On Fair Division for Indivisible Items

Bhaskar Ray Chaudhury
MPI for Informatics, Saarland Informatics Campus, Germany
s8bhrayc@stud.uni-saarland.de

Yun Kuen Cheung[1]
Singapore University of Technology and Design, Singapore
yunkuen_cheung@sutd.edu.sg

Jugal Garg[2]
Department of Industrial and Enterprise Systems Engineering
University of Illinois at Urbana-Champaign, USA
jugal@illinois.edu

Naveen Garg
Department of Computer Science, IIT Delhi, India
naveen@cse.iitd.ac.in

Martin Hoefer
Institut für Informatik, Goethe-Universität Frankfurt am Main, Germany
mhoefer@cs.uni-frankfurt.de

Kurt Mehlhorn
MPI for Informatics, Saarland Informatics Campus, Germany
mehlhorn@mpi-inf.mpg.de

Abstract

We consider the task of assigning indivisible goods to a set of agents in a fair manner. Our notion of fairness is Nash social welfare, i.e., the goal is to maximize the geometric mean of the utilities of the agents. Each good comes in multiple items or copies, and the utility of an agent diminishes as it receives more items of the same good. The utility of a bundle of items for an agent is the sum of the utilities of the items in the bundle. Each agent has a utility cap beyond which he does not value additional items. We give a polynomial time approximation algorithm that maximizes Nash social welfare up to a factor of $e^{1/e} \approx 1.445$. The computed allocation is Pareto-optimal and approximates envy-freeness up to one item up to a factor of $2 + \varepsilon$.

2012 ACM Subject Classification Theory of computation → Approximation algorithms analysis

Keywords and phrases Fair Division, Indivisible Goods, Envy-Free

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2018.25

Acknowledgements We want to thank Hannaneh Akrami for a careful reading of the paper.

[1] Most work done while the author was at MPI for Informatics, Saarland Informatics Campus. The author would like to acknowledge NRF 2018 Fellowship NRF-NRFF2018-07.
[2] Supported by NSF CRII Award 1755619.

38th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2018).
Editors: Sumit Ganguly and Paritosh Pandya; Article No. 25; pp. 25:1–25:17
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

We consider the task of dividing indivisible goods among a set of n agents in a fair manner. More precisely, we consider the following scenario. We have m distinct goods. Goods are available in several copies or items; there are k_j items of good j. The agents have decreasing utilities for the different items of a good, i.e., for all i and j

$$u_{i,j,1} \geq u_{i,j,2} \geq \ldots \geq u_{i,j,k_j}.$$

An allocation assigns the items to the agents. For an allocation x, x_i denotes the multi-set of items assigned to agent i, and $m(j, x_i)$ denotes the multiplicity of good j in x_i. Of course, $\sum_i m(j, x_i) = k_j$ for all j. The total utility of bundle x_i for agent i is given by

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

Each agent has a utility cap c_i. The capped utility of bundle x_i for agent i is defined as

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

Our notion of fairness is *Nash social welfare* (NSW) [13], i.e., the goal is to maximize the geometric mean

$$\mathrm{NSW}(x) = \left(\prod_{1 \leq i \leq n} \bar{u}_i(x_i) \right)^{1/n}$$

of the capped utilities. All utilities and caps are assumed to be integers. We give a polynomial-time approximation algorithm with approximation guarantee $e^{1/e} + \varepsilon \approx 1.445 + \varepsilon$ for any positive ε.

The problem has a long history. For divisible goods, maximizing Nash Social Welfare (NSW) for any set of valuation functions can be expressed via an Eisenberg-Gale program [8]. Notably, for *additive valuations* ($c_i = \infty$ for each agent i and $k_j = 1$ for each good j) this is equivalent to a Fisher market with identical budgets. In this way, maximizing NSW is achieved via the well-known fairness notion of competitive equilibrium with equal incomes (CEEI) [12].

For indivisible goods, the problem is NP-complete [14] and APX-hard [10]. Several constant-factor approximation algorithms are known for the case of additive valuations. They use different approaches.

The first one was pioneered by Cole and Gkatzelis [6] and uses spending-restricted Fisher markets. Each agent comes with one unit of money to the market. Spending is restricted in the sense that no seller wants to earn more than one unit of money. If the price p of a good is higher than one in equilibrium, only a fraction $1/p$ of the good is sold. Cole and Gkatzelis showed how to compute a spending restricted equilibrium in polynomial time and how to round its allocation to an integral allocation with good NSW. In the original paper they obtained an approximation ratio of $2e^{1/e} \approx 2.889$. Subsequent work [5] improved the ratio to 2.

The second approach is via stable polynomials. Anari et al. [1] obtained an approximation factor of e.

The third approach is via integral allocations that are Pareto-optimal and envy-free up to one good. It was introduced by Barman et al. [3]. An allocation is envy-free up to one good if for any two agents i and k there is a good j such that $u_i(x_k - j) \leq u_i(x_i)$, i.e., after removal of one good from k's bundle its utility for i is no larger than the utility of i's bundle for i. Caragiannis et al. [4] have shown that an allocation maximizing NSW is Pareto-optimal and envy-free up to one good. For a price vector p for the goods, the price $P(x_i)$ of a bundle is the sum of the prices of the goods in the bundle. An allocation is almost price-envy-free up to one good (ε-p-EF1) if $P(x_k - j) \le (1+\varepsilon)P(x_i)$ for all agents i and k and some good j, where ε is an approximation parameter. An allocation is MBB (maximum bang per buck) if $j \in x_i$ implies $u_{ij}/p_j = \max_\ell u_{i\ell}/p_\ell$ for all j and i. Barman et al. [3] studied allocations that are Pareto-optimal, almost price-envy-free up to one good, and MBB. They showed that such allocations are almost envy-free up to one good[3] and approximate NSW up to a factor $e^{1/e} + \varepsilon \approx 1.445 + \varepsilon$. They also showed how to compute such an allocation in polynomial time.

