

# Tight Approximation Algorithms for $p$ -Mean Welfare Under Subadditive Valuations

**Siddharth Barman**

Indian Institute of Science, Bangalore, India  
barman@iisc.ac.in

**Umang Bhaskar**

Tata Institute of Fundamental Research, Mumbai, India  
umang@tifr.res.in

**Anand Krishna**

Indian Institute of Science, Bangalore, India  
anandkrishna@iisc.ac.in

**Ranjani G. Sundaram**

Chennai Mathematical Institute, India  
ranjanigs@cmi.ac.in

---

## Abstract

---

We develop polynomial-time algorithms for the fair and efficient allocation of indivisible goods among  $n$  agents that have subadditive valuations over the goods. We first consider the Nash social welfare as our objective and design a polynomial-time algorithm that, in the value oracle model, finds an  $8n$ -approximation to the Nash optimal allocation. Subadditive valuations include XOS (fractionally subadditive) and submodular valuations as special cases. Our result, even for the special case of submodular valuations, improves upon the previously best known  $O(n \log n)$ -approximation ratio of Garg et al. (2020).

More generally, we study maximization of  $p$ -mean welfare. The  $p$ -mean welfare is parameterized by an exponent term  $p \in (-\infty, 1]$  and encompasses a range of welfare functions, such as social welfare ( $p = 1$ ), Nash social welfare ( $p \rightarrow 0$ ), and egalitarian welfare ( $p \rightarrow -\infty$ ). We give an algorithm that, for subadditive valuations and any given  $p \in (-\infty, 1]$ , computes (in the value oracle model and in polynomial time) an allocation with  $p$ -mean welfare at least  $1/s_n$  times the optimal.

Further, we show that our approximation guarantees are essentially tight for XOS and, hence, subadditive valuations. We adapt a result of Dobzinski et al. (2010) to show that, under XOS valuations, an  $O(n^{1-\varepsilon})$  approximation for the  $p$ -mean welfare for any  $p \in (-\infty, 1]$  (including the Nash social welfare) requires exponentially many value queries; here,  $\varepsilon > 0$  is any fixed constant.

**2012 ACM Subject Classification** Theory of computation  $\rightarrow$  Algorithmic game theory

**Keywords and phrases** Discrete Fair Division, Nash Social Welfare, Subadditive Valuations, Submodular Valuations

**Digital Object Identifier** 10.4230/LIPIcs.ESA.2020.11

**Related Version** A full version of the paper is available at <https://arxiv.org/abs/2005.07370>.

**Funding** *Siddharth Barman*: Supported by a Ramanujan Fellowship (SERB – SB/S2/RJN128/2015) and a Pratiksha Trust Young Investigator Award.

*Umang Bhaskar*: Supported by the Department of Atomic Energy, Government of India (project RTI4001), a Ramanujan Fellowship (SERB – SB/S2/RJN-055/2015), and an Early Career Research Award (SERB – ECR/2018/002766).



© Siddharth Barman, Umang Bhaskar, Anand Krishna, and Ranjani G. Sundaram;  
licensed under Creative Commons License CC-BY

28th Annual European Symposium on Algorithms (ESA 2020).

Editors: Fabrizio Grandoni, Grzegorz Herman, and Peter Sanders; Article No. 11; pp. 11:1–11:17

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

## 1 Introduction

In discrete fair division, given a set of  $m$  goods and  $n$  agents, the problem is to integrally allocate the set of goods to the agents in a fair and (economically) efficient manner [8, 17, 3]. In this thread of work, the Nash social welfare – defined as the geometric mean of the agents’ valuations for their assigned bundles – has emerged as a fundamental and prominent measure of the quality of an allocation. It provides a balance between two central objectives: the social welfare (the sum of the agents’ valuations) and the egalitarian welfare (the minimum valuation across the agents). Note that social welfare is a standard measure of (economic) efficiency, whereas egalitarian welfare is a fairness objective.

A Nash optimal allocation (i.e., an allocation that maximizes Nash social welfare) satisfies other fairness and efficiency criteria as well. Such an allocation is clearly Pareto optimal. Furthermore, if agents have additive valuations, then a Nash optimal allocation is known to be fair in the sense that it is guaranteed to be *envy-free up to one good* (EF1) [9] and *proportional up to one good* (PROP1) [15].<sup>1</sup>

As an objective, Nash social welfare is scale invariant: multiplicatively scaling any agent’s valuation function by a nonnegative factor does not change the Nash optimal allocation. Furthermore, interesting connections have been established between market models and this welfare function; see, e.g., [13, 5]. As a practical application, the website [spliddit.org](http://spliddit.org) uses the Nash social welfare as the optimization objective when partitioning indivisible goods [22, 9].

