Journal of Artificial Intelligence Research 74 (2022) 111-142 Submitted 04/2021; published 05/2022

Fair Division of Indivisible Goods for a Class of Concave Valuations

Bhaskar Ray Chaudhury BRAYCHA@ILLINOIS.EDU
Univ. of Illinois at Urbana-Champaign, USA

Yun Kuen Cheung YUNKUEN.CHEUNG@RHUL.AC.UK
Royal Holloway University of London, United Kingdom

Jugal Garg JUGAL@ILLINOIS.EDU
Univ. of Illinois at Urbana-Champaign, USA

Naveen Garg NAVEEN@CSE.IITD.AC.IN
IIT Delhi, India

Martin Hoefer MHOEFER@EM.UNI-FRANKFURT.DE
Goethe-Universität Frankfurt am Main, Germany

Kurt Mehlhorn MEHLHORN@MPI-INF.MPG.DE
MPI for Informatics, Saarland Informatics Campus, Germany

Abstract

We study the fair and efficient allocation of a set of indivisible goods among agents, where each good has several copies, and each agent has an additively separable concave valuation function with a threshold. These valuations capture the property of *diminishing marginal returns*, and they are more general than the well-studied case of additive valuations. We present a polynomial-time algorithm that approximates the optimal Nash social welfare (NSW) up to a factor of $e^{1/e} \approx 1.445$. This matches with the state-of-the-art approximation factor for additive valuations. The computed allocation also satisfies the popular fairness guarantee of envy-freeness up to one good (EF1) up to a factor of $2 + \varepsilon$. For instances without thresholds, it is also approximately Pareto-optimal. For instances satisfying a large market property, we show an improved approximation factor. Lastly, we show that the upper bounds on the optimal NSW introduced in Cole and Gkatzelis (2018) and Barman et al. (2018) have the same value.

1. Introduction

Fair division of (scarce) resources is a fundamental problem in various multi-agent settings where the goal is to distribute resources among agents in a way that is "fair" (no agent is significantly unhappy with her allocation) and "efficient" (there is no other "fair" allocation that achieves a better total happiness). Mentions of such problems date back to the Bible (land division between Abraham and Lotte) and ancient Greek mythology (Hesios's Theogeny[1], dating back 2800 years). Even today, the problems in this field are directly motivated from several real-life scenarios like dividing inheritance property, splitting rent among tenants, splitting taxi fare among passengers, dividing household tasks/chores among

1. Prometheus and Zeus argued about how to share an ox, and agreed finally on the protocol that Prometheus cut the ox in two pieces, then Zeus chooses his piece.

all the tenants and so on; see Spliddit[2] and Fairoutcomes[3] for applications of fair division protocols in real-life scenarios. With the advent of the Internet, the development of online platforms and substantial growth in computational power, there has been a significant interest in economics and computer science to design computationally tractable protocols for fair allocation of resources.

In a standard problem instance of (discrete) fair division, there are a set N of agents and a set M of goods. Each agent $i \in N$ has a valuation function $u_i \colon 2^M \to \mathbb{R}_{\geq 0}$, representing agent i's utility for each subset of goods. The goal is to distribute the goods in a fair and efficient manner. Several problem variants have been studied, depending on the type of goods (divisible or indivisible), the type of valuation functions, the fairness criteria, and the measures of efficiency. A detailed discussion on all of them is beyond the scope of this paper. We refer readers to Moulin (2019) for a summary of important results on fair division.

In this paper, we focus on *indivisible* goods, while each good can have several copies. The agents have *capped additively separable concave* (CASC) valuation functions, which is a generalization of the valuation functions studied in Anari et al. (2018) and Garg et al. (2018).[4] Our measure of efficiency is *Nash social welfare*, which is the geometric mean of the utilities attained by all agents. The fairness notion under consideration is *Envy-Freeness up to one good* (EF1), a well-known relaxation of envy-freeness in the settings with divisible goods. We now elaborate each of the above notions.

Indivisible vs. Divisible Goods. There has been substantial progress in the direction of fairly allocating divisible goods. We briefly highlight some important results and concepts in this setting. Competitive equilibrium with equal incomes (CEEI) has emerged as the best mechanism of allocating goods. In this mechanism, a virtual market is created comprising of the same set of agents and goods, while each agent has the same purchasing power (say \$1). The goal is to determine *market clearing prices* for the goods such that *demand equals supply*, i.e., a set of prices for the goods and an allocation where each agent is allocated the bundle that maximizes her utility under the spending constraint of \$1, and all the goods are completely allocated. This allocation is envy-free (no agent strictly prefers the bundle of any other agent to her own) and Pareto-optimal (there is no allocation that increases the utility of any agent without decreasing the utility of some other agent). The early existence proofs of such prices and allocations involved a fixed point formulation. Eisenberg and Gale (1959) showed that when agents have linear valuation functions, then CEEI can be determined by a convex program that maximizes the geometric mean of the valuations, also known as the Nash social welfare. There are several algorithms that solve this convex program in polynomial time (Devanur et al., 2008, Jain and Vazirani, 2010, Orlin, 2010, Végh, 2012).

Although practically relevant, the systematic study of fair division of indivisible goods is more recent, probably because the indivisible setting poses notable challenges. Classical fairness notions like envy-freeness cannot always be guaranteed. Furthermore, maximizing efficiency objectives like the Nash social welfare is APX-hard (Lee, 2017). Despite

2. `http://www.spliddit.org/`
3. `https://fairproposals.com/`
4. Anari et al. (2018) named the valuation functions "separable piecewise linear concave" (SPLC), a name borrowed from the setting with divisible goods. However, "piecewise linear" is clearly associated to the divisible setting. This is why we rename them for indivisible setting as "additively separable concave".

these challenges, algorithmic ideas for the divisible setting have proved useful in designing polynomial-time algorithms for the indivisible setting, especially when we want to compute fair and efficient allocations for agents with additive valuation functions. In this paper, we extend these results to a setting where agents have much more general valuation functions.

Capped Additively Separable Concave (CASC) Valuation Functions. In our setting, there is a set M of m distinct goods. For each good $j \in M$, there are $k_j \geq 1$ copies; each copy is indivisible. Each agent has a valuation function which is more general than additive valuation functions. Our valuation functions primarily capture the property of *diminishing marginal utility* with every additional copy of a good. We first define additively separable concave (ASC) valuation functions. Given an agent $i \in N$ and a good $j \in M$, we define the marginal utility of the ℓ^{th} copy of good j to agent i as $u_{i,j,\ell}$. We assume that $u_{i,j,1} \geq u_{i,j,2} \geq \ldots \geq u_{i,j,k_j}$ to capture the standard economic assumption of diminishing marginal returns. The total utility an agent i derives from r copies of j is $\sum_{1\leq\ell\leq r} u_{i,j,\ell}$. The total utility an agent i derives from her bundle x_i is additively separable over all the goods:

$$u_i(x_i) = \sum_{j\in M} \sum_{1\leq\ell\leq m(j,x_i)} u_{i,j,\ell} \ ,$$

where $m(j, x_i)$ is the number of copies of good j in x_i. A capped ASC (CASC) valuation function is simply an ASC valuation function with a cap, i.e., the valuation function of agent i is of the form

$$\bar{u}_i(x_i) = \min(c_i, u_i(x_i)) \ . \tag{1}$$

We note that CASC is more general than ASC by setting $c_i \geq \sum_{j\in M} \sum_{\ell=1}^{k_j} u_{i,j,\ell}$. They are a special case of submodular functions but incomparable to gross substitutes functions, which are studied frequently especially in the special case of *capped additive functions*; see Dobzinski et al. (2021), Feldman et al. (2016), Garg et al. (2018), Roughgarden and Talgam-Cohen (2015).

Capped valuation function imposes that an agent's valuation cannot grow beyond a particular threshold, irrespective of the number of distinct goods her bundle contains. Capped valuation functions capture non-separable diminishing marginal returns, i.e., once an agent values her current bundle at a capped threshold, her utility will not increase even if she gets additional copies of the goods of which she has very few copies in her bundle.

Fairness: Envy-Freeness up to One Good (EF1). A quintessential notion of fairness is envy-freeness. An allocation $x = \langle x_1, x_2, \ldots, x_n\rangle$ is said to be envy-free if for every pair of agents i and k we have $u_i(x_i) \geq u_i(x_k)$, i.e., each agent prefers her own bundle at least as much as the bundle of any other agent. While envy-free allocations always exist when the goods are divisible, they rarely exist for indivisible goods. For illustration, consider the case with two agents and a single valuable indivisible good. Clearly, the person who does not get the good will envy the person who does.

Over the last decade, several interesting relaxations of envy-freeness have been formulated. One of the most popular ones is called *Envy-Freeness up to one good* (EF1) (Budish, 2011, Lipton et al., 2004), where no agent envies another one after the removal of some single good in the other agent's bundle. Formally, an allocation $x = \langle x_1, x_2, \ldots, x_n\rangle$ is said to be EF1 if for every pair of agents i and k, there is some $g \in x_k$ such that $u_i(x_i) \geq u_i(x_k \setminus \{g\})$.

Lipton et al. (2004) showed that EF1 allocations always exist, even when agents have much more general valuation functions than additive ones.[5]

While an EF1 allocation always exists, an EF1 allocation can be severely unsatisfactory in term of efficiency. Consider the following example with two agents and two goods. Each agent has an additive valuation, where $u_{11} = 1$, $u_{12} = 0$, $u_{21} = 0$ and $u_{22} = 1$. The allocation where agent 1 gets good 2 and agent 2 gets good 1 is clearly EF1. However, this allocation is unsatisfactory, as there is another *envy-free* allocation where both agents have higher valuations: Agent 1 gets good 1 and agent 2 gets good 2. This example demonstrates that EF1 alone may lead to unsatisfactory allocations. This brings us to the desirable property of efficiency.

Efficiency: Nash Social Welfare. Efficiency of an allocation is a measure of how much total happiness (or equivalently, welfare) the allocation achieves. A classical measure of efficiency is *social welfare*, which is the sum of utilities of the agents: $\sum_{i \in N} u_i(x_i)$. However, social welfare is not scale-invariant and therefore incompatible with the fairness notions of envy-freeness and its relaxations. For instance, consider two agents 1 and 2 and a set of goods M, and all goods are valuable to each agent. Suppose $u_1(g) \gg u_2(g)$ for all $g \in M$. Any allocation that achieves a good approximation of the optimum social welfare must allocate most of the goods to agent 1 and very few to agent 2. Clearly, this allocation is far from satisfying envy-freeness or any of its relaxations.

There are alternative measures of efficiency that are (more) compatible with envy-freeness such as Pareto-optimality. An allocation $x = \langle x_1, x_2, \ldots x_n \rangle$ is said to be Pareto-optimal if there does not exist another allocation $x' = \langle x'_1, x'_2, \ldots, x'_n \rangle$, such that $u_i(x'_i) \geq u_i(x_i)$ for all $i \in N$, with a strict inequality for at least one agent. The question then becomes how to obtain an EF1 allocation that is also Pareto-optimal. Caragiannis et al. (2016) showed that when the agents have additive valuation functions, any allocation that maximizes the *Nash social welfare* (NSW) is guaranteed to be both EF1 and Pareto-optimal, where the NSW of an allocation x is the geometric mean of the utilities attained by all agents. In our setting, when the agents have CASC valuation functions in Equation (1), the NSW of an allocation x is

$$\mathrm{NSW}(x) = \left(\prod_{i \in N} \bar{u}_i(x_i) \right)^{1/n}. \tag{2}$$

We note that the result of Caragiannis et al. (2016) does not generalize to ASC or CASC valuation functions: when agents have such valuation functions, a NSW-optimal allocation may not be EF1, although it is still Pareto-optimal.

The above discovery naturally motivates the problem of computing NSW-optimal allocations. Lee (2017) showed that this problem is APX-hard for additive valuation functions. Hence, attention has shifted to designing polynomial-time algorithms that compute approximately NSW-optimal allocations for additive and more general valuation functions; see Anari et al. (2017, 2018), Barman et al. (2018, 2020), Chaudhury et al. (2021), Cole and Gkatzelis (2018), Cole et al. (2017), Garg et al. (2020, 2021), Li and Vondrák (2021), McGlaughlin and Garg (2020).

5. In fact, this holds for any *weakly monotone* valuation functions, i.e., $u_i(x_i \cup \{g\}) \geq u_i(x_i)$ for all $x_i \subseteq M$ and $g \in M$.

In this work, we present a polynomial-time algorithm that computes an approximately NSW-optimal allocation, while the allocation also approximately satisfies Pareto-optimality and EF1, for instances where the agents have CASC valuation functions. We proceed to state our main contributions.

1.1 Our Contribution

Our main contribution is a polynomial-time algorithm to compute an allocation x that is $e^{1/e} \approx 1.445$-approximation of the optimal NSW when the agents have CASC valuation functions. The most relevant prior works for our setting are Anari et al. (2018), Barman et al. (2018), Garg et al. (2018). We improve upon them in terms of the family of valuation functions and/or the approximation guarantee. Anari et al. (2018) presented a polynomial-time algorithm that achieves a 2.718-approximation when the agents have ASC valuation functions. Garg et al. (2018) presented a polynomial-time algorithm that achieves a 2.414-approximation when the agents have capped additive valuation functions. Barman et al. (2018) presented a polynomial-time algorithm that achieves a 1.445-approximation when the agents have additive valuation functions; moreover, the allocation computed by their algorithm is EF1 and Pareto-optimal.