There are also constant-factor approximation algorithms beyond additive utilities.

Garg et al. [9] studied budget-additive utilities ($k_j = 1$ for all goods j and arbitrary c_i). They showed how to generalize the Fisher market approach and obtained an $2e^{1/2e} \approx 2.404$-approximation.

Anari et al. [2] investigated multi-item concave utilities ($c_i = \infty$ for all i and k_j arbitrary). They generalized the Fisher market and the stable polynomial approach and obtained approximation factors of 2 and e^2, respectively.

We show that the price-envy-free allocation approach can handle both generalizations combined. We obtain an approximation ratio of $e^{1/e}+\varepsilon \approx 1.445+\varepsilon$. The allocation computed by our algorithm is Pareto-optimal and guarantees $u_i(x_k - j) \le (2+\varepsilon)u_i(x_i)$ for any two agents i and k, i.e., it approximates envy-freeness up to one item up to a factor of essentially two. The approach via price-envy-freeness does not only yield better approximation ratios, it is, in our opinion, also simpler to state and simpler to analyze.

The paper is structured as follows. In Section 2 we give the algorithm and analyze its approximation ratio (Section 2.3), guarantee to individual agents (Section 2.4), and running time (Section 2.5). In Section 3 we show that the analysis is essential tight by establishing a lower bound of 1.44 on the approximation ratio of the algorithm, in Section 4 we discuss certification of the approximation ratio, and in Section 5 we show that for the multi-copy case and the capped case optimal allocations are not necessarily envy-free up to one good.

2 Algorithm and Analysis

Let us recall the setting. Items are indivisible. There are n agents and m goods. There are k_j items or copies of good j. Let $M = \sum_j k_j$ be the total number of items. The agents have decreasing utilities for the different items of a good, i.e., for all i and j

$$u_{i,j,1} \ge u_{i,j,2} \ge \ldots \ge u_{i,j,k_j}.$$

For an allocation x, x_i denotes the multi-set of items assigned to agent i, and $m(j, x_i)$ denotes the multiplicity of good j in x_i. The total utility of bundle x_i for agent i is given by

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

[3] Consider two bundles x_k and x_i and assume $P(x_k - j) \le (1+\varepsilon)P(x_i)$ for some $j \in x_k$. Let $\alpha_i = \max_\ell u_{i\ell}/p_\ell$. Then $u_i(x_k - j) = \sum_{\ell \in x_k - j} u_{i\ell} \le \alpha_i \sum_{\ell \in x_k - j} p_\ell \le (1+\varepsilon)\alpha_i \sum_{\ell \in x_i} p_\ell = (1+\varepsilon)\sum_{\ell \in x_i} u_{i\ell}$.

Each agent has a utility cap c_i. The capped utility of bundle x_i for agent i is defined as

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

Following [9], we assume w.l.o.g. $u_{i,j,\ell} \le c_i$ for all i, j, and ℓ. In the algorithm, we ensure this assumption by capping every $u_{i,*,*}$ at c_i. All utilities and caps are assumed to be integers.

2.1 A Reduction to Rounded Utilities and Caps

Let $r \in (1, 3/2]$. 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 and u agree. Similarly, for c_i let d_i be the next larger power of r. It is well-known that 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 γ. Then x approximates the NSW for the original problem up to a factor γr.*

Proof. Let x^* be an optimal allocation for the original problem. Let us write $\mathrm{NSW}(x^*, u, c)$ for the Nash social welfare of the allocation x^* with respect to the utilities u and caps c. Define $\mathrm{NSW}(x, u, c)$, $\mathrm{NSW}(x^*, v, d)$, and $\mathrm{NSW}(x, v, d)$ analogously. We need to upper bound $\mathrm{NSW}(x^*, u, c)/\mathrm{NSW}(x, u, c)$. Since $u \le v$ and $c \le d$ componentwise, $\mathrm{NSW}(x^*, u, c) \le \mathrm{NSW}(x^*, v, d)$. Since x approximates the NSW for the rounded problem up to a factor γ, $\mathrm{NSW}(x^*, v, d) \le \gamma \mathrm{NSW}(x, v, d)$. Since $v \le ru$ and $d \le rc$ componentwise, $\mathrm{NSW}(x, v, d) \le r\mathrm{NSW}(x, u, c)$. Thus

$$\frac{\mathrm{NSW}(x^*, u, c)}{\mathrm{NSW}(x, u, c)} \le \frac{\gamma \mathrm{NSW}(x, v, d)}{\mathrm{NSW}(x, v, d)/r} = \gamma r.$$ ◀

2.2 The Algorithm

Barman et al. [3] gave a highly elegant approximation algorithm for the case of a single copy per good and no utility caps. We generalize their approach. The algorithm uses an approximation parameter $\varepsilon \in (0, 1/4]$. Let $r = 1 + \varepsilon$. The nonzero utilities are assumed to be powers of r.

The algorithm maintains an integral assignment x, a price p_j for each good, and an MBB-ratio[4] α_i for each agent. Of course, $\sum_i m(j, x_i) = k_j$ for each good j. The prices, MBB-ratios, and multiplicity of goods in bundles are related through the following inequalities:

$$\frac{u_{i,j,m(j,x_i)+1}}{p_j} \le \alpha_i \le \frac{u_{i,j,m(j,x_i)}}{p_j}, \tag{1}$$

i.e., if $u_{i,j,\ell}/p_j > \alpha_i$, then at least ℓ copies of j are allocated to agent i and if $u_{i,j,\ell}/p_j < \alpha_i$, then less than ℓ copies of j are allocated to agent i. If no copy of good j is assigned to i, the upper bound for α_i is infinity. If all copies of good j are assigned to i, the lower bound for α_i is zero. Note that if α_i is equal to its upper bound in (1), we may take one copy of 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 (1), we may assign an additional copy of j to i without violating the inequality as the lower bound becomes the new upper bound.