However, computing a Nash optimal allocation is APX-hard, even when the agents have additive valuations [24]. In terms of approximation algorithms, the problem of maximizing Nash social welfare has received considerable attention in recent years [14, 13, 2, 6, 1, 5, 19]. In particular, a polynomial-time  $e^{1/e}$ -approximation algorithm is known for additive valuations [5]. This algorithm preserves EF1, up to a factor of  $(1 + \varepsilon)$ , and Pareto optimality. The approximation guarantee of  $e^{1/e} \approx 1.45$  also holds for budget-additive valuations [11]. The work of Garg et al. [20] extends this line of work by considering Nash social welfare maximization under submodular valuations.

Submodular valuations capture the diminishing marginal returns property. They constitute a subclass of subadditive valuations, which, in turn, model complement-freeness. Formally, a set function  $v$  (defined over a set of indivisible goods) is subadditive if it satisfies  $v(A \cup B) \leq v(A) + v(B)$ , for all subsets of goods  $A$  and  $B$ . Complement-freeness is a very common assumption on valuation functions. Hence, fair division with subadditive valuations is an encompassing and important problem.

For Nash social welfare maximization under submodular valuations, Garg et al. [20] obtain an  $O(n \log n)$ -approximation algorithm. Prior to their work, the best known approximation ratio for submodular valuations was  $(m - n + 1)$ , which also extends to subadditive valuations [26]; here,  $m$  denotes the number of goods and  $n$  the number of agents. For a constant number of agents with submodular valuations, Garg et al. [20] provide an  $e/(e - 1)$ -approximation algorithm and show that, even in this setting, improving upon  $e/(e - 1)$  is NP-hard.

---

<sup>1</sup> An allocation is said to be EF1 if for any pair of agents  $i$  and  $j$ , there exists a good  $g$  in  $j$ ’s bundle, such that  $i$  prefers her bundle to the one obtained after removing  $g$  from  $j$ ’s bundle. An allocation is said to be PROP1 if for each agent  $i$  there exists a good  $g$  with the property that including  $g$  into  $i$ ’s bundle ensures that  $i$  achieves a proportional share, i.e., her valuation ends up being at least  $1/n$  times her value for all the goods.

In the context of allocating indivisible goods, two other well-studied welfare objectives are the social welfare and the egalitarian welfare. These represent, respectively, for an allocation, the average valuation of the agents and the minimum valuation of any agent. For the social welfare objective, a tight approximation factor of  $e/(e-1)$  is known under submodular valuations [29]. For subadditive valuations, Feige [18] shows that social welfare maximization admits a polynomial-time 2-approximation, assuming oracle access to *demand queries*.<sup>2</sup>

For maximizing egalitarian welfare under additive valuations, Chakrabarty et al. [10] provide an  $\tilde{O}(n^\varepsilon)$ -approximation algorithm that runs in time  $n^{O(1/\varepsilon)}$ , for any  $\varepsilon > 0$ . Under submodular valuations, egalitarian welfare maximization admits an  $\tilde{O}(n^{1/4}m^{1/2})$ -approximation algorithm [21]. Khot and Ponnuswami [23] provide a  $2n$ -approximation algorithm for maximizing egalitarian welfare under subadditive valuations. As a lower bound, with submodular valuations, egalitarian welfare cannot be approximated within a factor of 2, unless  $P = NP$  [7].

In this work we develop a unified treatment of fairness and efficiency objectives, including the welfare functions mentioned above. In particular, we develop an approximation algorithm for computing allocations that maximize the *generalized mean* of the agents' valuations. Formally, for exponent parameter  $p \in \mathbb{R}$ , the  $p^{\text{th}}$  *generalized mean* of a set of  $n$  positive reals  $v_1, v_2, \dots, v_n$  is defined as  $(\frac{1}{n} \sum_{i=1}^n v_i^p)^{1/p}$ . For an allocation (partition)  $\mathcal{A} = (A_1, \dots, A_n)$  of the indivisible goods among the  $n$  agents, we define the  $p$ -*mean welfare* of  $\mathcal{A}$  as the generalized mean of the values  $(v_i(A_i))_{i \in [n]}$ ; here  $v_i(A_i)$  is the value that agent  $i$  has for the bundle  $A_i$  assigned to it. Indeed, with different values of  $p$ , the  $p$ -mean welfare encompasses a range of objectives: it corresponds to the social welfare (arithmetic mean) for  $p = 1$ , the Nash social welfare (geometric mean) for  $p \rightarrow 0$ , and the egalitarian welfare for  $p \rightarrow -\infty$ . In fact,  $p$ -mean welfare functions with  $p \in (-\infty, 1]$  exactly correspond to the collection of functions characterized by a set of natural axioms, including the Pigou-Dalton transfer principle [25]. Hence,  $p$ -mean welfare functions, with  $p \in (-\infty, 1]$ , constitute an important and axiomatically-supported family of objectives.