The allocation x our algorithm computes is also *approximately* EF1, i.e., the allocation is $1/(2+\gamma)$-EF1 such that for any two agents i and k, $\bar{u}_i(x_i) \geq 1/(2+\gamma) \cdot \bar{u}_i(x_k \setminus \{g\})$ for some $g \in x_k$ for any given constant $\gamma > 0$. Furthermore, x is also $(1+\gamma/4)$-approximately Pareto-optimal when the agents have uncapped valuation functions, i.e., there does not exist another allocation $x' = \langle x'_1, x'_2, \dots x'_n \rangle$ such that $u_i(x'_i) \geq (1+\gamma/4) \cdot u_i(x_i)$ for any agent i. The precise statement of our main result is as follows.

Theorem 1. *Given a set of agents with CASC valuation functions, a set of goods where each good has several copies, and a constant* $\gamma \in (0, 1]$*, there is a polynomial-time algorithm that computes an allocation* $x = \langle x_1, x_2, \dots, x_n \rangle$*, such that*

- $(1+\gamma/4) \cdot e^{e^{-1/(1+\gamma)}} \cdot \mathrm{NSW}(x) \geq \mathrm{NSW}(x^{opt})$ *where* x^{opt} *is an allocation that maximizes the Nash social welfare;*
- x *is* $1/(2+\gamma)$*-EF1; and*
- x *is* $(1+\gamma/4)$*-approximately Pareto-optimal when the agents have uncapped valuation functions.*

When $\gamma \searrow 0$, the approximation guarantee for NSW approaches $e^{1/e} \approx 1.445$, while the approximation factors for EF1 and Pareto-optimality approach $1/2$ and 1 respectively.

In addition to our main result, we present an improved approximation guarantee for the NSW under a *large-market* condition. Roughly speaking, a market is sufficiently *large* when there are a large number of copies of goods, and no single copy yields a significant utility to any agent in the market. In this case, the approximation guarantee can be improved to a ratio close to 1.

Finally, on a more technical side, we show that the upper bounds on the optimal NSW proved by Cole and Gkatzelis (2018) (CG-bound) and by Barman et al. (2018) (BKV-bound) are equal.

A preliminary version of this paper appeared in FSTTCS 2018 (Chaudhury et al., 2018).

1.2 Our Techniques

The backbone of our algorithm comes from the algorithm for additive valuation functions by Barman et al. (2018). We will refer to their work as BKV for the rest of this paper. The algorithm iteratively updates the prices and allocations of the goods, until they satisfy a property called *price envy-freeness up to one good* (p-EF1), which resembles the market clearing condition in the setting with divisible goods. To generalize this idea to CASC valuation functions, we use the notions of maximum bang per buck (MBB) ratio and total spending for such valuation functions, which were first proposed by Anari et al. (2018). We also need to make crucial modifications to the BKV algorithm; see Remark 2. Introducing caps into agents' valuations poses new challenges, since we must distinguish between capped agents (agents that attain their cap utilities) and uncapped agents in the algorithm and its analysis.

1.3 Roadmap of This Paper

In Section 2, we specify the model and formally define the problem, we present our approximation algorithm, and we discuss the key components of the algorithm. In Section 3, we present the analysis of the algorithm. In Section 4, we present several extended results about our algorithm. In Section 4.1, we show that our analysis for the NSW-approximation is tight, by presenting instance for which the algorithm outputs an allocation that is worse than 1/1.44 times the optimal. In Section 4.2, we present a certificate which convinces end-users of the algorithm the output is indeed approximating the optimal allocation. In Section 4.3, we explain why we settle with 1/2-EF1 by presenting instances for which the algorithm outputs are not EF1. In Section 4.4, we present a *large-market* condition that implies a better approximation guarantee of the optimal NSW. Finally, in Section 5, we show that the CG-bound and the BKV-bound are equal.

2. Algorithm

In this section, we present an algorithm that approximately maximizes the NSW defined in Equation (2), where $\bar{u}_i$ is the CASC valuation function of agent i defined in Equation (1). Observe that when the copies of goods are indivisible, the valuation function $\bar{u}_i$ does not change if we replace any $u_{i,j,\ell}$ by $\min\{u_{i,j,\ell}, c_i\}$. Thus, we may assume without loss of generality that $u_{i,j,\ell} \leq c_i$.

Let $1 < r \leq 5/4$. For every non-zero utility $u_{i,j,\ell}$, let $v_{i,j,\ell}$ be the next larger power of r. For zero utilities $v_{i,j,\ell}$ and $u_{i,j,\ell}$ are the same. Similarly, for the cap c_i, let d_i be the next larger power of r. Consider the rounded problem where each agent i has the rounded utilities $v_{i,j,\ell}$ and rounded cap d_i. By the lemma below, it suffices to solve the rounded problem with a good approximation guarantee.

Lemma 1. *Let x approximate the NSW for the rounded problem up to a factor of κ. Then x approximates the NSW for the original problem up to a factor κr.*

Proof. Let x^{opt} be an optimal allocation for the original problem. We write $\mathrm{NSW}(x^{opt}, u, c)$ for the Nash social welfare of the allocation x^{opt} with respect to utilities u and caps c. Define $\mathrm{NSW}(x, u, c)$, $\mathrm{NSW}(x^{opt}, v, d)$, and $\mathrm{NSW}(x, v, d)$ analogously. We need to bound

$\mathrm{NSW}(x^{opt}, u, c)/\mathrm{NSW}(x, u, c)$. Since $u \leq v$ and $c \leq d$ componentwise, $\mathrm{NSW}(x^{opt}, u, c) \leq \mathrm{NSW}(x^{opt}, v, d)$. Since x approximates the NSW for the rounded problem up to a factor κ, $\mathrm{NSW}(x^{opt}, v, d) \leq \kappa \cdot \mathrm{NSW}(x, v, d)$. Since $v \leq ru$ and $d \leq rc$ componentwise, $\mathrm{NSW}(x, v, d) \leq r \cdot \mathrm{NSW}(x, u, c)$. Thus

$$\frac{\mathrm{NSW}(x^{opt}, u, c)}{\mathrm{NSW}(x, u, c)} \;\leq\; \frac{\kappa \cdot \mathrm{NSW}(x, v, d)}{\mathrm{NSW}(x, v, d)/r} \;=\; \kappa r. \qquad \square$$

Barman et al. (2018) presented an elegant approximation algorithm for the case of a single copy per good and no utility caps. We generalize their approach. The algorithm uses a parameter $\varepsilon \in (0, 1/4]$, and $r = 1 + \varepsilon$. Due to Lemma 1, we assume *rounded utilities*, i.e., all nonzero utilities and caps are rounded up to powers of r.

The algorithm updates an integral allocation x, a price p_j for each good j, and a maximum bang per buck (MBB) ratio α_i for each agent i. It is an ascending price algorithm. Its output is an integral allocation and a price vector which satisfy 4ε-price-envy-freeness up to one good (4ε-p-EF1), a property specified in Definition 1 below. As we shall see in Section 3, this property implies that the allocation achieves the approximation guarantee promised in Theorem 1.

2.1 Three Invariants

Throughout the algorithm, we maintain three invariants. Recall that there are $k_j \geq 1$ copies of good j, and $m(j, x_i)$ is the number of copies of good j in x_i.

Invariant 1. All copies of all goods are always allocated to the agents. In other words, the allocation x always satisfy $\sum_i m(j, x_i) = k_j$ for each good j.

In the BKV algorithm when there is one copy per good, the MBB ratio $\alpha_i = u_{i,j}/p_j$ whenever (the single copy of) good j is assigned to i, and $\alpha_i \geq u_{i,\ell}/p_\ell$ for all goods ℓ, i.e., α_i is the maximum utility per unit of money that agent i can get at the given prices. In our case of multiple copies per good, we adopt a generalization described below. For the sake of presentation, we let $u_{i,j,0} = +\infty$ and $u_{i,j,k_j+1} = 0$ for each agent i and good j.

Invariant 2. The prices and MBB ratios are related through the following inequalities:

$$\frac{u_{i,j,m(j,x_i)+1}}{p_j} \;\leq\; \alpha_i \;\leq\; \frac{u_{i,j,m(j,x_i)}}{p_j}\;. \tag{3}$$

Invariant 2 implies that if $u_{i,j,\ell}/p_j > \alpha_i$, then at least ℓ copies of j are allocated to agent i. Also, if $u_{i,j,\ell}/p_j < \alpha_i$, then less than ℓ copies of j are allocated to agent i. Note that if α_i is equal to its upper bound in (3), we may take one copy of good j away from i without violating the inequality, as the upper bound becomes the new lower bound. Similarly, if α_i is equal to its lower bound in (3), we may assign an additional copy of good j to i without violating the inequality as the lower bound becomes the new upper bound.

Since (3) must hold for every good j, α_i must lie in the intersection of the intervals for the different goods j, i.e.,

$$\max_j \frac{u_{i,j,m(j,x_i)+1}}{p_j} \;\leq\; \alpha_i \;\leq\; \min_j \frac{u_{i,j,m(j,x_i)}}{p_j}\;.$$

The *value* of bundle x_i for agent i is given by

$$P_i(x_i) = \frac{u_i(x_i)}{\alpha_i} = \frac{1}{\alpha_i} \sum_j \sum_{1 \le \ell \le m(j,x_i)} u_{i,j,\ell} \ . \tag{4}$$

Remark 1. In case of one copy per good, $P_i(x_i) = u_i(x_i)/\alpha_i = \sum_{j \in x_i} p_j$ is the total price of the goods in the bundle. We reuse the letter P when there are multiple copies per good, although $P_i(x_i) = (1/\alpha_i) \cdot \sum_j \sum_{1 \le \ell \le m(j,x_i)} u_{i,j,\ell}$ is no longer the total price of the goods in the bundle. Clearly, the MBB ratio can differ for different goods and for different copies of the same good in an agent's bundle. The $P_i(x_i)$ captures the cost of x_i if we change the prices so that the MBB ratio of each copy of every good in x_i is the same as α_i, the lowest MBB ratio among all copies of all goods in x_i. Clearly, $P_i(x_i) \ge \sum_j m(j,x_i) \cdot p_j$.

The notions α_i and $P_i(x_i)$ in (3) and (4) are inspired by Anari et al. (2018). We say that α_i is equal to the upper bound for the pair (i,j) if α_i is equal to its upper bound in (3), and that α_i is equal to the lower bound for the pair (i,j) if α_i is equal to its lower bound in (3). An agent i is *capped* if $u_i(x_i) \ge c_i$ and is *uncapped* otherwise.

The entire algorithm is given in Algorithm 1. It starts with a greedy assignment. For each good j, it assigns each copy to the agent that values it most. The price of each good is set to the utility of the assignment of its last copy, and all MBB ratios α_i are set to one. This guarantees (3) for every pair (i,j). Also, due to the rounded utilities, all initial prices and MBB ratios are powers of r.

Invariant 3. All prices and MBB ratios are powers of r. Only the *final* price increase (line 24 of Algorithm 1, when $\beta_3 \le \min\{\beta_1, \beta_2, \beta_4\}$) may destroy this invariant.

2.2 Discussion and Definitions

To proceed, we need to define a few new notions. Recall that x_i is a multi-set. In the multi-set $x_i - j$, the number of copies of good j is reduced by one, i.e., $m(j, x_i - j) = m(j, x_i) - 1$.

Definition 1 (least spending uncapped agent, ε-p-EF1)**.** *An agent i is a* least spending uncapped agent *if she is uncapped and $P_i(x_i) \le P_k(x_k)$ for every other uncapped agent k. An agent i ε-p-envies agent k up to one good if $P_k(x_k - j) > (1+\varepsilon) \cdot P_i(x_i)$ for every good $j \in x_k$. An allocation is ε-p-*envy-free up to one good (ε-p-EF1) *if no uncapped agent ε-p-envies another agent up to one good, i.e., for every uncapped agent i and every other agent k, there is a good $j \in x_k$ such that $P_k(x_k - j) \le (1+\varepsilon)P_i(x_i)$.*

To compute an 4ε-p-EF1 allocation, we want to make updates that eventually eliminate any 4ε-p-envy from any uncapped agent. To this end, the algorithm is designed to do either of the following two steps repeatedly:

(i) Remove a copy of good from an agent k who is ε-p-envied by a least spending uncapped agent i, in hope of easing the envy to agent k from agent i.[6] To do this while maintaining the three invariants, it turns out that we need to re-assign a sequence of copies of goods. We call this a "swap" operation, which is implemented in lines 13–16 of Algorithm 1.

6. Note that $P_k(x_k - j) = P_k(x_k) - u_{k,j,m(j,x_k)}/\alpha_k < P_k(x_k)$.

Algorithm 1: Approximate the NSW for Capped Additively Separable Concave Valuations

Input : Fair Division Problem given by utilities $u_{i,j,\ell}$, $i \le n$, $j \le m$, $\ell \le k_j$, utility caps c_i, and approximation parameter $\varepsilon \in (0, 1/4]$. Let $r = 1+\varepsilon$. Non-zero $u_{i,j,\ell}$'s and c_i's are powers of r.

Output: Price vector p and a 4ε-p-EF1 integral allocation x.