[4] In the case of one copy per good, $\alpha_i = u_{i,j}/p_j$ whenever (the single copy of) good j is assigned to i and $\alpha_i \ge u_{i,\ell}/p_\ell$ for all goods ℓ. Thus α_i is the maximum utility per unit of money (maximum bang per buck (MBB)) that agent i can get.

Since (1) must hold for every good j, α_i must lie in the intersection of the intervals for the different goods j, i.e.,

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

The value of bundle x_i for i is given by[5]

$$P_i(x_i) = \frac{u_i(x_i)}{\alpha_i} = \frac{1}{\alpha_i} \sum_j \sum_{1 \le \ell \le m(j,x_i)} u_{i,j,\ell}. \tag{2}$$

Definitions (1) and (2) are inspired by Anari et al [2]. 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 (1) 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 (1).

An agent i is *capped* if $u_i(x_i) \ge c_i$ and is *uncapped* otherwise.

The algorithm 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-values are set to one. Note that this setting guarantees (1) for every pair (i,j). Also, all initial prices and MBB-values are powers of r. It is an invariant of the algorithm that prices are powers of r. Only the final price increase in the main-loop may destroy this invariant.

After initialization, the algorithm enters a loop. We need some more definitions. An agent i is a *least spending* uncapped agent if it 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 item if $P_k(x_k - j) > (1+\varepsilon) \cdot P_i(x_i)$ for every good $j \in x_k$. Recall that x_k is a multi-set. In the multi-set $x_k - j$, the number of copies of good j is reduced by one, i.e., $m(j, x_k - j) = m(j, x_k) - 1$. Therefore $P_k(x_k - j) = P_k(x_k) - u_{k,j,m(j,x_k)}/\alpha_k$. An allocation is *$\varepsilon$-p-envy free up to one item (ε-p-EF1)* if for every uncapped agent i and every other agent k there is a good j such that $P_k(x_k - j) \le (1+\varepsilon)P_i(x_i)$.

We also need the notion of the *tight graph*. It 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 $\alpha_i = u_{ijm(j,x_i)+1}/p_j$, 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 $\alpha_i = u_{ijm(j,x_i)}/p_j$, i.e., α_i is at its upper bound for the pair (i,j). Note that necessarily $m(j,x_i) \ge 1$ in the latter case, since otherwise good j does not impose an upper bound for α_i.

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 $P_{a_\ell}(x_{a_\ell} - g_\ell) \le (1+\varepsilon)P_i(x_i)$ for $1 \le \ell < h$.

Let i be the least spending uncapped agent. We perform a breadth-first search in the tight graph starting at i. If the BFS discovers an improving path starting at i, we use the shortest such path to improve the allocation. Note that if i ε-*p*-envies some node 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 BFS discovers an improving path starting at i or not.

Assume first that 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 $P_{a_{h-1}}(x_{a_{h-1}} + g_h - g_{h-1}) \le$

[5] In the 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 for the value of a bundle, 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.

Algorithm 1: Approximate Nash Social Welfare for Multi Item Concave Utilities with Caps.

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

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

1 **for** i, j, ℓ **do**
2     $u_{i,j,\ell} \leftarrow \min(c_i, u_{i,j,\ell})$
3 **for** $j \in G$ **do**
4   **for** $\ell \in [k_j]$ *in increasing order* **do**
5   assign the ℓ-th copy of j to $i_0 = \mathrm{argmax}_i\, u_{ij,m(j,x_i)+1}$;
6   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
7 **for** $i \in A$ **do**
8     $\alpha_i = 1$
9 **while** *true* **do**
10   **if** *allocation x is ε-p-EF1* **then**
11   **break** from the loop and terminate
12   Let i be a least spending uncapped agent
13   Perform a BFS in the tight graph starting at i
14   **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**
15   Set $\ell \leftarrow h$
16   **while** $\ell > 0$ *and* $P_{a_\ell}(x_{a_\ell} - g_\ell) > (1+\varepsilon)P_i(x_i)$ **do**
17   remove g_ℓ from x_{a_ℓ} and assign it to $a_{\ell-1}$; $\ell \leftarrow \ell - 1$
18   **else**
19   Let S be the set of goods and agents that can be reached from i in the tight graph
20         $\beta_1 \leftarrow \min_{k \in S;\ j \notin S} \alpha_k / (u_{k,j,m(j,x_k)+1}/p_j)$ (add a good to S)
21         $\beta_2 \leftarrow \min_{k \notin S;\ j \in S} (u_{k,j,m(j,x_k)}/p_j)/\alpha_k$ (add an agent to S)
22         $\beta_3 \leftarrow \frac{1}{r^2 P_i(x_i)} \max_{k \notin S} \min_{j \in x_k} P_k(x_k - j)$ (i is happy)
23         $\beta_4 \leftarrow r^s$, 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)
24         $\beta \leftarrow \min(\beta_1, \beta_2, \max(1, \beta_3), \beta_4)$
25   multiply all prices of goods in S by β and divide all MBB-values of agents in S by β
26   **if** $\beta_3 \le \min(\beta_1, \beta_2, \beta_4)$ **then**
27   break from the while-loop

$(1+\varepsilon)P_i(x_i)$ we stop. Otherwise, we take g_{h-1} away from a_{h-1} and assign it to a_{h-2}. If we now have $P_{a_{h-2}}(x_{a_{h-2}} + g_{h-1} - g_{h-2}) \le (1+\varepsilon)P_i(x_i)$ we stop. Otherwise, ... 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 $P_{a_{h'}}(x_{a_{h'}} + g_{h'+1} - g_{h'}) \le (1+\varepsilon)P_i(x_i)$. If h' exists, then we take a copy of g_ℓ away from a_ℓ and assign it to $a_{\ell-1}$ for $h' < \ell \le h$. If h' does not exist, we do so for $1 \le \ell \le h$. Let us call the above a sequence of swaps.

