $j > K$ has b_j^* big goods, while, for agents $i \leq K$, the numbers of big goods can be different from b_i^*.

To show the approximation factor of our algorithm, we relate $\tilde{O}^{\tilde{T}}$ to the output of our algorithm, i.e., the output of BALANCE$(\mathcal{I}, \tilde{O})$ to the one of BALANCE$(\mathcal{I}, O)$. For this purpose, we track the allocations in BALANCE$(\mathcal{I}, \tilde{O})$ and simultaneously apply them on O. Let O^t denote the allocation after t time steps. We couple the changes to big goods in $\tilde{O}^t$ and O^t in the following way:

1. In a step t of phase 2, a globally small good from S is added to both O^t and $\tilde{O}^t$. It is given to an agent with the current smallest valuation in the respective allocation.
2. In a step t of phase 3, in which a big good is removed from the bundle of agent $i \leq K$ in $\tilde{O}^t$, we also remove one big good from i's bundle in O^t. The good is given to an agent with the current smallest valuation in the respective allocation.

Note that we couple the removal of the big good, but as small it gets assigned to potentially different agents in O^t and $\tilde{O}^t$.

Let T be the final step of BALANCE$(\mathcal{I}, O)$. Observe that in every step $t \leq T$, we can assume that the coupled process on O behaves exactly like BALANCE$(\mathcal{I}, O)$. However, it might be that $\tilde{T} \neq T$. Then, if $\tilde{T} > T$, the coupled process forces BALANCE$(\mathcal{I}, O)$ to continue turning big goods into small ones although this is not profitable for the NSW.

We also observe that, if i is the agent with current lowest valuation in $O^{\tilde{T}}$, then she will receive a small good. More generally, we show that for any $t < \tilde{T}$:

1. no agent $i \leq K$ has a minimum valuation in O^t,
2. no agent $j > K$ receives big goods,
3. no agent $i > K$ has a maximum valuation in O^t .

These properties also lead to the following result.

Lemma 4. *NSW*$(O^{\tilde{T}}) \leq$ *NSW*(O^T).

As a consequence, we can upper-bound the approximation factor of our algorithm by NSW$(\tilde{O}^{\tilde{T}})$/NSW$(O^{\tilde{T}})$.

Finally, to bound the approximation factor of Algorithm 1, we show that we can partition the agents into two groups. In one group, the agents have the same valuation in $O^{\tilde{T}}$ and $\tilde{O}^{\tilde{T}}$. In the other, the following properties are satisfied: 1) the utilitarian social welfare and, in particular, the number of goods assigned as big/small is the same 2) in $O^{\tilde{T}}$ valuations of any pair of agents differ by at most v. The properties suffice to prove the main result of this section.

Theorem 2. *Algorithm 1 has an approximation factor of at most* $\frac{24}{29} \exp\left(\frac{110}{493}\right) < 1.0345$.

Observe that Example 1 provides a lower bound to Algorithm 1 of ≈ 1.01418. It is an interesting open problem whether the approximation of Algorithm 1 can be tight.

NP-Hardness when $p \geq 3$

In this section we almost complement our positive results on polynomial-time NSW optimization. In particular, we show:

Theorem 3. *No polynomial-time algorithm computes an allocation with optimal NSW for 2-value instances, for any coprime integers $q > p \geq 3$, unless P=NP.*

We provide a reduction from *Exact-p-Dimensional-Matching (Ex-p-DM)*: Given a graph G consisting of p disjoint vertex sets $V_1, \ldots, V_p$, each of size n, and a set $E \subseteq V_1 \times \ldots \times V_p$ of m edges, it is NP-hard to decide whether there exists a p-dimensional perfect matching in G or not. Note that for $p = 3$ the problem is Ex-3-DM and thus NP-hard. NP-hardness for $p > 3$ follows by simply copying the third set of vertices in the Ex-3-DM instance $p - 3$ times, thereby also extending the edges to the new vertex sets.