1 $u_{i,j,\ell} \leftarrow \min(c_i, u_{i,j,\ell}), \forall i, j, \ell$
2 **for** $j \in M$ **do**
3   **for** $\ell \in [k_j]$ *in increasing order* **do**
4   assign the ℓ-th copy of j to $i_0 = \operatorname{argmax}_i u_{i,j,m(j,x_i)+1}$
5   Set $p_j \leftarrow u_{i_0,j,m(j,x_{i_0})}$, where i_0 is the agent to which the k_j-th copy of j was assigned
6 $\alpha_i \leftarrow 1, \forall i \in N$
7 **repeat**
8   **if** *allocation x is ε-p-EF1* **then**
9   **break** from the loop and terminate
10   Let i be a least spending uncapped agent
11   Perform a BFS in the tight graph starting at i
12   **if** *the BFS-search discovers an improving path starting in i, let $P = (i = a_0, g_1, a_1, \ldots, g_h, a_h)$ be a shortest such path* **then**
13   Set $\ell \leftarrow h$
14   **while** $\ell > 0$ *and* $P_{a_\ell}(x_{a_\ell} - g_\ell) > (1+\varepsilon)P_i(x_i)$ **do**
15   remove g_ℓ from x_{a_ℓ} and assign it to $a_{\ell-1}$
16             $\ell \leftarrow \ell - 1$
17   **else**
18   Let S be the set of goods and agents that can be reached from i in the tight graph
19         $\beta_1 \leftarrow \min_{k \in S;\ j \notin S} \alpha_k / (u_{k,j,m(j,x_k)+1}/p_j)$ `// add a good to` S
20         $\beta_2 \leftarrow \min_{k \notin S;\ j \in S} (u_{k,j,m(j,x_k)}/p_j)/\alpha_k$ `// add an agent to` S
21         $\beta_3 \leftarrow \frac{1}{r^2 P_i(x_i)} \max_{k \notin S} \min_{j \in x_k} P_k(x_k - j)$ `//` i `is happy`
22         $\beta_4 \leftarrow r^s$, where s is the smallest integer such that $r^{s-1} \le P_h(x_h)/P_i(x_i) < r^s$ and h is the least spending uncapped agent outside S `// new least spender`
23         $\beta \leftarrow \min(\beta_1, \beta_2, \max(1, \beta_3), \beta_4)$
24   multiply all prices of goods in S by β and divide all MBB ratios of agents in S by β
25   **if** $\beta_3 \le \min(\beta_1, \beta_2, \beta_4)$ **then**
26   **break** from the loop and terminate
27 **until** `False`

(ii) Raise the prices of some goods. This will lead to a drop of α_i and thus a raise of $P_i(x_i)$, which makes agent i less likely to ε-p-envy another agent. To do this while maintaining the three invariants, it turns out that the MBB ratios of some other agents who are not ε-p-envied by agent i will be reduced simultaneously. We call this a "price increase" operation, which is implemented in lines 18–24 of Algorithm 1.

We will show that by repeating steps (i) and (ii) polynomially many times, a 4ε-p-EF1 allocation is reached. In order to execute step (i) while maintaining Invariant 2, the following two notions are helpful.

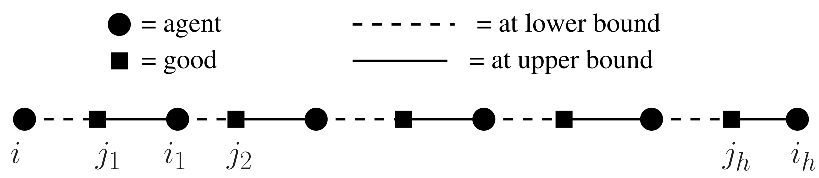

Figure 1: An improving path. Agents and goods alternate on the path and the path starts and ends with an agent. For the solid edges (j, i), α_i is at its upper bound for the pair (i, j) and for the dashed edges (i, j), α_i is at its lower bound for the pair (i, j).

Definition 2 (tight graph)**.** *A* tight graph *is a directed bipartite graph with the agents on one side and the goods on the other side. We have a directed edge (i, j) from agent i to good j if $\alpha_i = u_{i,j,m(j,x_i)+1}/p_j$, i.e., α_i is at its lower bound for the pair (i, j). We have a directed edge (j, i) from good j to agent i if $\alpha_i = u_{i,j,m(j,x_i)}/p_j$, i.e., α_i is at its upper bound for the pair (i, j). Note that necessarily $m(j, x_i) \geq 1$ in the latter case, since otherwise j does not impose an upper bound for α_i.*

Intuitively, a directed edge (i, j) from agent i to good j means we can allocate one more copy of good j to x_i without violating Invariant 2. A directed edge (j, i) from good j to agent i means we can remove one copy of good j from x_i without violating Invariant 2.

Definition 3 (improving path)**.** *An* improving path *starting at an agent i is a simple path $P = (i = a_0, g_1, a_1, \ldots, g_h, a_h)$ in the tight graph starting at i and ending at another agent a_h such that $P_{a_h}(x_{a_h} - g_h) > (1 + \varepsilon)P_i(x_i)$ and $P_{a_\ell}(x_{a_\ell} - g_\ell) \leq (1 + \varepsilon)P_i(x_i)$ for $1 \leq \ell < h$; see Figure 1 for an illustration.*

Let agent i be a least spending uncapped agent. We perform a breadth-first search (BFS) in the tight graph starting from i. If the BFS discovers an improving path starting from i, we use the shortest such path to improve the allocation. Note that if agent i ε-*p*-envies some agent that is reachable from i in the tight graph, then the BFS will discover an improving path.

In the main loop, we distinguish cases according to whether the BFS discovers an improving path starting at i or not. In the former case, we perform a swap operation; and in latter case, we perform a price increase operation. Next, we discuss the two operations.

2.3 Swap Operations

Suppose the BFS discovers the improving path $P = (i = a_0, g_1, a_1, \ldots, g_h, a_h)$. We take g_h away from a_h and assign it to a_{h-1}. If we now have $P_{a_{h-1}}(x_{a_{h-1}} + g_h - g_{h-1}) \leq (1+\varepsilon)P_i(x_i)$ we stop. Otherwise, we take g_{h-1} away from a_{h-1} and assign it to a_{h-2}. If we now have $P_{a_{h-2}}(x_{a_{h-2}} + g_{h-1} - g_{h-2}) \leq (1 + \varepsilon)P_i(x_i)$ we stop. Otherwise, we take g_{h-2} away from a_{h-2} and assign it to a_{h-3}. We continue in this way until we stop or assign g_1 to a_0. In other words, let $h' < h$ be maximum such that $P_{a_{h'}}(x_{a_{h'}} + g_{h'+1} - g_{h'}) \leq (1+\varepsilon)P_i(x_i)$. If h' exists, then we take a copy of g_ℓ away from a_ℓ and assign it to $a_{\ell-1}$ for $h' < \ell \leq h$. If h' does not exist, we do so for $1 \le \ell \le h$. Let us call the above a sequence of swaps. We summarize the above discussions in the following lemma, which follows easily from Definition 3.

Lemma 2. *Consider an execution of lines 13–16 and let h' be the final value of ℓ (this agrees with the definition of h' in the preceding paragraph). Let x' be the resulting allocation. Then $x'_\ell = x_\ell$ for $0 \le \ell < h'$, $x'_{h'} = x_{h'} + g_{h'+1}$, $x'_\ell = x_\ell + g_{\ell+1} - g_\ell$ for $h' < \ell < h$, and $x'_h = x_h - g_h$. Also,*

1. *$P_{a_h}(x_{a_h}) \ge P_{a_h}(x'_{a_h}) > (1+\varepsilon)P_i(x_i)$;*
2. *if $h' \ge 1$, then $P_{a_{h'}}(x'_{a_{h'}} - g_{h'}) = P_{a_{h'}}(x_{a_{h'}} + g_{h'+1} - g_{h'}) \le (1+\varepsilon)P_i(x_i)$;*
3. *if $h' = 0$, then $P_{a_0}(x'_{a_0} - g_1) = P_{a_0}(x_{a_0}) \le (1+\varepsilon)P_i(x_i)$;*
4. *for $h' < \ell < h$, $P_{a_\ell}(x'_{a_\ell}) = P_{a_\ell}(x_{a_\ell} + g_{\ell+1} - g_\ell) > (1+\varepsilon)P_i(x_i)$ and $P_{a_\ell}(x'_{a_\ell} - g_{\ell+1}) = P_{a_\ell}(x_{a_\ell} - g_\ell) \le (1+\varepsilon)P_i(x_i)$;*
5. *for $0 \le \ell < h'$, $P_{a_\ell}(x'_{a_\ell} - g_\ell) = P_{a_\ell}(x_{a_\ell} - g_\ell) \le (1+\varepsilon)P_i(x_i)$.*

If agent i is still the least spending uncapped agent after an execution of lines 13–16, we search for another improving path starting from i. We will show below that agent i can stay as the least spending agent for at most n^2m iterations. Intuitively, this holds because for any agent (factor n) and any fixed length shortest improving path (factor n), we can have at most m iterations for which the shortest improving path ends in this particular agent.

2.4 Price Increase Operations

We turn to the case when the BFS does not discover an improving path starting at i. This implies that agent i does not ε-p-envy any agent that she can reach in the tight graph. We then raise some prices and reduce some MBB ratios. Let S be the set of agents and goods that can be reached from i in the tight graph.

Lemma 3. *If a good j belongs to S and α_k is at its upper bound for the pair (k, j), then k belongs to S. If an agent k belongs to S and α_k is at its lower bound for the pair (k, j), then j belongs to S.*

Proof. Consider any good $j \in S$. Since j belongs to S, there is an alternating path starting in i and ending in j. If the path contains k, k belongs to S. If the path does not contain k, we can extend the path from j to k. In either case, k belongs to S.

Consider any agent $k \in S$. Since k belongs to S, there is an alternating path starting in i and ending in k. If the path contains j, j belongs to S. If the path does not contain j, we can extend the path from k to j. In either case, j belongs to S. □

We multiply all prices of goods in S and divide all MBB ratios of agents in S by a common factor $\beta > 1$. What is the effect?

- Let $u_{k,j,m(j,x_k)+1}/p_j \le \alpha_k \le u_{k,j,m(j,x_k)}/p_j$ be the inequality (3) for the pair (k, j). The endpoints do not move if $j \notin S$ and are divided by β for $j \in S$. Similarly, α_k does not move if $k \notin S$ and are divided by β if $k \in S$. So in order to preserve the inequality, we must have: If α_k is equal to the upper endpoint and p_j moves, i.e.,

$j \in S$, then α_k must also move. If α_k is equal to the lower endpoint and α_k moves then p_j must also move. Both conditions are guaranteed by Lemma 3.

- If k and j are both in S, then α_k and the endpoints of the interval for (k, j) move in sync. So agents and goods reachable from i in the tight graph stay reachable.
- If $k \notin S$, there might be a $j \in S$ such that α_k becomes equal to the right endpoint of the interval for (k, j). Then k is added to S.
- If $k \in S$, there might be a $j \notin S$ such that α_k becomes equal to the left endpoint of the interval for (k, j). Then j is added to S.
- For agents in S, $P_k(x_k)$ is multiplied by β. For agents outside S, $P_k(x_k)$ stays unchanged.

How is the common factor β chosen? There are four limiting events. Either S grows and this may happen by the addition of a good (factor β_1) or an agent (factor β_2); or $P_i(x_i)$ comes close to the largest value of $\min_{j \in x_k} P_k(x_k - j)$ for any other agent (factor β_3), or $P_i(x_i)$ becomes larger than $P_h(x_h)$ for some uncapped agent h outside S (factor β_4). Since we want prices to stay powers of r, β_4 is chosen as a power of r. The factor β_3 might be smaller than one. Since we never want to decrease prices, we take the maximum of 1 and β_3.

Lemma 4 (Invariant 3)**.** *Prices and MBB ratios are powers of r, except maybe at termination.*

Proof. This is true initially, since prices are utility values and utility values are assumed to be powers of r, and since MBB ratios are equal to one. If prices and MBB ratios are powers of r before a price update, β_1, β_2, and β_4 are powers of r. Thus prices and MBB ratios are after the price update, except maybe when the algorithm terminates. □

We next show that the algorithm terminates with an allocation that is almost price-envy-free up to one good.

Lemma 5. *Suppose $\varepsilon \leq 1/4$, then upon termination of the algorithm, x is a 4ε-p-EF1 allocation.*

Proof. If the algorithm terminates at line 9, the lemma holds trivially. Otherwise, it terminates after an execution of lines 18–24 with a final price increase, and $\beta_3 \leq \min(\beta_1, \beta_2, \beta_4)$. Suppose agent i is the least spending uncapped agent during this last execution. Let p, q denote the price vectors just before/after the final price increase respectively, and $P_k(x_k), Q_k(x_k)$ denote the values of agent k just before/after the final price increase respectively. Let h be the least spending uncapped agent after the increase; $h = i$ is possible.

We first show that $Q_i(x_i) \leq rQ_h(x_h)$. This is trivially true if $h = i$. If $h \neq i$, then we must have $h \notin S$. Since the price increase is limited by β_4, we have

$$Q_i(x_i) = \beta P_i(x_i) \leq \beta_4 P_i(x_i) = r \cdot r^{s-1} \cdot P_i(x_i) \leq rP_h(x_h) = rQ_h(x_h).$$

In either case, we have $Q_i(x_i) \leq rQ_h(x_h)$. Moreover, $Q_h(x_h) \leq Q_i(x_i)$ because h is a least spending uncapped agent after the price increase.