▶ **Lemma 2.** *Consider an execution of lines (15) to (17) 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,*

- $P_{a_h}(x_{a_h}) \ge P_{a_h}(x'_{a_h}) > (1+\varepsilon)P_i(x_i)$,

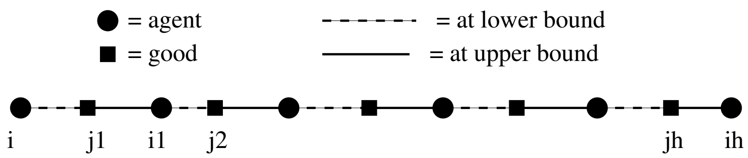

Figure 1 An improving path. Agents and goods alternate on the path and the path starts and ends with an agent. For the solid edges (j,i), α_i is at its upper bound for the pair (i,j) and for the dashed edges (i,j), α_i is at its lower bound for the pair (i,j).

- $P_{a_{h'}}(x'_{a_{h'}} - g_{h'}) = P_{a_{h'}}(x_{a_{h'}} + g_{h'+1} - g_{h'}) \leq (1+\varepsilon)P_i(x_i)$ *if* $h' \geq 1$
- $P_{a_0}(x'_{a_0} - g_1) = P_{a_0}(x_{a_0}) \leq (1+\varepsilon)P_i(x_i)$ *if* $h' = 0$.
- $P_{a_\ell}(x'_{a_\ell}) = P_{a_\ell}(x_{a_\ell} + g_{\ell+1} - g_\ell) > (1+\varepsilon)P_i(x_i)$ *and* $P_{a_\ell}(x'_{a_\ell} - g_{\ell+1}) = P_{a_\ell}(x_{a_\ell} - g_\ell) \leq (1+\varepsilon)P_i(x_i)$ *for* $h' < \ell < h$.
- $P_{a_\ell}(x'_{a_\ell} - g_\ell) = P_{a_\ell}(x_{a_\ell} - g_\ell) \leq (1+\varepsilon)P_i(x_i)$ *for* $0 \leq \ell < h'$.

Proof. Immediate from the above. ◀

If i is still the least spending uncapped agent after an execution of lines (15) to (17), we search for another improving path starting from i. We will show below that i can stay 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.

We come to the else-case, i.e., BFS does not discover an improving path starting at i. This implies that i does not ε-p-envy any agent that it can reach in the tight graph. We then increase some prices and decrease some MBB-values. 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 by 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 by j. In either case, j belongs to S. ◀

We multiply all prices of goods in S and divide all MBB-values of agents in S by a common factor $t \geq 1$. What is the effect?

- Let $u_{k,j,(j,x_k)+1}/p_j \leq \alpha_k \leq u_{k,j,m(j,x_k)}/p_j$ be the inequality (1) for the pair (k,j). The endpoints do not move if $j \notin S$ and are divided by t for $j \in S$. Similarly, α_k does not move if $k \notin S$ and are divided by t 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 t. For agents outside S, $P_k(x_k)$ stays unchanged.

How is the common factor t 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.** *Prices and MBB-values 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-values are equal to one. If prices and MBB-values are powers of r before a price update, β_1, β_2, and β_4 are powers of r. Thus prices and MBB-values 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 item.

▶ **Lemma 5.** *Assume $\varepsilon \le 1/4$. When the algorithm terminates, x is a 4ε-p-EF1 allocation.*

Proof. Let q be the price vector after the price increase and let h be the least spending uncapped agent after the increase; $h = i$ is possible. We first show that that $Q_i(x_i) \le rQ_h(x_h)$. This is certainly true if $h = i$. If $h \notin S$, since the price increase is limited by β_4, we have

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

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

If the algorithm terminates, we have $\beta_3 \le \beta_4$. Consider any agent k. Then, for $k \in S$,

$$Q_k(x_k - j_k) \le (1 + \varepsilon)Q_i(x_i) \le (1 + \varepsilon) \cdot r \cdot Q_h(x_h)$$

and, for $k \notin S$,

$$Q_k(x_k - j_k) = P_k(x_k - j_k) \le \beta_3(1 + \varepsilon)rP_i(x_i) = (1 + \varepsilon)rQ_i(x_i) \le (1 + \varepsilon) \cdot r^2 \cdot Q_h(x_h).$$

Thus we are returning an allocation that is $((1 + \varepsilon)r^2 - 1)$-$q$-EF1. Finally, note that $(1 + \varepsilon)r^2 = (1 + \varepsilon)^3 \le (1 + 4\varepsilon)$ for $\varepsilon \le 1/4$. ◀

▶ Remark. We want to point out the differences to the algorithm by Barman et al. Our definition of alternating path is more general than theirs since it needs to take into account that the number of items of a particular good assigned to an agent may change. For this reason, we need to maintain the MBB-ratio explicitly. In the algorithm by Barman et al. the MBB ratio of agent i is equal to the maximum utility to price ratio $\max_j u_{ij}/p_j$ 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 uncapped agent and not from the least agent.

2.3 Analysis of the Approximation Factor

The analysis refines the analysis given by Barman et al. Let (x^{alg}, p, α) denote the allocation and price and MBB vector returned by the algorithm. Recall that x^{alg} is γ-p-EF1 with $\gamma = 4\varepsilon$ with respect to p and (1) holds for every i. We scale all the utilities of agent i and its utility cap by α_i, i.e., we replace $u_{i,j,\ell}$ by $u_{i,j,\ell}/\alpha_i$ and c_i by c_i/α_i and use $u_{i,j,\ell}$ and c_i also for the scaled utilities and scaled utility cap. The scaling does not change the integral allocation maximizing Nash Social Welfare. Inequality (1) becomes

$$\frac{u_{i,j,m(j,x_i^{alg})+1}}{p_j} \le 1 \le \frac{u_{i,j,m(j,x_i^{alg})}}{p_j}, \tag{3}$$

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

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

All $u_{i,*,*}$ are at most c_i.