Reduction: There is one good for each vertex of G, call them *vertex goods*. Additionally, there are $q(m - n)$ *dummy goods*. For each edge of G, there is one agent who values the p incident vertex goods 1 and all other goods p/q.

Lemma 5. *If G has a perfect matching, then there is an allocation $\mathcal{A}$ of goods with NSW$(\mathcal{A}) = p$.*

Proof. Suppose there exists a perfect matching in G. We allocate the goods as follows: Give each agent corresponding to a matching edge all p incident vertex goods. Now there are $m - n$ agents left. Give each of them q dummy goods. As each agent has valuation p, the NSW is p as well. □

Lemma 6. *If G has no perfect matching, then for every allocation $\mathcal{A}$ of goods, NSW$(\mathcal{A}) < p$.*

Proof. Suppose there is an allocation $\mathcal{A} = (A_1, \ldots, A_m)$ of goods with NSW$(\mathcal{A}) \geq p$. We show that in this case there must be a perfect matching in G. First, observe that if we allocate each good to an agent with maximal value for it, we obtain an upper bound on the average utilitarian social welfare of $\mathcal{A}$, i.e. $\frac{1}{m}\sum_i v_i(A_i) \leq \frac{1}{m}(pn + q(m-n) \cdot \frac{p}{q}) = p$. Applying the AM-GM inequality gives us also NSW$(\mathcal{A}) = (\prod_i v_i(A_i))^{1/m} \leq p$, and, in particular, NSW$(\mathcal{A}) = p$ iff $v_i(A_i) = p$ for all agents i. Hence each agents valuation must be p in $\mathcal{A}$ and each vertex good must be allocated to an incident agent. The next claim allows to conclude that there are only two types of agents in $\mathcal{A}$:

Claim. *If an agent i has valuation $v_i(A_i) = p$, then she either gets her p incident vertex goods or q other goods.*

We show that $(p, 0)$ and $(0, q)$ are the only integral solutions (i, j) of the equation $p = i + j\frac{p}{q}$, where $i, j \geq 0$. Clearly, every solution different from the above must satisfy $0 < i < p$. Assume for contradiction that there exists such a solution. Then it must hold $(p - i)q = jp$. Since p and q are coprime, $p - i$ must be a multiple of p and thus $i \in \{0, p\}$, a contradiction. This concludes the proof of the claim.

Let b be the number of agents receiving their p incident vertex goods in $\mathcal{A}$, and $m - b$ the number of agents receiving q other goods. Since each vertex good must be allocated to an incident agent, $bp = np$ and thus $b = n$. Hence there must be n agents receiving their p incident vertex goods, which implies that there is a perfect matching in G. □

Lemma 5 and Lemma 6 yield the proof of Theorem 3.

Using a similar reduction (with slightly different number of dummy goods), we provide a gap-preserving reduction from *Gap-4D-Matching* with almost perfect completeness to get the following APX-hardness result.

Theorem 4. *No polynomial-time algorithm approximates the maximum NSW for 2-value instances to within a factor better than* 1.000015*, unless P=NP.*

Acknowledgements

MH and GV were supported by DFG grant Ho 3831/5-1.

References

Amanatidis, G.; Birmpas, G.; Filos-Ratsikas, A.; Hollender, A.; and Voudouris, A. 2020. Maximum Nash welfare and other stories about EFX. In *Proc. 29th Int. Joint Conf. Artif. Intell. (IJCAI)*, 24–30.

Anari, N.; Gharan, S. O.; Saberi, A.; and Singh, M. 2017. Nash Social Welfare, Matrix Permanent, and Stable Polynomials. In *Proc. 8th Symp. Innov. Theoret. Comput. Sci. (ITCS)*.

Anari, N.; Mai, T.; Gharan, S. O.; and Vazirani, V. 2018. Nash Social Welfare for Indivisible Items under Separable, Piecewise-Linear Concave Utilities. In *Proc. 29th Symp. Discret. Algorithms (SODA)*, 2274–2290.