It suffices to show that agent h does not ε-q-envies any other agent k. If $k \in S$, then agent i does not ε-p-envies agent k. Since the final price increase changes the values of these two agents by the same factor, agent i does not ε-q-envies agent k. Thus, there exists a good $j \in x_k$ such that

$$Q_k(x_k - j) \leq (1+\varepsilon)Q_i(x_i) \leq (1+\varepsilon) \cdot rQ_h(x_h) = r^2 Q_h(x_h).$$

If $k \notin S$, then by the definition of β_3, there exists a good $j \in x_k$ such that

$$Q_k(x_k - j) = P_k(x_k - j) \leq \beta_3 r^2 P_i(x_i) = r^2 Q_i(x_i) \leq r^3 Q_h(x_h).$$

Thus we are returning an allocation that is $(r^3 - 1)$-q-EF1. We are done by noting that $r^3 = (1+\varepsilon)^3 \leq (1+4\varepsilon)$ for $\varepsilon \leq 1/4$. $\square$

Remark 2. We point out the differences between our algorithm and the BKV algorithm. Our definition of improving path is more general than theirs since it needs to take into account that the number of copies of a particular good assigned to an agent may change. For this reason, we need to maintain the MBB ratio explicitly. In the BKV algorithm, the MBB ratio of agent i is equal to the maximum utility to price ratio $\max_j u_{i,j}/p_j$ and only MBB goods can be assigned to an agent. As a consequence, if a good belongs to S, the agent owning it also belongs to S. In price changes, there is no need for the quantity β_2. In the definition of β_3, we added an additional factor r^2 in the denominator. We cannot prove polynomial running time without this factor. Finally, we start the search for an improving path from the least spending *uncapped* agent but not from the least spending agent.

3. Analysis

Let (x^{alg}, p, α) denote the allocation, the prices and the MBB ratios returned by the algorithm. Let x^{opt} denote the integral allocation that maximizes the NSW. Recall that x^{alg} is γ-p-EF1 with $\gamma = 4\varepsilon$, and inequality (3) holds for every i. We scale all the utilities of agent i and its utility cap by a factor of $1/\alpha_i$, i.e., we replace $u_{i,j,\ell}$ by $u_{i,j,\ell}/\alpha_i$ and c_i by c_i/α_i. We call the valuations before rounding and scaling the *original valuations*, the ones after rounding up to powers of r the *rounded valuations*, and the ones after rounding and scaling the *scaled valuations*. The scaling does not change the integral allocation maximizing NSW. Inequality (3) becomes

$$\frac{u_{i,j,m(j,x_i^{alg})+1}}{p_j} \;\leq\; 1 \;\leq\; \frac{u_{i,j,m(j,x_i^{alg})}}{p_j} \quad , \tag{5}$$

i.e., the goods allocated to agent i have a utility-to-price ratio of one or more, and the goods that are not allocated to agent i have a utility-to-price ratio of one or less. Also, the value of bundle x_i for agent i is now equal to its utility for agent i:

$$P_i(x_i^{alg}) = u_i(x_i^{alg}) = \sum_j \sum_{1 \leq \ell \leq m(j,x_i^{alg})} u_{i,j,\ell}. \tag{6}$$

In the lemma below, we establish that x^{opt} is an r-approximately Pareto-optimal allocation for uncapped original valuations.

Lemma 6. *Let u_i be the scaled valuations. Then we have:*

1. *x^{alg} maximizes the uncapped social welfare, i.e., $x^{alg} = \arg\max_x \sum_i u_i(x_i)$.*
2. *$\sum_i u_i(x_i^{opt}) \le \sum_i u_i(x_i^{alg})$.*
3. *x^{alg} is Pareto-optimal for uncapped, rounded valuations.*
4. *Let u^{orig} be the original valuations (unrounded and unscaled). Then for any integral allocation y of the goods, $\sum_i u_i^{\text{orig}}(y_i)/\alpha_i \le r \cdot \sum_i u_i^{\text{orig}}(x_i^{alg})/\alpha_i$.*
5. *x^{alg} is r-approximately Pareto-optimal for uncapped original valuations.*

Proof. For Part 1, let x^{SW} be the allocation that maximizes the uncapped social welfare for the scaled valuations. We can obtain x^{SW} from x^{alg} by moving copies of goods as follows. Set $x \leftarrow x^{alg}$. Consider any good j. As long as the multiplicities of j in the bundles of x and x^{SW} are not the same, identify two agents i and k, where x_i contains more copies of j than x_i^{SW} and x_k contains fewer copies of j than x_k^{SW}. Move a copy of j from i to k.

Taking a copy of good j from agent i makes the social welfare drop by at least p_j, but assigning this copy to agent k raises the social welfare by at most p_j. Thus the social welfare cannot go up by reassigning. This proves Part 1.

Parts 2 and 3 are obvious consequences of Part 1. Note that scaling does not affect Pareto-optimality, thus Part 3 also holds for the (unscaled) rounded valuations.

For Part 4, we observe that $u_i^{\text{orig}}(y_i)/\alpha_i \le u_i(y_i)$ since each original non-zero utility is rounded up to the next power of r, $\sum_i u_i^{\text{orig}}(y_i)/\alpha_i \le \sum_i u_i(y_i) \le \sum_i u_i(x_i^{alg})$ by Part 2, and $u_i(x_i^{alg}) \le r \cdot u_i^{\text{orig}}(x_i^{alg})/\alpha_i$.

Part 5 follows from combining Parts 3 and 4. $\square$

We remark that the main insights of Lemma 6 refer to scaled valuations – for example, x^{alg} does not maximize social welfare for the unscaled, rounded valuations.

Next, we focus on NSW, and u_i always refers to the scaled valuation of agent i. Let N_c and N_u be the set of capped and uncapped agents in x^{alg}, and let $c := |N_c|$, thus $n-c = |N_u|$. We number the uncapped agents such that $u_1(x_1^{alg}) \ge u_2(x_2^{alg}) \ge \ldots \ge u_{n-c}(x_{n-c}^{alg})$. Let $L := u_{n-c}(x_{n-c}^{alg})$ be the minimum utility attained by an uncapped agent. The capped agents are numbered $n-c+1$ to n. A crucial component of the analysis is to define an auxiliary problem and a further relaxed problem that facilitate bounding of NSW(x^{opt}) and NSW(x^{alg}).

3.1 Auxiliary Problem

We define an auxiliary problem with $\sum_j k_j$ distinct goods and one copy of each good. The goods are denoted by triples (i,j,ℓ), where $1 \le \ell \le m(j, x_i^{alg})$. The utility of good (i,j,ℓ) is uniform for all agents and is equal to $u_{i,j,\ell}$. Formally,

$$v_{*,(i,j,\ell)} = u_{i,j,\ell}, \tag{7}$$

where v is the utility function for the auxiliary problem. The cap of agent i is c_i. Since v is uniform over all agents, we can write $v(x_i)$ instead of $v_i(x_i)$. The capped utility of x_i

for agent i is $\bar{v}_i(x_i) = \min(c_i, v(x_i))$. Note that v is uniform, but $\bar{v}$ is not. Let x^{optaux} be a NSW-optimal allocation for the auxiliary problem.

Lemma 7. $\mathrm{NSW}(x^{opt}) = \left(\prod_i \bar{u}_i(x_i^{opt})\right)^{1/n} \le \left(\prod_i \bar{v}_i(x_i^{optaux})\right)^{1/n} = \mathrm{NSW}(x^{optaux})$.

Proof. We interpret x^{opt} as a feasible allocation $\hat{x}$ for the auxiliary problem as follows. For each good j, every agent i is allocated $m(j, x_i^{opt})$ goods with triples $(*, j, *)$ in $\hat{x}_i$, while these goods must include all goods with triples $\{(i, j, \ell) \mid 1 \le \ell \le \min(m(j, x_i^{opt}), m(j, x_i^{alg}))\}$. Due to the construction of v in (7) and inequality (5), we have $u_i(x_i^{opt}) \le v(\hat{x}_i)$ for all i, and hence $\mathrm{NSW}(x^{opt}) \le \mathrm{NSW}(\hat{x}) \le \mathrm{NSW}(x^{optaux})$. □

3.2 Relaxed Auxiliary Problem

By (6) and the definition of γ-p-EF1, for any agent i, there exists $b_i \in x_i^{alg}$ such that

$$u_i(x_i^{alg} - b_i) \le (1 + \gamma)L. \tag{8}$$

Clearly, $u_i(x_i^{alg} - b_i) = u_i(x_i^{alg}) - u_{i,b_i,m(b_i,x_i^{alg})}$. Note that there might be multiple choices of $b_i \in x_i^{alg}$ such that (8) holds. When this happens, we can use any of such choices arbitrarily for the construction below.

Let $B = \{\, (i, b_i, m(b_i, x_i^{alg}))\, ;\, 1 \le i \le n \,\}$ be a set of goods in the auxiliary problem. We now consider allocations for a relaxation of the auxiliary problem, which allows partially fractional allocation: the goods in B must be allocated integrally, but the other goods can be assigned fractionally. For convenience of notation, let g_i denote the good $(i, b_i, m(b_i, x_i^{alg}))$ in the auxiliary problem. The following lemma is crucial for the analysis.

Lemma 8. *There is a NSW-optimal allocation for the relaxed auxiliary problem, in which every good g_i is allocated to agent i.*

Proof. Assume otherwise. Among the allocations maximizing Nash social welfare for the relaxed auxiliary problem, let x^{optrel} be the one that maximizes the number of agents i that are allocated their own good g_i.

Assume first that there is an agent i who is allocated no good in B. Then g_i is allocated to some agent k different from i. Since $b_i \in x_i^{alg}$, $v(g_i) = u_{i,b_i,m(b_i,x_i^{alg})} \le c_i$; this inequality holds since utilities $u_{i,*,*}$ are capped at c_i during initialization. We move g_i from k to i and $\min(v(g_i), v(x_i^{optrel}))$ value from i to k. This is possible since only divisible goods are allocated to i. If we move $v(g_i)$ from i to k, the NSW does not change. If $v(g_i) > v(x_i^{optrel})$, then $c_i \ge v(g_i) > v(x_i^{optrel})$, and hence the product $\bar{v}_i(x_i) \cdot \bar{v}_k(x_k)$ changes from

$$\begin{aligned}
\min(c_i, v(x_i^{optrel})) \cdot \min(c_k, v(x_k^{optrel}))&\\
&= v(x_i^{optrel}) \cdot \min(c_k, v(x_k^{optrel} - g_i + g_i))\\
&= \min(c_k \cdot v(x_i^{optrel}), v(x_k^{optrel} - g_i) \cdot v(x_i^{optrel}) + v(g_i) \cdot v(x_i^{optrel}))
\end{aligned} \tag{9}$$

to

$$\begin{aligned}
\min(c_i, v(g_i)) \cdot \min(c_k, v(x_k^{optrel} - g_i) + v(x_i^{optrel}))&\\
&= v(g_i) \cdot \min(c_k, v(x_k^{optrel} - g_i) + v(x_i^{optrel}))\\
&= \min(c_k \cdot v(g_i), v(x_k^{optrel} - g_i) \cdot v(g_i) + v(x_i^{optrel}) \cdot v(g_i)).
\end{aligned} \tag{10}$$

The arguments of the min in (10) are componentwise larger than those of the min in (9). We have now modified x^{optrel} such that the NSW does not decrease, but the number of agents owning their own good increases. The above applies as long as there is an agent owning no good in B.

Next, assume every agent i owns a good in B, but not necessarily g_i. Let i be such that $v(g_i)$ is the largest among all goods g_i that are not allocated to its own agent. Then g_i is allocated to some agent k different from i. The value of the good g_ℓ allocated to i is at most $v(g_i)$ since $\ell \neq i$ and by the choice of i. We move g_i from k to i and $\min(v(g_i), v(x_i^{optrel}))$ value from i to k. This is possible since $v(g_\ell) \leq v(g_i)$ and all other goods assigned to i are divisible. Following the same argument in the last paragraph, we have now modified x^{optrel} such that the NSW does not decrease, but the number of agents owning their own good increases. We continue in this way until g_i is allocated to i for every i. □

3.3 Approximation Ratio Analysis

Let x^{optrel} be an optimal allocation for the relaxed auxiliary problem in which every good $g_i \in B$ is allocated to agent i. We will use $\mathrm{NSW}(x^{optrel})$ to upper bound $\mathrm{NSW}(x^{opt})$.

Recall that L denotes the minimum utility of an uncapped agent at allocation x^{alg}. Let $\alpha > 0$ be the real number such that

$$\alpha L = \min\{\, v(x_i^{optrel})\, ;\, v(x_i^{optrel}) < c_i \,\} \tag{11}$$

is the minimum utility of any agent that is uncapped at allocation x^{optrel}. Let $\alpha = \infty$ if every agent is capped in x^{optrel}. Let N_c^{optrel} and N_u^{optrel} be the set of capped and uncapped agents in x^{optrel}. Let h be such that $u_h(x_h^{alg}) > \alpha L \geq u_{h+1}(x_{h+1}^{alg})$.