Let A_c and A_u be the set of capped and uncapped agents in x^{alg}, let $c = |A_c|$ and $n - c = |A_u|$ be their cardinalities. 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 $\ell = u_{n-c}(x_{n-c}^{alg})$ be the minimum utility of a bundle assigned to an uncapped agent. The capped agents are numbered $n - c + 1$ to n. Let x^* be an integral allocation maximizing Nash social welfare.

We define an auxiliary problem with $\sum_j k_j$ 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,

$$v_{*,(i,j,\ell)} = u_{i,j,\ell}, \tag{5}$$

where v is the utility function for the auxiliary problem. The cap of agent i is c_i. Since v is uniform, 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 an optimal allocation for the auxiliary problem.

▶ **Lemma 6.** *We have:*

(a) $\sum_i u_i(x_i^*) \le \sum_i u_i(x_i^{alg}) = \sum_{i,j,1 \le \ell \le m(j,x_i^{alg})} v_{*,(i,j,\ell)}$.

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

(c) x^{alg} *is Pareto-optimal.*

Proof. We can obtain x^* from x^{alg} by moving copies of goods.

Set $x \leftarrow x^{alg}$. Consider any good j. As long as the multiplicities of j in the bundles of x and x^* are not the same, identify two agents i and k, where x_i contains more copies of j than x_i^* and x_k contains fewer copies of j than x_k^*, and move a copy of j from i to k. Each copy taken away has a utility of at least p_j, each copy assigned additionally has a utility of at most p_j. Thus the total utility cannot go up by reassigning. This proves (a).

For part (b), we interpret x^{alg} as an allocation for the auxiliary problem; goods (i, j, ℓ) with $1 \le \ell \le m(j, x_i^{alg})$ are allocated to agent i. We then move goods exactly as in (a). We obtain an allocation $\hat{x}$ for the auxiliary problem with $u_i(x_i^*) \le v(\hat{x}_i)$ for all i.

For part (c), assume that x^{alg} is not Pareto-optimal. Then there is an integral allocation y with $u_i(y_i) \ge u_i(x_i^{alg})$ for all i and at least one strict inequality. These inequalities are not affected by our scaling of the utilties. However, the reasoning of part (a) applied to y and x^{alg} shows $\sum_i u_i(y_i) \le \sum_i u_i(x_i^{alg})$ for all i for the scaled utilities. ◀

We stress that Lemma 6 refers to the scaled utilities. For the scaled utilities x^{alg} maximizes social welfare. It does not do so for the unscaled utilities.

For any agent i, let $b_i \in x_i^{alg}$ be such that $u_i(x_i^{alg} - b_i) \le (1+\gamma)\ell$. Note that $u_i(x_i^{alg} - b_i) = u_i(x_i^{alg}) - u_{i,b_i,m(b_i,x_i^{alg})}$. Let $B = \{\, (i, b_i, m(b_i, x_i^{alg}))\,;\, 1 \le i \le n \,\}$ be the goods in the auxiliary problem corresponding to the b_i's. We now consider allocations for the auxiliary problem that are allowed to be partially fractional. We require that the goods in B are allocated integrally and allow all other goods to be assigned fractionally. For convenience of notation, let $g_i = (i, b_i, m(b_i, x_i^{alg}))$. The following lemma is crucial for the analysis.

▶ **Lemma 7.** *There is an optimal allocation for the relaxed auxiliary problem in which 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 to which no good in B is allocated. Then g_i is allocated to some agent k different from i. Since $b_i \in x_i^{alg}$, $v(g_i) = u_{i,b_i,m(b_i,x_i^{alg})} \le c_i$. The 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 $v(g_i) > v(x_i^{optrel})$ and hence $c_i \ge v(g_i) > v(x_i^{optrel})$, the product $\bar{v}_i(x_i) \cdot \bar{v}_k(x_k)$ changes from

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

to

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

The arguments of the min in the lower line are componentwise larger than those of the min in the upper line. We have now modified x^{optrel} such that the NSW did not decrease and the number of agents owning their own good increased. The above applies as long as there is an agent owning no good in B.

So assume every agent i owns a good in B, but not necessarily g_i. Let i be such that $v(g_i)$ is largest among all goods g_i that are not allocated to their i. 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 $\ell \ne i$ 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 $v(g_\ell) \le v(g_i)$ and all other goods assigned to i are divisible. We have now modified x^{optrel} such that the NSW did not decrease and the number of agents owning their own good increased. We continue in this way until g_i is allocated to i for every i. ◀

Let x^{optrel} be an optimal allocation for the relaxed auxiliary problem in which good g_i is contained in the bundle x_i^{optrel} for every i. Let α be such that

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

is the minimum value of any agent that is uncapped in x^{optrel}. Let $\alpha = \infty$, if every agent is capped in x^{optrel}. Let A_c^{optrel} and A_u^{optrel} be the set of capped and uncapped agents in x^{optrel}. Let h be such that $u_h(x_h^{alg}) > \alpha\ell \geq u_{h+1}(x_{h+1}^{alg})$.

▶ **Lemma 8.** *For $i \leq h$, $v(x_i^{optrel}) \leq u_i(x_i^{alg})$. For all i, $u_i(x_i^{alg}) \leq v(x_i^{optrel}) + (1+\gamma)\ell$. For $i \in A_u \cap A_c^{optrel}$, $c_i \leq \alpha\ell$ and $i \notin [h]$.*

Proof. Consider any $i \leq h$. $v(x_i^{optrel}) \leq u_i(x_i^{alg})$ is obvious, if $v(x_i^{optrel}) \leq \alpha\ell$. If $v(x_i^{optrel}) > \alpha\ell$, then $\alpha < \infty$ and hence A_u^{optrel} is non-empty. We claim that $x_i^{optrel} = \{g_i\}$, i.e., x_i^{optrel} is a singleton consisting only of g_i. Assume otherwise, then also some divisible goods are assigned to i. We can move some of them to an agent that is uncapped in x^{optrel} and has value $\alpha\ell$. This increases the NSW, a contradiction.