Babaioff, M.; Ezra, T.; and Feige, U. 2021. Fair and Truthful Mechanisms for Dichotomous Valuations. In *Proc. 35th Conf. Artif. Intell. (AAAI)*, 5119–5126.

Barman, S.; Bhaskar, U.; Krishna, A.; and Sundaram, R. 2020. Tight Approximation Algorithms for p-Mean Welfare Under Subadditive Valuations. In *Proc. 28th European Symp. Algorithms (ESA)*, 11:1–11:17.

Barman, S.; Krishnamurthy, S. K.; and Vaish, R. 2018a. Finding Fair and Efficient Allocations. In *Proc. 19th Conf. Econ. Comput. (EC)*, 557–574.

Barman, S.; Krishnamurthy, S. K.; and Vaish, R. 2018b. Greedy Algorithms for Maximizing Nash Social Welfare. In *Proc. 17th Conf. Auton. Agents and Multi-Agent Syst. (AAMAS)*, 7–13.

Caragiannis, I.; Kurokawa, D.; Moulin, H.; Procaccia, A.; Shah, N.; and Wang, J. 2016. The unreasonable fairness of maximum Nash welfare. In *Proc. 17th Conf. Econ. Comput. (EC)*, 305–322.

Chaudhury, B.; Cheung, Y. K.; Garg, J.; Garg, N.; Hoefer, M.; and Mehlhorn, K. 2018. On Fair Division for Indivisible Items. In *Proc. 38th Conf. Found. Software Tech. Theor. Comput. Sci. (FSTTCS)*, 25:1–25:17.

Chaudhury, B. R.; Garg, J.; and Mehta, R. 2021. Fair and Efficient Allocations under Subadditive Valuations. In *Proc. 35th Conf. Artif. Intell. (AAAI)*, 5269–5276.

Cole, R.; Devanur, N.; Gkatzelis, V.; Jain, K.; Mai, T.; Vazirani, V.; and Yazdanbod, S. 2017. Convex Program Duality, Fisher Markets, and Nash Social Welfare. In *Proc. 18th Conf. Econ. Comput. (EC)*, 459–460.

Cole, R.; and Gkatzelis, V. 2018. Approximating the Nash Social Welfare with Indivisible Items. *SIAM J. Comput.*, 47(3): 1211–1236.

Garg, J.; Hoefer, M.; and Mehlhorn, K. 2018. Approximating the Nash Social Welfare with Budget-Additive Valuations. In *Proc. 29th Symp. Discret. Algorithms (SODA)*, 2326–2340.

Garg, J.; Husic, E.; and Végh, L. 2021. Approximating Nash social welfare under Rado valuations. In *Proc. 53rd Symp. Theory Comput. (STOC)*, 1412–1425.

Garg, J.; Kulkarni, P.; and Kulkarni, R. 2020. Approximating Nash Social Welfare under Submodular Valuations through (Un)Matchings. In *Proc. 31st Symp. Discret. Algorithms (SODA)*, 2673–2687.

Garg, J.; and Murhekar, A. 2021. Computing Fair and Efficient Allocations with Few Utility Values. In *Proc. 14th Symp. Algorithmic Game Theory (SAGT)*.

Li, W.; and Vondrák, J. 2021. A constant-factor approximation algorithm for Nash Social Welfare with submodular valuations. In *Proc. 62nd Symp. Found. Comput. Sci. (FOCS)*. To appear.

Lipton, R.; Markakis, E.; Mossel, E.; and Saberi, A. 2004. On approximately fair allocations of indivisible goods. In *Proc. 5th Conf. Electr. Commerce (EC)*, 125–131.

Nguyen, T. T.; and Rothe, J. 2014. Minimizing envy and maximizing average Nash social welfare in the allocation of indivisible goods. *Discret. Appl. Math.*, 179(31): 54–68.