Lemma 9. *We have the followings:*

1. *For $i \leq h$, $v(x_i^{optrel}) \leq u_i(x_i^{alg})$.*
2. *For all i, $u_i(x_i^{alg}) \leq v(x_i^{optrel}) + (1+\gamma)L$.*
3. *For $i \in N_u \cap N_c^{optrel}$, $c_i \leq \alpha L$ and $i \notin [h]$.*

Proof. We first prove Part 1. Consider any $i \leq h$. If $v(x_i^{optrel}) \leq \alpha L$, then $v(x_i^{optrel}) \leq \alpha L < u_h(x_h^{alg}) \leq u_i(x_i^{alg})$. If $v(x_i^{optrel}) > \alpha L$, then $\alpha < \infty$ and hence N_u^{optrel} is non-empty. We claim that $x_i^{optrel} = \{g_i\}$. Assume otherwise, then some divisible goods are also assigned to i. We can move some of them to an agent $k \in N_u^{optrel}$ where $v(x_k^{optrel}) = \alpha L$. This increases the NSW, a contradiction. Then the claim implies $v(x_i^{optrel}) = v(g_i) \leq u_i(x_i^{alg})$.

For Part 2, it holds since $g_i \in x_i^{optrel}$, and hence by Inequality (8), we have $u_i(x_i^{alg}) = u_i(x_i^{alg} - b_i) + u_i(b_i) \leq (1+\gamma)L + v(g_i) \leq (1+\gamma)L + v(x_i^{optrel})$.

Finally, we prove Part 3. If N_u^{optrel} is empty, then $\alpha = \infty$, so Part 3 holds trivially. Otherwise, suppose there is $i \in N_u \cap N_c^{optrel}$, such that $c_i > \alpha L$. If x^{optrel} assigns divisible goods to i, then we can move some of them to $k \in N_u^{optrel}$ where $v(x_k^{optrel}) = \alpha L$. This increases the NSW, a contradiction. Thus x_i^{optrel} consists only of g_i. But then $v(g_i) \leq u_i(x_i^{alg}) < c_i$, so i does not belong to N_c^{optrel}, a contradiction. This shows $c_i \leq \alpha L$. Then also $i \notin [h]$ for otherwise $c_i < u_i(x_i^{alg})$ and hence $i \in N_c$, a contradiction. □

In the next lemma, we apply Lemma 9 to bound $\mathrm{NSW}(x^{opt})$ from above via $\mathrm{NSW}(x^{optrel})$. To proceed, we point out a simple fact. The set of agents N can be partitioned into the following three subsets:

$$N \;=\; [h] \;\cup\; \Big(N_c \cup (N_u \cap N_c^{optrel})\Big) \;\cup\; \Big(N_u \setminus ([h] \cup N_c^{optrel})\Big). \tag{12}$$

Moreover, the size of the last subset is $|N_u| - h - |N_u \cap N_c^{optrel}| = n - c - h - |N_u \cap N_c^{optrel}|$.

Lemma 10.

$$\mathrm{NSW}(x^{opt}) \leq \mathrm{NSW}(x^{optrel}) \leq \left((\alpha L)^{n-c-h-|N_u \cap N_c^{optrel}|} \cdot \prod_{i \in N_c \cup (N_u \cap N_c^{optrel})} c_i \cdot \prod_{1 \leq i \leq h} u_i(x_i^{alg}) \right)^{\frac{1}{n}}.$$

Proof. We partition the agents in N according to (12). For the agents in the first group $i \in [h]$, we have $v(x_i^{optrel}) \leq u_i(x_i^{alg})$ from Lemma 9. This takes care of the last product term. Since $\bar{v}(x_i^{optrel}) \leq c_i$ for any i and in particular for the agents in the second group, this takes care of the middle product term.

Now we are left with the first product term for the agents in the third group. For $i \in N_u \setminus ([h] \cup N_c^{optrel})$, note that $i \in N_u^{optrel}$, hence $v(x_i^{optrel}) \geq \alpha L$ by (11). On the other hand, since i is in N_u but not in $[h]$, we have $v(g_i) \leq u_i(x_i^{alg}) \leq \alpha L$. Hence all value in $v(x_i^{optrel})$ strictly above αL would be due to fractional goods. These fractional goods can be reassigned to an agent $i' \in N_u^{optrel}$ with $v(x_{i'}^{optrel}) = \alpha L$ (such i' exists due to (11)) to strictly improve the NSW, a contradiction. Thus, we conclude that for the agents $i \in N_u \setminus ([h] \cup N_c^{optrel})$, we have $v(x_i^{optrel}) = \alpha L$. This takes care of the first product term. □

Finally, we need to bound $\mathrm{NSW}(x^{alg})$ from below. We consider allocations x for the auxiliary problem that agree with x^{alg} for the agents in $N_c \cup [h]$, and reassign the value $\sum_{i \in N_u \setminus [h]} u_i(x_i^{alg})$ fractionally. Note that for any $i \in N_u \setminus [h]$, $L \leq u_i(x_i^{alg}) \leq \min(c_i, \alpha L)$. The former inequality follows from $i \in N_u$ and the latter inequality follows from the definition of h and $i \in N_u$. We reassign value so as to move $u_i(x_i)$ towards the bounds L and $\min(c_i, \alpha L)$. As long as there are two agents whose values are not at one of their bounds, we shift value from the smaller to the larger. This decreases NSW. We end when all but one agent have an extreme allocation, either L or $\min(c_i, \alpha L)$. One agent ends up with an allocation βL with $\beta \in [1, \alpha]$.

By (12), $N_u \setminus [h]$ is the disjoint union of $N_u \cap N_c^{optrel}$ and $N_u \setminus ([h] \cup N_c^{optrel})$. By Part 3 of Lemma 9, for any agent $i \in N_u \cap N_c^{optrel}$, $c_i \leq \alpha L$. Also, since $N_u \setminus ([h] \cup N_c^{optrel})$ is a subset of N_u^{optrel}, for any agent i in this set, $\alpha L \leq c_i$ due to (11). Write $N_u \cap N_c^{optrel}$ as $S \cup T$, where the agents $i \in T$ end up at c_i and the agents in S end up at L. Also let s and t be the number of agents in $N_u \setminus ([h] \cup N_c^{optrel})$ that end up at L and αL respectively. Then

$$\mathrm{NSW}(x^{alg}) \geq \left(\prod_{i \in N_c} c_i \cdot \prod_{1 \leq i \leq h} u_i(x_i^{alg}) \cdot L^s \cdot (\alpha L)^t \cdot (\beta L) \cdot \prod_{i \in T} c_i \cdot L^{|S|} \right)^{1/n}.$$

Note that $n - c - h = s + t + 1 + |S| + |T| = s + t + 1 + |N_u \cap N_c^{optrel}|$. By Lemma 10,

$$\frac{\text{NSW}(x^{opt})}{\text{NSW}(x^{alg})} \leq \left(\alpha^s \cdot \frac{\alpha}{\beta} \cdot \prod_{i \in S} \frac{c_i}{L}\right)^{1/n} \leq \left(\left(\frac{s\alpha + \frac{\alpha}{\beta} + \sum_{i \in S} \frac{c_i}{L}}{s + 1 + |S|}\right)^{s+1+|S|}\right)^{1/n},$$

where for the second inequality we apply the arithmetic and geometric means inequality.

Following the proof of Lemma 10, the total value allocated by x^{optrel} to the agents in $N_u \setminus [h]$ is $(s+t+1)\alpha L + \sum_{i \in S \cup T} c_i$. On the other hand, the total value allocated by x^{alg} to the agents in $N_u \setminus [h]$ is $sL + t\alpha L + \beta L + \sum_{i \in T} c_i + |S|L$ on the agents in $N_u \setminus [h]$. Moreover, by Part 2 of Lemma 9, for each $i \in N_c \cup [h]$, we have $u_i(x_i^{alg}) - v(x_i^{optrel}) \leq (1+\gamma)L$. Thus,

$$\begin{aligned} 0 \;=\; \sum_{i \in N} u_i(x_i^{alg}) - v(x_i^{optrel}) \;&\leq\; (|N_c| + h)(1+\gamma)L + \left(sL + t\alpha L + \beta L + \sum_{i \in T} c_i + |S|L\right) \\ &\quad - \left((s + t + 1)\alpha L + \sum_{i \in S \cup T} c_i\right), \end{aligned}$$

and hence

$$(s + t + 1)\alpha L + \sum_{i \in S \cup T} c_i \leq (|N_c| + h)(1 + \gamma)L + sL + t\alpha L + \beta L + \sum_{i \in T} c_i + |S|L.$$

After rearranging, dividing by L and adding α/β on both sides,

$$\begin{aligned} s\alpha + \frac{\alpha}{\beta} + \sum_{i \in S} \frac{c_i}{L} &\leq (1 + \gamma)(|N_c| + h) + s + |S| + \frac{\alpha}{\beta} + \beta - \alpha \\ &\leq (1 + \gamma)(|N_c| + h) + s + |S| + 1 \leq (1 + \gamma)n. \end{aligned}$$

Note that $\beta + \alpha/\beta - \alpha \leq 1$ for $\beta \in [1, \alpha]$, since the expression is one at $\beta = 1$ and $\beta = \alpha$ and the second derivative as a function of β is positive. Thus,

$$\begin{aligned} \frac{\text{NSW}(x^{opt})}{\text{NSW}(x^{alg})} &\leq \left(\left(\frac{(1 + \gamma)(|N_c| + h) + s + |S| + 1}{s + 1 + |S|}\right)^{s+1+|S|}\right)^{1/n} \\ &\leq \left(\frac{(1 + \gamma)n}{s + 1 + |S|}\right)^{(s+1+|S|)/n} \leq e^{e^{-1/(1+\gamma)}}, \end{aligned}$$

since $((1+\gamma)\delta)^{1/\delta}$ as a function of δ attains its maximum for $\delta = \frac{1}{(1+\gamma)} e^{1/(1+\gamma)}$. The value of the maximum is $\exp(\exp(-1/(1 + \gamma)))$; when $\gamma \searrow 0$, this value tends to $e^{-1/e} \approx 1/1.44467$.

Theorem 2. *Suppose $\varepsilon \in (0, 1/4]$, $\gamma = 4\varepsilon$, x^{alg} is the allocation computed by the algorithm for the rounded valuations, and x^{opt} is an allocation maximizing Nash social welfare for the* original *valuations. Then*

$$\frac{\text{NSW}(x^{opt})}{\text{NSW}(x^{alg})} \leq (1 + \gamma/4) \cdot e^{e^{-1/(1+\gamma)}}.$$

3.4 Polynomial Running Time

Recall that n is the number of agents, there are k_j copies of good j, and $m = \sum_j k_j$ is the total number of indivisible goods (i.e., all copies of all distinct goods). We define U as the ratio of the maximum to minimum non-zero utility, i.e., $U := \max_{i,j,k} u_{i,j,k} / \min_{i,j,k:u_{i,j,k}\neq 0} u_{i,j,k}$.

Our analysis follows Barman et al. (2018) with one crucial difference. Lemma 12 is new. For its proof, we need the revised definition of β_3.

Lemma 11. *The value of the least spending uncapped agent is non-decreasing.*

Proof. This is clear for price increases. Consider a sequence of swaps along an improving path $P = (i = a_0, g_1, a_1, \ldots, g_h, a_h)$, where the agent a_h loses a good, the agents a_ℓ, $h' < \ell < h$, lose and gain a good, and the agent $a_{h'}$ gains a good. By Lemma 2, all agents a_ℓ with $h' < \ell \leq h$ have a value of at least $(1+\varepsilon)P_i(x_i)$ after the swap. Also the value of agent $a_{h'}$ does not decrease. □

Lemma 12. *For any agent k, let j_k be a highest price good in x_k. Then $\max_k P_k(x_k - j_k)$ does not increase in the course of the algorithm as long as this value is above $(1+\varepsilon)\min_{uncapped\ i} P_i(x_i)$. Once $\max_k P_k(x_k - j_k) \leq (1+\varepsilon)\min_{uncapped\ i} P_i(x_i)$, the algorithm terminates.*

Proof. We first consider price increases and then a sequence of swaps.

Consider any price increase which is not the last. Such price increase is performed in lines 18–24 of Algorithm 1. Then $\beta_3 > \min(\beta_1, \beta_2, \beta_4)$. Let i and h be the least uncapped spender before and after the price increase respectively. Let q be the price vector after the increase and $Q_i(x_i)$ be the agent i's value at q. Then $Q_h(x_h) \leq Q_i(x_i) \leq rQ_h(x_h)$.

Since there is no improving path starting from agent i, for any $k \in S$, we have $Q_k(x_k - j_k) \leq (1+\varepsilon)Q_i(x_i) \leq (1+\varepsilon)rQ_h(x_h)$. For the agent $k \notin S$ defining β_3, we have

$$Q_k(x_k - j_k) = P_k(x_k - j_k) = \beta_3(1+\varepsilon)rP_i(x_i) \geq (1+\varepsilon)rQ_i(x_i) \geq (1+\varepsilon)rQ_h(x_h);$$

we used the equality $r = 1+\varepsilon$ and the inequality $Q_i(x_i) = \beta P_i(x_i) \leq \beta_3 P_i(x_i)$ in this derivation. Hence $\max_k Q_k(x_k - j_k) = \max_k P_k(x_k - j_k)$.