For the upper bound, we observe that $g_i \in x_i^{optrel}$ and $u_i(x_i^{alg} - b_i) \leq (1+\gamma)\ell$.

Consider next any $i \in A_u \cap A_c^{optrel}$. Assume $c_i > \alpha\ell$. If x^{optrel} assigns divisible goods to i, we can move some of them to an agent that is uncapped in x^{optrel} and has value $\alpha\ell$. This increases the NSW. Thus x_i^{optrel} consists only of g_i. But then $v(g_i) \leq u_i(x_i^{alg}) < c_i$ and i does not belong to A_c^{optrel}. This shows $c_i \leq \alpha\ell$. Then also $i \notin [h]$ because otherwise $c_i < u_i(x_i^{alg})$ and hence i would be capped in x^{alg}. ◀

▶ **Lemma 9.**

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

Moreover, $c_i \leq \alpha\ell$ for any $i \in A_u \cap A_c^{optrel}$.

Proof. If $v(x_i^{optrel}) \neq \alpha\ell$ then either $i \in A_c$ or $i \in A_u \cap A_c^{optrel}$ or $i \in A_u \setminus A_c^{optrel}$. In the first case, $v(x_i^{optrel}) \leq c_i$. In the second case, $v(x_i^{optrel}) = c_i \leq \alpha\ell$ and $i \notin [h]$ by Lemma 8. In the third case, $v(x_i^{optrel}) \leq u_i(x_i^{alg})$ for $i \leq h$. So assume $i > h$. Then $v(g_i) \leq u_i(x_i^{alg}) \leq \alpha\ell$ and hence all value in $v(x_i^{optrel})$ above $\alpha\ell$ would be by fractional goods. They could be reassigned for an increase in NSW. We conclude that for the agents $i \in A_u \setminus A_c^{optrel}$ with $i > h$, we have $v(x_i^{optrel}) = \alpha\ell$. ◀

We next bound $\mathrm{NSW}(x^{alg})$ from below. We consider assignments x for the auxiliary problem that agree with x^{alg} for the agents in $A_c \cup [h]$ and reassign the value $\sum_{i \in A_u - [h]} u_i(x_i^{alg})$ fractionally. Note that for any $i \in A_u - [h]$, $\ell \leq u_i(x_i^{alg}) \leq \min(c_i, \alpha\ell)$. The former inequality follows from $i \in A_u$ and the latter inequality follows from the definition of h and $i \in A_u$. We reallocate value so as to move $u_i(x_i)$ towards the bounds ℓ and $\min(c_i, \alpha\ell)$. As long as there are two agents whose value is 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 $\min(c_i, \alpha\ell)$. One agent ends up with an allocation $\beta\ell$ with $\beta \in [1, \alpha]$.

Let us introduce some more notation. Write $A_u \cap A_c^{optrel}$ 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 $A_u \setminus A_c^{optrel}$ that end up at ℓ and $\alpha\ell$ respectively. Then

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

Note that $n - c - h = s + t + 1 + |S| + |T|$. Therefore

$$\frac{\mathrm{NSW}(x^*)}{\mathrm{NSW}(x^{alg})} \le \left(\alpha^s \cdot \frac{\alpha}{\beta} \cdot \prod_{i \in S} \frac{c_i}{\ell}\right)^{1/n} \le \left(\left(\frac{s\alpha + \frac{\alpha}{\beta} + \sum_{i \in S} \frac{c_i}{\ell}}{s + 1 + |S|}\right)^{s+1+|S|}\right)^{1/n},$$

where we used the inequality between geometric mean and arithmetic mean for the second inequality.

The total mass allocated by x^{optrel} to the agents in $A_u - [h]$ is $(s+t+1)\alpha\ell + \sum_{i \in S \cup T} c_i$. The allocation x^{alg} wastes up to $(1+\gamma)\ell$ for each $i \in A_c \cup [h]$ and uses $s\ell + t\alpha\ell + \beta\ell + \sum_{i \in T} c_i + |S|\ell$ on the agents in $A_u - [h]$. Therefore

$$(s + t + 1)\alpha\ell + \sum_{i \in S \cup T} c_i \le (|A_c| + h)(1 + \gamma)\ell + s\ell + t\alpha\ell + \beta\ell + \sum_{i \in T} c_i + |S|\ell$$

and hence after rearranging, dividing by ℓ and adding α/β on both sides

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

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

$$\begin{aligned} \frac{\mathrm{NSW}(x^{optrel})}{\mathrm{NSW}(x^{alg})} &\le \left(\left(\frac{(1 + \gamma)(|A_c| + h) + s + |S| + 1}{s + 1 + |S|}\right)^{s+1+|S|}\right)^{1/n} \\ &\le \left(\frac{(1 + \gamma)n}{s + 1 + |S|}\right)^{(s+1+|S|)/n} \le e^{e^{-1/(1+\gamma)}}, \end{aligned}$$

since the maximum of $((1 + \gamma)\delta)^{1/\delta}$ is attained for $\delta = \frac{1}{(1+\gamma)} e^{1/(1+\gamma)}$ and is equal to $\exp(\exp(-1/(1 + \gamma)))$. The following table contains concrete values for small non-negative values of γ.

$1 + \gamma$	1.00	1.01	1.02	1.03	1.04
$\exp(\exp(-1/(1 + \gamma)))$	1.44467	1.44997	1.45523	1.46046	1.46566

▶ Remark. The paper [5] introduces a mathematical program for maximizing NSW in the case of additive valuations. The program has an integrality gap of $e^{1/e}$. We believe that the fact that the same expression $e^{1/e}$ appears at two places is coincidence and does not point to some hidden relationship. In particular, Barman et al.'s algorithm computes the optimal allocation for the instances which [5] uses to demonstrate the integrality gap.