Consider next a sequence of swaps. We have an improving path from i to k, say $P = (i = a_0, g_1, a_1, \ldots, g_h, a_h = k)$. Let x' be the allocation after the sequence of swaps. Then $\min_j P_k(x'_k - j) \leq \min_j P_k(x_k - j)$ since k loses a good. Also, $\min_j P_\ell(x'_\ell - j) \leq (1+\varepsilon)P_i(x_i)$ for all $\ell \in [0, h-1]$ by Lemma 2. □

Lemma 13. *The number of subsequent iterations with no change of the least spending uncapped agent and no price increase is bounded by $n^2 m$.*

Proof. Let i be the least spending agent. To have no price increase, every iteration must find an improving path starting from i. We count for any other agent k, how often the improving path can end in k. For each fixed length of the improving path, this can happen at most m times (for details see Barman et al. (2018)); the argument is similar to the argument used in the strongly polynomial algorithms for weighted matching (Edmonds and Karp, 1972). □

Lemma 14. *If the least spending uncapped agent changes after a price increase, the value of the old least spending uncapped agent increases by a factor of at least r.*

Proof. The least spending uncapped agent changes if $\beta = \beta_4$ and β_4 is at least r. So $P_i(x_i)$ increases by at least r. □

Theorem 3. *The number of iterations is bounded by $n^3 m^2 \log_r(mU)$.*

Proof. Divide the execution into two parts. In the first part, there are agents that own no good, and in the second part every agent owns at least one good and hence all the $P_i(x_i)$ are non-zero.

In any iteration of the first part $P_i(x_i) = 0$, where i is a least spending agent. A shortest improving path $P = (i = a_0, g_1, a_1, \ldots, g_h, a_h)$ starting in i visits agents a_1 to a_{h-1} owning exactly one good and ends in agent a_h owning more than one good. The sequence of swaps will take away g_h from a_h and assign g_{i+1} to a_i for $0 \leq i < h$. Since every price increase will grow S by either a good or an agent, an improving path will exist after at most $n + m$ iterations. Since there are at most n agents that own no good, there are only $O(n(n + m))$ iterations in the first part.

We come to the second part. Divide its execution into maximum subsequences with the same least spender. Consider any fixed agent i and the subsequences where i is the least spender. At the end of each subsequence, i either receives an additional good, or there is a price increase. In the latter case, $P_i(x_i)$ is multiplied by at least r.

Consider the subsequences between price increases. At the end of a subsequence i receives an additional good. It may or may not keep this good until the beginning of the next subsequence. But whenever i loses a good via swapping, the value of i remains at least r times the value of the least spender at that time. By Lemma 11, the value of i is at least r times of her value in the previous subsequence.

Together with Lemma 13, we have shown: After at most $m \cdot n^2 m$ iterations with i being the least spender, $P_i(x_i)$ is multiplied by a factor r. Thus there can be at most $n^2 m^2 \log_r(mU)$ such iterations. Multiplication by n yields the bound on the number of iterations. □

3.5 Approximate Envy-Freeness Up to One Good

The allocation computed by our algorithm maximizes NSW up to a factor of 1.445. By Lemma 5, it also gives any uncapped agent i the guarantee that $\min_{j \in x_k} P_k(x_k - j) \leq (1 + 4\varepsilon) P_i(x_i)$ for every other agent k. This guarantee is not meaningful for agent i as the left hand side is in terms of the utility for agent k. We now show that it implies $\min_{j \in x_k} u_i(x_k - j) \leq (2 + 4\varepsilon) \cdot u_i(x_i)$, i.e., the utility for i of k's bundle minus one good is essentially bounded by twice the utility of i's bundle for i. The proof considers the additional utility for i of the goods that k has in excess of i up to one good. This additional utility is bounded by $(1 + \varepsilon) u_i(x_i)$. If there is only one copy of each good, x_k and x_i are disjoint and hence any copy of a good in x_k is in excess of i's possession of the same good.

Theorem 4. *The allocation computed by the algorithm satisfies $\min_{j \in x_k} u_i(x_k - j) \leq (2 + 4\varepsilon) \cdot u_i(x_i)$ for any agent k and any uncapped agent i.*

Proof. Let g be such that $P_k(x_k - g) \le (1+4\varepsilon)P_i(x_i)$. Then

$$
\begin{aligned}
u_i(x_k - g) &\le u_i(x_i \cup x_k - g) && \text{(more never harms)}\\
&= u_i(x_i) + \sum_j \sum_{\ell = m(j,x_i)+1}^{m(j, x_k \cup x_i - g)} u_{i,j,\ell} \\
&\le u_i(x_i) + \sum_j \sum_{\ell = m(j,x_i)+1}^{m(j, x_k \cup x_i - g)} \alpha_i p_j && \text{since } u_{i,j,\ell}/p_j \le \alpha_i \text{ for } \ell > m(j,x_i)\\
&= u_i(x_i) + \sum_j \sum_{\ell=1}^{m(j,x_k - g)} \alpha_i p_j \\
&\le u_i(x_i) + \sum_j \sum_{\ell=1}^{m(j,x_k - g)} \alpha_i \frac{u_{k,j,\ell}}{\alpha_k} && \text{since } u_{k,j,\ell}/p_j \ge \alpha_k \text{ for } k \le m(j,x_k)\\
&= u_i(x_i) + \alpha_i P_k(x_k - g) && \text{definition of } P_k(x_k - g)\\
&\le u_i(x) + \alpha_i(1+4\varepsilon)P_i(x_i) && \text{since } P_k(x_k - g) \le (1+4\varepsilon)P_i(x_i)\\
&= (2+4\varepsilon)u_i(x_i) && \text{since } u_i(x_i) = \alpha_i P_i(x_i).
\end{aligned}
$$

□

For the case of only a single copy per good, $\min_{j \in x_k} u_k(x_k - j) \le (1+\varepsilon)u_i(x_i)$ was shown by Barman et al. (2018). We do not know whether the factor 2 in the theorem above is best possible. We show in Section 4.3 that a factor larger than 1.2 is necessary.

4. Extensions

In this section, we present several extended results about our algorithm.

4.1 A Lower Bound on the Approximation Ratio of the Algorithm

We show that the approximation ratio of the algorithm for NSW is no better than a ratio of 1.440. Let k, s and K be positive integers with $K \ge k$ which we fix later. Consider the following instance. We have $h = s(k-1)$ goods of value K and $n = h + s$ goods of value 1. There is one copy of each good. The number of agents is n, and all agents value the goods in the same way.

The algorithm may construct the following allocation. There are h agents that are allocated a good of value 1 and a good of value K, and there are s agents that are allocated a good of value 1. This allocation can be constructed during initialization. The prices are set to the values and the algorithm terminates.

The optimal allocation will allocate a good of value K to h players and spread the $h + s = sk$ goods of value 1 across the remaining s agents. So s agents get value k each. Thus

$$\frac{\mathrm{NSW}(x^{opt})}{\mathrm{NSW}(x^{alg})} = \left(\frac{K^h k^s}{(K+1)^h}\right)^{1/(h+s)} = \left(\left(\frac{K}{K+1}\right)^{(k-1)s} k^s\right)^{1/ks} = \left(\frac{K}{K+1}\right)^{(k-1)/k} k^{1/k}.$$

The term involving K is always less than one. It approaches 1 as K goes to infinity. The second term $k^{1/k}$ has it maximal value at $k = e$. However, we are restricted to integral values. We have $2^{1/2} = 1.41$ and $3^{1/3} = 1.442$. For $k = 3$, $(K/(K+1))^{2/3} = \exp(\frac{2}{3}\ln(1 - 1/(K+1))) \approx \exp(-\frac{2}{3(K+1)}) \approx 1 - \frac{2}{3(K+1)}$. So for $K = 666$, the factor is less than $1 - 1/1000$ and therefore $\mathrm{NSW}(x^{opt})/\mathrm{NSW}(x^{alg}) \geq 1.440$.

4.2 Certification of the Approximation Ratio

How can a user of an implementation of the algorithm be convinced that the solution returned has a NSW no less than 1/1.445 times the optimum? She may read this paper and convince herself that the program indeed implements the algorithm described in this article. This may be unsatisfactory; see McConnell et al. (2011). In this section, we describe a simple certificate.

The algorithm returns an allocation x^{alg}, prices p_j for the goods, and MBB-ratios α_i for the agents. After scaling all utilities and the utility cap of agent i by α_i, we have the inequality (5). The user needs to understand that this scaling has no effect on the optimal allocation. As in Section 3, we introduce the auxiliary problem with $m = \sum_j k_j$ goods and one copy of each good. The agents have uniform utilities. The user needs to understand that the NSW of the auxiliary problem is an upper bound (Lemma 7). We are left with the task of convincing the user of an upper bound on the NSW of the auxiliary problem. We assume that $m \geq n$, for otherwise the optimal NSW is zero.

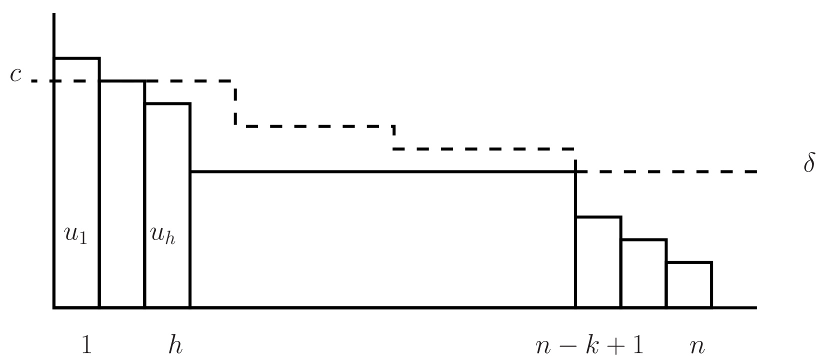

Figure 2: The allocation constructed in the proof of Theorem 5. The dashed line above agents 1 to $n-k$ indicates the utility caps. The solid rectangles visualize the values of the bundles.

Theorem 5. *Let $c_1 \geq c_2 \geq \ldots \geq c_n$ be the utility caps of the agents, let $u_1 \geq u_2 \geq \ldots \geq u_m$ be the utilities of the m goods of the auxiliary problem. Let $u_0 := +\infty$, $c_0 := +\infty$ and $c_{n+1} := 0$. Then there exist integers h, k satisfying $h, k \geq 0$ and $h + k \leq n$, such that by defining $\delta(h,k) := \left(\sum_{h+1 \leq j \leq m} u_j - \sum_{n-k+1 \leq i \leq n} c_i\right)/(n-h-k)$, we have $c_{n-k+1} \leq \delta(h,k) < c_{n-k}$*

and $\delta(h,k) < u_h$. Then for any optimal allocation x^{optaux} for the auxiliary problem,

$$\text{NSW}(x^{optaux}) \le \left(\prod_{1 \le i \le h} \min(c_i, u_i) \cdot (\delta(h,k))^{n-h-k} \cdot \prod_{n-k+1 \le i \le n} c_i \right)^{1/n}.$$

The right hand side is illustrated in Figure 2.

Proof. First we verify the existence of h, k. Find the least $k \ge 0$ such that $c_{n-k+1} \le \delta(0,k) < c_{n-k}$.[7] With this k, we can set $h = 0$ to guarantee that $\delta(h,k) < u_h = +\infty$.[8] There might be other possible values of h, k, which can be found by an exhaustive search.

We insist that the goods 1 to h are allocated integrally (when $h = 0$ this means no good is allocated integrally), while allowing the remaining goods to be allocated fractionally.

Clearly, we cannot allocate more than c_i to any agent, in particular, not to agents $n-k+1$ to n and to agents 1 to h. The optimal way to distribute value $\sum_{h+1 \le j \le m} u_j$ to agents $h+1$ to n is clearly to allocate δ each to agents $h+1$ to $n-k$ which all have a cap of more than δ and to the assign their cap to agents $n-k+1$ to n. The goods u_1 to u_h of value more than δ are best assigned to the agents with the largest utility caps. Assume that two such goods, say u_ℓ and u_k, are allocated to the same agent. Then one of the first h agents is allocated no such good; let v be the value allocated to this agent. Moving u_k to this agent and value $\min(u_k, v)$ from this agent in return, does not decrease the NSW. Also, if any fractional goods are assigned in addition to the first h agents, we move them to agents $h+1$ to $n-k$ and increase the NSW. This establishes the upper bound. □

The upper bound can be computed in time $O(n^2 + m)$. We conjecture that it can be computed in linear time $O(n + m)$. We also conjecture that the bound is never worse than the bound used in the analysis of the algorithm. It can be better as the following example shows. We have two uncapped agents and three goods of value $u_1 = 3$, $u_2 = 1$ and $u_3 = 1$, respectively. The algorithm may assign the first two goods to the first agent and the third good to the second agent. The set B in the analysis of the algorithm (Section 3) consists of the first good and the last good. Then $\ell = 1$. The optimal allocation allocates 3 to the first agent and 2 to the second agent. Thus $\alpha L = 2$. The analysis uses the upper bound $\sqrt{4 \cdot 2}$ for the NSW of the optimal allocation. The theorem above gives the upper bound $\sqrt{3 \cdot 2}$; note that $h = 1$, $k = 0$, and $\delta = 2$.

4.3 Lower Bounds on Approximate Envy-Freeness up to One Copy

For the case of additive valuation functions and one copy of each good, the optimal NSW allocation is EF1 as shown in Caragiannis et al. (2016). The allocation computed by the BKV algorithm is also EF1. In this section, we show that these properties hold neither for the multi-copy case nor for the capped case.

Let ε be a small positive real number, say $\varepsilon = 0.01$. Let $r = 1 + \varepsilon$, and let s be the smallest power of r greater or equal to $2r^2$. Then $2 < s < 2.04$. We first give an example for the multi-copy uncapped case. There are two agents and two goods. Good 1 has 5 copies,

7. It is possible that $k = 0$ when, for instance, all caps are $+\infty$. It is also possible that $k = n$ when, for instance, $\sum_{j=1}^{m} u_j \gg \sum_{i \in N} c_i$.

8. If every u_j is tiny, m is huge and n is small, h has to be zero due to the constraint $u_h > \delta(h,k)$.

and good 2 has 2 copies. For the first agent, the utility vector for good 1 is $(s, s, 0, 0, 0)$ and for good 2 is $(1, 0)$. For the second agent, the utility vector for good 1 is $(s, s, s, 0, 0)$ and for good 2 is (s, s). Then in the optimal NSW allocation x, the first agent is allocated two copies of good 1 and none of good 2, while the second agent is allocated three copies of good 1 and two copies of good 2. For this allocation, $\mathrm{NSW}(x) = (2s \cdot 5s)^{1/2} = 10^{1/2}s$. Note that allocating one copy of the second good to each agent gives a NSW of $((2s+1)\cdot 4s)^{1/2} = (8s^2 + 4s)^{1/2} < (10s^2)^{1/2}$ since $4 < 2s$. Clearly, in x the first agent envies the second agent even after removing one copy (of either good) from the allocation of the second agent, because $u_1(x_2 - g) \geq (2s+1) > 2s = u_1(x_1)$ for any choice of g.

Lemma 15. *In the case of several copies per good, the allocation maximizing* NSW *is not necessarily envy-free up to one copy.*

What does the algorithm do? The initial assignment is equal to the optimal assignment and sets $p_1 = p_2 = s$ and $\alpha_1 = \alpha_2 = 1$. Agent 1 is the least spending uncapped agent. The allocation is not ε-p-EF1, since $P_1(x_1) = 2s$ and $P_2(x_2) = 5s$ and $\min_{g \in x_2} P_2(x_2 - g) = 4s$. The constraints on α_1 are $[0, 1]$ by the first good and $[1/s, 1]$ by the second good. The tight graph consists only of agent 1. We enter the else-case of the main loop with $S = \{1\}$. Then $\beta_1 = s > 2$, $\beta_2 = \infty$, $\beta_3 = 4s/(2s \cdot r^2) = 2/r^2 < 2$ and $\beta_4 = r^{1+\lfloor \log_r 5/2 \rfloor} \geq 5/2 \geq \beta_3$. Thus $\beta = \beta_3$. We decrease $\alpha_1$to $r^2/2 \approx 1/2$ and terminate. Now $P_1(x_1) = (2/r^2)\cdot 2s = 4s/r^2$ and hence $(1+4\varepsilon)P_1(x_1) \geq 4s = P_2(x_2 - g) = 4s$. The optimal allocation is now 4ε-p-envy free up to one copy.

We turn to possible improvements of Theorem 4. Since $\min_{g \in x_2} u_1(x_2 - g) = 2s+1$ and $u_1(x_1) = 2s$, in order to have $\min_{g \in x_2} u_1(x_2 - g) \leq (c + \varepsilon)u_1(x_1)$, we need $c \geq 1 + (1 - 2\varepsilon s)/(2s) \geq 1.2$.

Lemma 16. *With $\alpha_1 = r^2/2$, $\alpha_2 = s$, $p_1 = p_2$, the optimal allocation in the example above is 4ε-p-envy free up to one copy.*

Lemma 17. *Theorem 4 does not hold when the constant 2 is replaced by 1.2.*

For the linear capped case, again we have two agents, and this time we have four goods with one copy each. The utility vectors of both agents are (s, s, s, s), but the first agent is capped at 3, while the second agent is uncapped. Then the optimal NSW allocation x allocates one good to the first agent and three goods to the second agent for $\mathrm{NSW}(x) = (s \cdot 3s)^{1/2}$. Note that allocating 2 goods to each agent yields a NSW of $(3 \cdot 2s)^{1/2} < (3s^2)^{1/2}$ since $s > 2$. In the optimal assignment, the first agent envies the second agent, even after removing one good from the allocation of the second agent.

What does the algorithm do? It may construct the optimal assignment during initialization; the prices of all four goods are set to s and both α-values are set to one. Agent 1 is the least spending uncapped agent. The tight graph consists of the edges from agent 1 to the goods owned by agent 2 and from these goods to agent 1. An improving path exists and one of these goods is reassigned to agent 1. The algorithm terminates with an allocation in which both agents own two goods.

Lemma 18. *In the case of single copies per good but with utility caps, the allocation maximizing* NSW *is not necessarily envy-free up to one good.*

4.4 Large Markets

In this section, we show that the approximation guarantee of our algorithm can be improved on *large* instances, in the sense that there are a large number of copies of goods, and no copy yields a significant utility to any agent in the market. Precisely, we call a market δ-large if $u_{i,j,\ell} \leq \delta \cdot u_i(G)/n$, where G is the multiset of all copies of items, and thus $u_i(G) = \sum_j \sum_{1 \leq \ell \leq k_j} u_{i,j,\ell}$.

For simplicity, we restrict to instances without utility caps. With utility caps, the treatment becomes more clumsy, but does not give additional insights.

Theorem 6. *Suppose $\delta < 1$. For a δ-large market in which all non-zero utilities are powers of $r = 1 + \varepsilon$,*

$$\frac{\mathrm{NSW}(x^{opt})}{\mathrm{NSW}(x^{alg})} \leq \frac{1+4\varepsilon}{1-\delta}.$$

Proof. Let (x^{alg}, p, α) be the allocation, price vector, and scaling factors returned by the algorithm. For simplicity we use $x = x^{alg}$. We scale all utilities $u_{i,*,*}$ by α_i. We have

$$u_{i,j,m(j,x_i)+1} \leq p_j \leq u_{i,j,m(j,x_i)}$$

for all i and j. Let $U = \sum_i u_i(x_i) = \sum_i \sum_j \sum_{1 \leq \ell \leq m(j,x_i)} u_{i,j,\ell}$.

Let x^{opt} be the allocation maximizing NSW. Then $\sum_i u_i(x_i^{opt}) \leq U$ by Lemma 7(a), and hence $\mathrm{NSW}(x^{opt}) \leq ((U/n)^n)^{1/n} = U/n$.

We next prove a lower bound on $\mathrm{NSW}(x)$. Note that $u_i(G) \leq U$ due to Lemma 7(a). Thus, $u_{i,j,\ell} \leq \delta U/n$.

For any i, we have $P_i(x_i) = u_i(x_i)$. Since the allocation returned by the algorithm is 4ε-p-envy-free up to one copy, we have $\min_{g \in x_k} u_k(x_k - g) \leq (1+4\varepsilon) u_i(x_i)$ for every agent k. Let g_k be the good that minimizes the left hand side. Summing over all k yields

$$\sum_k u_k(x_k) - \sum_k u_k(g_k) \leq (1+4\varepsilon)n \cdot u_i(x_i),$$

and hence

$$u_i(x_i) \geq \frac{U - n(\delta/n)U}{(1+4\varepsilon)n} = \frac{1-\delta}{1+4\varepsilon} \cdot \frac{U}{n}.$$

Thus

$$\frac{\mathrm{NSW}(x^{opt})}{\mathrm{NSW}(x^{alg})} \leq \frac{U/n}{\left(\prod_i u_i(x_i)\right)^{1/n}} \leq \frac{U/n}{\left(\left(\frac{1-\delta}{1+4\varepsilon} \cdot \frac{U}{n}\right)^n\right)^{1/n}} = \frac{1+4\varepsilon}{1-\delta} \quad . \qquad \square$$

For a δ-large market where non-zero utilities are not powers of $r = 1+\varepsilon$, we first round the non-zero utilities up to the nearest power of r. After rounding, the market is δr-large. Theorem 6 and Lemma 1 implies that the approximation guarantee is $(1+4\varepsilon)r/(1-\delta r) \leq (1+6\varepsilon)/(1-\delta-\delta\varepsilon)$ when $\varepsilon \leq 1/4$ and $\delta < 4/5$.

5. The CG-Bound and the BKV-Bound are Equal

In this section, we restrict the discussion to the case of a single copy per good and no utility caps, i.e., the standard case of additive utilities. We have n agents and m goods. Cole and Gkatzelis (2018) and Barman et al. (2018) defined upper bounds on the NSW of any integral allocation of the goods. We show that the bounds are equivalent.

Cole and Gkatzelis (2018) defined their upper bound via spending-restricted Fisher markets: Each agent has one unit of money and each good has one unit of supply. Goods can be allocated fractionally and $x_{i,j}$ is the fraction of good j allocated to agent i. A solution to the market is an allocation x and a price p_j for each good j such that

1. Each agent spends all her money i.e., $\sum_j x_{i,j} p_j = 1$.
2. An agent i spends money only on goods with maximum bang-per-buck i.e., $x_{i,j} > 0$ implies $u_{i,j}/p_j = \alpha_i$ where $\alpha_i = \max_\ell u_{i,\ell}/p_\ell$.
3. Goods with price less than 1 (low-price goods) are sold completely. Formally, let $S_s = \{j \mid p_j \le 1\}$. Then for all $j \in S_s$, $\sum_i x_{i,j} = 1$.
4. Exactly one unit of money is spent on each good with price at least 1 (large good). Formally, let $S_\ell = \{j \mid p_j > 1\}$. Then for all $j \in S_\ell$, $\sum_i x_{i,j} p_j = 1$.

The last constraint is the spending constraint and gives the market its name. Cole and Gkatzelis (2018) showed that the NSW of any integral allocation of goods to agents is at most

$$\text{CG-UB} := \left(\prod_{j \in S_\ell} p_j \prod_i \alpha_i \right)^{1/n}.$$

The following bound is implicit in the work of Barman et al. (2018). For any scaling vector $\alpha = (\alpha_1, \ldots, \alpha_n)$, define uniform utilities u_j by $u_j = \max_i u_{i,j}/\alpha_i$. For a set S of more than $m - n$ goods, let

$$a(S) = \frac{\sum_{j \in S} u_j}{|S| - (m-n)} = \frac{\sum_{j \in S} u_j}{n - |\overline{S}|} ;$$

note that $|S| - (m-n) = n - (m - |S|) = n - |\overline{S}|$. So $a(S)$ is the amount per agent if the total utility of the goods in S is distributed uniformly over $n - |\overline{S}|$ agents. Finally, let

$$\mathcal{S}_\alpha = \{\, S \,;\, |S| > m - n \text{ and } u_j > a(S) \text{ for } j \notin S \,\}.$$

Then the BKV-bound is defined as follows:

$$\text{BKV-UB} := \min_{\alpha > 0} \min_{S \in \mathcal{S}_\alpha} \left(\prod_{j \notin S} u_j \cdot a(S)^{n - |\overline{S}|} \cdot \prod_i \alpha_i \right)^{1/n}.$$

Lemma 19. BKV-UB *is an upper bound on the Nash social welfare of any integral allocation.*

Proof. Scaling the utilities of agent i by α_i does not change the optimal allocation and changes the NSW of any allocation by $(\prod_i \alpha_i)^{1/n}$. Replacing $u_{i,j}$ by $u_j = \max_h u_{h,j}$ for every agent i can only increase NSW. Allowing to allocate the goods in S fractionally can only increase NSW. Since S is such that $u_j > a(S)$ for $j \notin S$, the optimal partially fractional allocation is to allocate each u_j, $j \notin S$, to a distinct agent and to allocate $a(S)$ to each one of the remaining agents. □

Lemma 20. *For fixed α, the BKV-bound is minimized for $S \in \mathcal{S}_\alpha$ satisfying $u_j > a(S)$ for $j \notin S$ and $u_j \le a(S)$ for $j \in S$. This S is unique.*

Proof. Assume $u_h > a(S)$ for some $h \in S$. Let $T = S - h$. Then

$$\left(u_h a(T)^{n-|\overline{T}|}\right)^{1/(n-|\overline{S}|)} < a(S),$$

since the LHS is the geometric mean of u_h and $n - |\overline{T}|$ copies of $a(T)$ and the RHS is equal to their arithmetic mean; note that $(u_h + (n - |\overline{T}|)a(T))/(n - |\overline{S}|) = a(S)$.

We can determine S greedily. Start with S equal to the set of all goods. As long as there is a $j \in S$ such that $u_j > a(S)$, remove j from S. For[9] such a j, $a(S \setminus j) \le a(S)$ and hence any candidate for removal stays a candidate for removal. So the removal process always ends up with the same S. Also note that for $|\overline{S}| = n - 1$, $a(S) = \sum_{j \in S} u_j \ge u_j$ for all $j \in S$ and hence the process stops before all goods are removed from S. □

Lemma 21. BKV-UB $\le$ CG-UB.