2.4 Guarantees for Individual Agents

The allocation computed by our algorithm is Pareto-optimal and maximizes NSW up to a factor 1.45. It also gives any uncapped agent i the guarantee $\min_{j \in x_k} P_k(x_k - j) \le (1+\varepsilon)P_i(x_i)$ for every other agent k. This guarantee is not meaningful for agent i. We now show that it implies $\min_{j \in x_k} u_i(x_k - j) \le (2 + \varepsilon)u_i(x_i)$, i.e., the utility for i of k's bundle minus one item is essentially bounded by twice the utility of i's bundle for i. The proof shows that the additional utility for i of the items that k has in excess of i up to one item is bounded by $(1 + \varepsilon)u_i(x_i)$. In the case of one copy per good, x_k and x_i are disjoint and hence any item in x_k is in excess of i's possession of the same good.

▶ **Theorem 10.** *The allocation computed by the algorithm satisfies* $\min_{j \in x_k} u_i(x_k - j) \leq (2 + \varepsilon) u_i(x_i)$ *for any agent* k *and any uncapped agent* i.

Proof. Let g be such that $u_k(x_k - g) = \min_{j \in x_k} u_k(x_k - j)$. Then

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

2.5 Polynomial Running Time

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

▶ **Lemma 11.** *The price 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 1, all agents a_ℓ with $h' < \ell \leq h$ have a price of at least $(1 + \varepsilon) P_i(x_i)$ after the swap. Also the price of agent $a_{h'}$ does not decrease. ◀

▶ **Lemma 12.** *For any agent* k*, let* j_k *be a highest price item 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* $(1 + \varepsilon) \min_{\textit{uncapped } i} P_i(x_i)$*. Once* $\max_k P_k(x_k - j_k) \leq (1 + \varepsilon) \min_{\textit{uncapped } i} P_i(x_i)$*, the algorithm terminates.*

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

Consider any price increase which is not the last. Then $\beta_4 \leq \beta_3$. Let h be the least uncapped spender after the price increase and q be the price vector after the increase. Then $Q_h(x_h) \leq Q_i(x_i) \leq r Q_h(x_h)$. For $k \in S$, we have $min_j Q_k(x_k - j) \leq (1 + \varepsilon) Q_i(x_i) \leq (1 + \varepsilon) r Q_h(x_h)$, i.e., agents in S can become violators but we can bound how bad they can become. For the agent $k \notin S$ defining β_3, we have

$$\min_j P_k(x_\ell - j) = \beta_3 (1 + \varepsilon) r P_i(x_i) \geq (1 + \varepsilon) r Q_i(x_i) \geq (1 + \varepsilon) r Q_h(x_h)$$

and hence the worst violator stays outside S. 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.

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 $\min_j P_k(x'_k - j) \le \min_j P_k(x_k - j)$ since k looses a good and $\min_j P_\ell(x'_\ell - j) \le (1+\varepsilon)P_i(x_i)$ for all $\ell \in [0, h-1]$ by Lemma 2. ◀

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

Proof. Let i be the least spending agent. 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 [3]). The argument is similar to the argument used in the strongly polynomial algorithms for weighted matchings [7]. ◀

▶ **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 uncapped spender changes if $\beta = \beta_4$ and β_4 is at least r. So $P_i(x_i)$ increases by at least r. ◀

▶ **Theorem 15.** *The number of iterations is bounded by $n^3M^2 \log_r MU$.*

Proof. Divide the 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 receives an additional item, or we have 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 item. It may or may not keep this item until the beginning of the next subsequence. If there are more than M subsequences with i being the least spender, there must be two subsequences such that i looses an item between these subsequences. According to Lemma 2, the value of i after the swap is at least r times the minimum price of any bundle and hence at least r times the price of bundle i when i was least spender for the last time. Thus $P_i(x_i)$ increases by a factor of at least r.

We have now shown: After at most $M \cdot n^2M$ iterations with i being the least spender, $P_i(x_i)$ is multiplied by a factor r. Thus there can be at most $n^2M^2 \log_r MU$ such iterations. Multiplication by n yields the bound on the number of iterations. ◀

3 A Lower Bound on the Approximation Ratio of the Algorithm

We show that the performance of the algorithm is no better than 1.44. 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

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

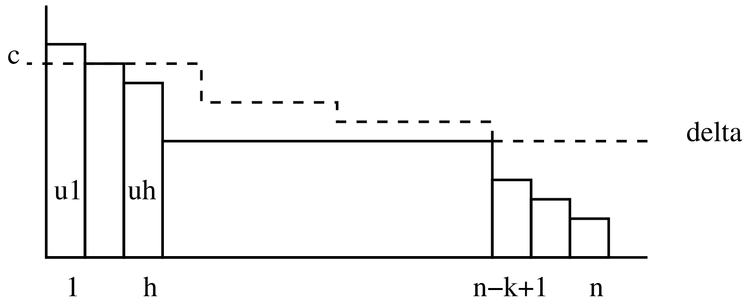

Figure 2 The allocation constructed in the proof of Theorem 16. The dashed line above agents 1 to $n-k$ indicates the utility caps. The solid rectangles visualize the values of the bundles.

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 $\mathrm{NSW}(OPT)/\mathrm{NSW}(ALG) \geq 1.440$.

4 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 more than 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 is unsatisfactory [11]. In this section, we describe an alternative 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 gap of agent i by α_i, we have (3). The user needs to understand that this scaling has no effect on the optimal allocation. As in Section 2.3, we introduce the auxiliary problem with $M = \sum_j k_j$ goods and one copy of each good. The goods have uniform utilities. The user needs to understand that the NSW of the auxiliary problem is an upper bound (Lemma 6). We are left with the task of convincing the user of an upper bound on the NSW of the auxiliary problem.