Proof. Consider a solution $(x_{i,j}, p_j)$ to the spending-restricted Fisher market. The scaling vector for the BKV-UB is now defined as $\alpha_i = \max_j u_{i,j}/p_j$. Let $\overline{u}_{i,j} = u_{i,j}/\alpha_i$ be the scaled utilities. Then $\overline{u}_{i,j} \le p_j$ and $\overline{u}_{i,j} = p_j$ whenever $x_{i,j} > 0$. Let $\overline{u}_j = \max_i \overline{u}_{i,j}$; then $\overline{u}_j = p_j$ since for every j, $x_{i,j} > 0$ for at least one i. Since the total money spent is n, one unit is spent on each good in S_ℓ, and p_j is spent on good $j \in S_s$, we have

$$n = \sum_{j \in S_s} p_j + |S_\ell|$$

and hence

$$a(S_s) = \frac{\overline{u}(S_s)}{n - |\overline{S_s}|} = \frac{\sum_{j \in S_s} p_j}{n - |S_\ell|} = 1.$$

Since $p_j > 1$ for $j \in S_\ell$ and $p_j \le 1$ for $j \in S_s$, the BKV-UB is minimized for the set $S_s \in S_\alpha$ and hence

$$\text{BKV-UB} \le \left(\prod_{j \in S_\ell} p_j \cdot a(S_s)^{n-|S_\ell|} \cdot \prod_i \alpha_i\right)^{1/n} = \left(\prod_{j \in S_\ell} p_j \prod_i \alpha_i\right)^{1/n} = \text{CG-UB}.$$

□

9. Let $k = n - |\overline{S}|$. Then $a(S \setminus j) = (u(S) - u_j)/(k-1) \le u(S)/k = a(S)$ iff $u(S) \le k u_j$ iff $a(S) \le u_j$.

For a set S of more than $m-n$ goods, let P_S be the following minimization problem in variables α_i and u_j.

$$\begin{array}{lll} \text{minimize} & f_S(\alpha,u) := \sum_{j \notin S} \ln u_j + (n - |\overline{S}|)\ln a(S) + \sum_i \ln \alpha_i & \\ \text{subject to} & u_j \ge u_{i,j}/\alpha_i & \text{for all } i \text{ and } j \\ & u_j \ge a(S) & \text{for } j \notin S \end{array}$$

If P_S is feasible, let b_S be the optimum objective value and let (α^S, u^S) be an optimum solution. If S is the set of all goods, $\alpha_i = 1$ for all i and $u_j = \max_i u_{i,j}$ is feasible solution. Let S^* be such that (1) P_{S^*} is feasible, (2) b_{S^*} is minimum, and (3) among the S satisfying the two constraints, S has largest cardinality.

Lemma 22. *For $S = S^*$, $u_j^S > a(S)$ for $j \notin S$, $u_j^S \le a(S)$ for $j \in S$, and $u_j^S = \max_i u_{i,j}/\alpha_i$.*

Proof. Assume first that $u_h^S > a(S)$ for some $h \in S$. Consider $T = S - h$. Then (α^S, u^S) is a feasible solution of P_T and $b_T < b_S$ by the proof of Lemma 20.

Assume next that $u_h^S = a(S)$ for some $h \notin S$. Let $T = S \cup h$. Then (α^S, u^S) is a feasible solution for P_T and $b_T = b_S$, a contradiction to the choice of S.

Assume $u_j^S > \max_i u_{i,j}/\alpha_i$ for some j. Since $u_j^S > a(S)$ if $j \notin S$, we may decrease u_j^S, staying feasible and decreasing the objective. □

Lemma 23. *Let $S = S^*$. Then (α^S, S) defines the BKV-bound.*

Proof. Let $(\alpha^{\mathrm{BKV}}, S^{\mathrm{BKV}})$ define the BKV-bound and let $u_j^{\mathrm{BKV}} = \max_i u_{i,j}/\alpha_i^{\mathrm{BKV}}$ for all j. Then $(\alpha^{\mathrm{BKV}}, u^{\mathrm{BKV}})$ is a feasible solution of $P_{S^{\mathrm{BKV}}}$ and

$$f_{S^{\mathrm{BKV}}}(\alpha^{\mathrm{BKV}}, u^{\mathrm{BKV}}) = \frac{1}{n} \ln \text{BKV-UB}.$$

Therefore

$$b_{S^*} \le b_{S^{\mathrm{BKV}}} \le \frac{1}{n} \ln \text{BKV-UB}.$$

Conversely, let $S = S^*$, let (α^S, u^S) be an optimal solution to P_S, and let S_{α^S} be the set minimizing the BKV-bound for α^S Then $S_{\alpha^S} = S^*$ by Lemma 22 and hence

$$\frac{1}{n} \ln \text{BKV-UB} \le b_{S^*}.$$

□

Lemma 24. CG-UB $\le$ BKV-UB.

Proof. Let $S = S^*$ and let (α^S, u^S) be an optimal solution of problem P_S. We have shown above that (α^S, S) defines the BKV-bound, and that $u_j^S > a(S)$ for $j \notin S$, $u_j^S \le a(S)$ for $j \in S$, and $u_j^S = \max_i u_{i,j}/\alpha_i$. Let $k = n - |\overline{S}|$.

The KKT conditions are necessary conditions for the optimum. Let $z_{i,j} \geq 0$ for all i and j, and $y_j \geq 0$ for $j \notin S$ be the multipliers. Then we need to have (write $u_j\alpha_i \geq u_{i,j}$ for the inequalities)

$$\begin{aligned}
1/u_j^S &= \sum_i z_{i,j}\alpha_i^S + y_j && \text{for } j \notin S \\
\frac{1}{a(S)} &= \sum_i z_{i,j}\alpha_i^S && \text{for } j \in S \\
1/\alpha_i^S &= \sum_j z_{i,j}u_j^S && \text{for all } i \\
z_{i,j} > 0 \Rightarrow \alpha_i^S u_j^S &= u_{i,j} && \text{for all } i \text{ and } j \\
y_j > 0 \Rightarrow u_j^S &= a(S) && \text{for all } j \notin S.
\end{aligned}$$

Define $p_j = u_j^S/a(S)$ and $x_{i,j} = a(S)z_{i,j}\alpha_i^S$ and call p_j the price of good j and $x_{i,j}$ the fraction of good j allocated to agent i. The bang-per-buck ratio of agent i is

$$\alpha_i = \max_j \frac{u_{i,j}}{p_j} = \frac{\alpha_i^S u_j^S}{u_j^S/a(S)} = a(S)\alpha_i^S.$$

We now rewrite and interpret the optimality conditions.

- Since $u_j^S > a(S)$ for $j \notin S$, $y_j = 0$ for $j \notin S$.
- The first condition becomes $1 = \sum_i x_{i,j}p_j$, i.e., exactly one unit of money is spent on each good $j \notin S$. Note that $p_j > 1$ for such goods.
- The third condition becomes $1 = \sum_j x_{i,j}p_j$, i.e., every agent spends exactly one unit of money.
- The second condition becomes $\sum_i x_{i,j} = 1$ for all $j \in S$, i.e. goods in S are completely allocated, but not overallocated.
- $x_{i,j} > 0$ implies $z_{i,j} > 0$ which in turn implies that $\alpha_i^S u_j^S = u_{i,j}$. Hence $u_{i,j}/p_j = a(S)\alpha_i^S = \alpha_i$ which means that good j is allocated to i only if it has the maximum bang-per-buck ratio.

This shows that the pair (x, p) is a solution to the spending-restricted Fisher market and the CG-bound for this solution is

$$\text{CG-UB} \leq \prod_{j \notin S} \frac{u_j^S}{a(S)} \cdot \prod_i a(S)\alpha_i^S = \prod_{j \notin S} u_j^S \cdot a(S)^{n-|\overline{S}|} \cdot \prod_i \alpha_i^S = \text{BKV-UB}.$$

□

We have now shown the main theorem of this section.

Theorem 7. *The CG-bound and the BKV-bound have the same value.*

6. Discussion

In this paper, we studied the problem of allocating a set of indivisible goods among agents with CASC valuation functions. CASC is a natural elementary class capturing concavity or submodularity. The CASC functions generalize additive functions and are a special case of submodular functions. We presented a polynomial-time algorithm that approximates the optimal NSW up to a factor of 1.445, which matches with the state-of-the-art approximation factor for additive valuations. Furthermore, we showed several interesting extensions. For example, we showed that the computed allocation satisfies EF1 up to a factor of $2+\varepsilon$, and for instances without utility caps, it is also Pareto-optimal.

We conclude with two interesting open questions. First, there is a considerable gap in the lower bound (1.069) on the approximation factor that is hard to achieve (see Garg et al. (2019)) and the approximation factor of 1.445, even for the additive valuations. It will be interesting to close this gap. Second, for the more general submodular valuations, Li and Vondrák (2021) recently obtained a 380-approximation algorithm for the optimal NSW. Clearly, the approximation factor is enormous. An exciting direction for future work is to reduce the approximation factor for this class.

Acknowledgments

We thank the reviewers and the associate editor for their thorough reviews and comments. We thank Hannaneh Akrami for a careful reading of the paper. Jugal Garg was supported by NSF grant CCF-1942321 (CAREER). Martin Hoefer acknowledges financial support by DFG grant Ho 3831/5-1.

References

N. Anari, S. O. Gharan, A. Saberi, and M. Singh. Nash Social Welfare, Matrix Permanent, and Stable Polynomials. In *8th Innovations in Theoretical Computer Science Conference (ITCS)*, pages 1–12, 2017.

N. Anari, T. Mai, S. O. Gharan, and V. V. Vazirani. Nash social welfare for indivisible items under separable, piecewise-linear concave utilities. In *Proc. 29th Symp. Discrete Algorithms (SODA)*, pages 2274–2290, 2018.

S. Barman, S. K. Krishnamurthy, and R. Vaish. Finding fair and efficient allocations. In *Proc. 19th Conf. Economics and Computation (EC)*, pages 557–574, 2018.

S. Barman, U. Bhaskar, A. Krishna, and R. G. Sundaram. Tight approximation algorithms for p-mean welfare under subadditive valuations. In *Proc. 28th European Symp. Algorithms (ESA)*, 2020.

E. Budish. The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes. *J. Political Economy*, 119(6):1061–1103, 2011.

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

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

B. R. Chaudhury, J. Garg, and R. Mehta. Fair and efficient allocations under subadditive valuations. In *Proc. 35th Conf. Artif. Intell. (AAAI)*, 2021.

R. Cole and V. Gkatzelis. Approximating the Nash social welfare with indivisible items. *SIAM J. Comput.*, 47(3):1211–1236, 2018.

R. Cole, N. Devanur, V. Gkatzelis, K. Jain, T. Mai, V. Vazirani, and S. Yazdanbod. Convex program duality, Fisher markets, and Nash social welfare. In *Proc. 18th Conf. Economics and Computation (EC)*, 2017.

N. Devanur, C. Papadimitriou, A. Saberi, and V. Vazirani. Market equilibrium via a primal–dual algorithm for a convex program. *J. ACM*, 55(5), 2008.

S. Dobzinski, U. Feige, and M. Feldman. Are gross substitutes a substitute for submodular valuations? In *Proc. 22nd Conf. Economics and Computation (EC)*, pages 390–408. ACM, 2021.

J. Edmonds and R. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *J. ACM*, 19(2):248–264, 1972.

E. Eisenberg and D. Gale. Consensus of subjective probabilities: The pari-mutuel method. *Ann. Math. Statist.*, 30(1):165–168, 1959.

M. Feldman, N. Gravin, and B. Lucier. Combinatorial Walrasian equilibrium. *SIAM J. Comput.*, 45(1):29–48, 2016.

J. Garg, M. Hoefer, and K. Mehlhorn. Approximating the Nash social welfare with budget-additive valuations. In *Proc. 29th Symp. Discrete Algorithms (SODA)*, 2018.

J. Garg, M. Hoefer, and K. Mehlhorn. Satiation in Fisher markets and approximation of Nash social welfare. *CoRR*, abs/1707.04428v3, 2019.

J. Garg, P. Kulkarni, and R. Kulkarni. Approximating Nash social welfare under submodular valuations through (un)matchings. In *Proc. 31st Symp. Discrete Algorithms (SODA)*, pages 2673–2687, 2020.

J. Garg, E. Husic, and L. A. Végh. Approximating Nash social welfare under Rado valuations. In *Proc. 53rd Symp. Theory of Computing (STOC)*, 2021. To appear.

K. Jain and V. Vazirani. Eisenberg-Gale markets: Algorithms and game-theoretic properties. *Games Econom. Behav.*, 70(1):84–106, 2010.

E. Lee. APX-hardness of maximizing Nash social welfare with indivisible items. *Inf. Process. Lett.*, 122:17–20, 2017.

W. Li and J. Vondrák. A constant-factor approximation algorithm for Nash social welfare with submodular valuations. *CoRR*, abs/2103.10536, 2021.

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

R. M. McConnell, K. Mehlhorn, S. Näher, and P. Schweitzer. Certifying algorithms. *Comput. Sci. Rev.*, 5(2):119–161, 2011.

P. McGlaughlin and J. Garg. Improving Nash social welfare approximations. *J. Artif. Intell. Res.*, 68:225–245, 2020.

H. Moulin. Fair division in the internet age. *Annual Review of Economics*, 11, 2019.

J. Orlin. Improved algorithms for computing Fisher's market clearing prices. In *Proc. 42nd Symp. Theory of Computing (STOC)*, pages 291–300, 2010.

T. Roughgarden and I. Talgam-Cohen. Why prices need algorithms. In *Proc. 16th Conf. Economics and Computation (EC)*, pages 19–36, 2015.

L. Végh. Strongly polynomial algorithm for a class of minimum-cost flow problems with separable convex objectives. In *Proc. 44th Symp. Theory of Computing (STOC)*, pages 27–40, 2012.