▶ **Theorem 16.** *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, and let x^{optaux} be an optimal allocation for the auxiliary problem. Then*

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

where $\delta = \left(\sum_{h+1 \leq j \leq M} u_j - \sum_{n-k+1 \leq i \leq n} c_i \right) / (n-h-k)$ and h and k are such that $h < n-k$ and $c_{n-k+1} \leq \delta < c_{n-k}$ and $\delta < u_h$. The right hand side is illustrated in Figure 2.

Proof. We insist that the goods 1 to h are allocated integrally and allow 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 $\sum_{h+1 \leq j \leq M} u_j$ 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 items 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 items, say u_ℓ and u_k, are allocated to the same agent. Then one of the first h agents is allocated no such item; 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 items 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 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 $\alpha\ell = 2$. 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 $\delta = 2$.

5 Envy-Freeness up to one Copy

For the case of additive valuations and one copy of each good, the optimal allocation is envy-free up to one good as shown in [4]. Also the allocation constructed by the algorithm by Barman et al. [3] is envy-free up to one good. In this section, we show that these properties hold neither for the multi-copy case nor for the capped case.

We first give an example for the multi-copy uncapped case. There are two agents and two goods. Good 1 has 5 copies, and good 2 has 2 copies. For the first agent, the utility vector for good 1 is $(1, 1, 0, 0, 0)$ and for good 2 is $(\delta, 0)$, where $\delta = 1/4$. For the second agent, the utility vector for good 1 is $(1, 1, 1, 0, 0)$ and for good 2 is $(1, 1)$. Then at the optimal NSW allocation, 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. Clearly, the first agent envies the second agent even after removing one copy (of either good) from the allocation of the second agent. However, $u_1(x_2) = 2 + \delta$.

What does the algorithm do? The initial assignment constructs the optimal assignment and sets $p_1 = p_2 = \alpha_1 = \alpha_2 = 1$. Agent 1 is the least spending uncapped agent. The constraints on α_1 are $[0, 1]$ by the first good and $[\delta, 1]$ 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 $\beta_1 = 1/\delta$, $\beta_2 = \infty$, $\beta_3 = 4/(2r^2) = 2/r^2$ and $\beta_4 = r^{1+\lfloor \log_r 5/2 \rfloor} \geq \beta_3$. Thus $\beta = \beta_3$. We decrease $\alpha_1$to $r^2/2 \approx 1/2$ and terminate. The optimal allocation is now ε-p-envy free up to one copy.

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 $(1, 1, 1, 1)$, but the first agent is capped at $1 + \delta$, while the second agent is uncapped. Again $\delta = 1/4$. Then the optimal NSW allocation allocates one good to the first agent and three goods to the second agent. Clearly, 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 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.

References

1 Nima Anari, Shayan Oveis Gharan, Amin Saberi, and Mohit Singh. Nash Social Welfare, Matrix Permanent, and Stable Polynomials. In *ITCS*, pages 36:1–36:12, 2017. doi:10.4230/LIPIcs.ITCS.2017.36.

2 Nima Anari, Tung Mai, Shayan Oveis Gharan, and Vijay V. Vazirani. Nash Social Welfare for Indivisible Items under Separable, Piecewise-Linear Concave Utilities. In *SODA*, pages 2274–2290, 2018. doi:10.1137/1.9781611975031.147.

3 Siddharth Barman, Sanath Kumar Krishna Murthy, and Rohit Vaish. Finding Fair and Efficient Allocations. *CoRR*, abs/1707.04731, 2017. to appear in EC 2018. arXiv:1707.04731.

4 Ioannis Caragiannis, David Kurokawa, Hervé Moulin, Ariel D. Procaccia, Nisarg Shah, and Junxing Wang. The Unreasonable Fairness of Maximum Nash Welfare. In *EC*, pages 305–322, 2016. doi:10.1145/2940716.2940726.

5 Richard Cole, Nikhil R. Devanur, Vasilis Gkatzelis, Kamal Jain, Tung Mai, Vijay V. Vazirani, and Sadra Yazdanbod. Convex Program Duality, Fisher Markets, and Nash Social Welfare. In *EC*, pages 459–460, 2017. doi:10.1145/3033274.3085109.

6 Richard Cole and Vasilis Gkatzelis. Approximating the Nash Social Welfare with Indivisible Items. In *STOC*, pages 371–380, 2015. doi:10.1145/2746539.2746589.

7 J. Edmonds and R.M. Karp. Theoretical Improvements in algorithmic efficiency for network flow problems. *J. ACM*, 19:248–264, 1972.

8 E. Eisenberg and D. Gale. Consensus of subjective probabilities: The Pari-Mutuel method. *The Annals Math. Statist.*, 30:165–168, 1959.

9 Jugal Garg, Martin Hoefer, and Kurt Mehlhorn. Approximating the Nash Social Welfare with Budget-Additive Valuations. In *SODA 2018*, pages 2326–2340, 2018. doi:10.1137/1.9781611975031.150.

10 Euiwoong Lee. APX-hardness of maximizing Nash social welfare with indivisible items. *Inf. Process. Lett.*, 122:17–20, 2017. doi:10.1016/j.ipl.2017.01.012.

11 R.M. McConnell, K. Mehlhorn, S. Näher, and P. Schweitzer. Certifying algorithms. *Computer Science Review*, 5(2):119–161, 2011. doi:10.1016/j.cosrev.2010.09.009.

12 Hervé Moulin. *Fair division and collective welfare*. MIT Press, 2003.

13 J. Nash. The Bargaining Problem. *Econometrica*, 18:155–162, 1950.

14 Nhan-Tam Nguyen, Trung Thanh Nguyen, Magnus Roos, and Jörg Rothe. Computational complexity and approximability of social welfare optimization in multiagent resource allocation. *Autonomous Agents and Multi-Agent Systems*, 28(2):256–289, 2014. doi:10.1007/s10458-013-9224-2.