**Our Contributions.** We develop a polynomial-time algorithm that, given a fair division instance with subadditive valuations and parameter  $p \in (-\infty, 1]$ , finds an allocation with  $p$ -mean welfare at least  $1/8n$  times the optimal  $p$ -mean welfare (Theorem 9). Our algorithm uses the standard value oracle model which, when queried with any subset of goods and an agent  $i$ , returns the value that  $i$  has for the subset. For different values of  $p$ , our algorithm changes minimally, differing only in the weights of edges for a computed matching. We thus present a unified analysis for this broad class of welfare functions, suggesting further connections between these objectives than the previously mentioned axiomatization. Our result matches the best known  $O(n)$ -approximation for egalitarian welfare [23] and improves upon the  $O(n \log n)$ -approximation guarantee of Garg et al. [20] for Nash social welfare with submodular valuations. Arguably, our algorithm (and the analysis) is simpler than the one developed in [20] and simultaneously more robust, since it obtains an improved approximation ratio for subadditive valuations and a notably broader class of welfare objectives.

For clarity of exposition, we first present an  $8n$ -approximation algorithm for maximizing Nash social welfare under subadditive valuations (Theorem 2). We then generalize the algorithm to the class of  $p$ -mean welfare objectives.

<sup>2</sup> A demand-query oracle, when queried with prices  $p_1, \dots, p_m \in \mathbb{R}$  associated with the  $m$  goods, returns  $\max_{S \subseteq [m]} (v(S) - \sum_{j \in S} p_j)$ , for an underlying valuation function  $v$ . The current paper works with more basic value oracle, which when queried with a subset of goods returns the value this subset. Any value query can be simulated via a polynomial number of demand queries. However, the converse is not true [27].

We complement these algorithmic results by adapting a result of Dobzinski et al. [16] to show that for XOS valuations, any  $O(n^{1-\varepsilon})$ -approximation for  $p$ -mean welfare requires an exponential number of value queries (Section 5). Hence, in the value oracle model, our approximation guarantee is essentially tight for XOS and, hence, for subadditive valuations. We note that these are the first polynomial lower bounds on approximating either the Nash social welfare or the egalitarian welfare.

Nguyen and Rothe [26] obtain an  $(m - n + 1)$ -approximation guarantee for maximizing Nash social welfare with subadditive valuations. We establish two extensions of this result. First, we show that, under subadditive valuations, an  $(m - n + 1)$ -approximation for the  $p$ -mean welfare can be obtained for all  $p \leq 0$ . However, for  $0 < p < 1$ , we establish that it is NP-hard to obtain an  $(m - n + 1)$ -approximation, even under additive valuations. An analogous hardness result holds for  $p = 1$  with submodular valuations.

**Independent Work.** In work independent of ours, Chaudhury et al. [12] also obtain an  $O(n)$ -approximation algorithm for maximizing generalized  $p$ -means under subadditive valuations. Their approach varies significantly from the current paper and, in particular, builds upon results on finding allocations that are approximately envy-free up to any good (EFX). Notably their algorithm computes allocations that satisfy additional fairness properties, including EF1 and either of two approximate versions of EFX.

Section 3 presents our approximation algorithm for maximizing Nash social welfare. Then, Section 4 shows that we can extend the algorithm for Nash social welfare to obtain the stated approximation bound for  $p$ -mean welfare. The tightness of these results is established in Section 5. Section 6 presents the results for the  $(m - n + 1)$ -approximation guarantees. All the missing proofs appear in the full version of this paper [4].

## 2 Notation and Preliminaries

An instance of a fair division problem is a tuple  $\langle [m], [n], \{v_i\}_{i=1}^n \rangle$ , where  $[m] = \{1, 2, \dots, m\}$  denotes the set of  $m \in \mathbb{N}$  indivisible goods that have to be allocated (partitioned) among the set of  $n \in \mathbb{N}$  agents,  $[n] = \{1, 2, \dots, n\}$ . Here,  $v_i : 2^{[m]} \mapsto \mathbb{R}_+$  represents the valuation function of agent  $i \in [n]$ . Specifically,  $v_i(S) \in \mathbb{R}_+$  is the value that agent  $i$  has for a subset of goods  $S \subseteq [m]$ . For  $g \in [m]$  and  $i \in [n]$ , write  $v_i(g)$  to denote agent  $i$ 's value for the good  $g$ , i.e., it denotes  $v_i(\{g\})$ .

We will assume throughout that the valuation function  $v_i$  for each agent  $i \in [n]$  is (i) nonnegative:  $v_i(S) \geq 0$  for all  $S \subseteq [m]$ , (ii) normalized:  $v_i(\emptyset) = 0$ , (iii) monotone:  $v_i(A) \leq v_i(B)$  for all  $A \subseteq B \subseteq [m]$ , and (iv) subadditive:  $v_i(A \cup B) \leq v_i(A) + v_i(B)$  for all subsets  $A, B \subseteq [m]$ .

Submodular and XOS (fractionally subadditive) valuations constitute subclasses of subadditive valuations. Formally, a set function  $v : 2^{[m]} \mapsto \mathbb{R}_+$  is said to be submodular if it satisfies the *diminishing marginal returns* property:  $v(A \cup \{g\}) - v(A) \geq v(B \cup \{g\}) - v(B)$ , for all subsets  $A \subseteq B \subsetneq [m]$  and  $g \in [m] \setminus B$ . A set function,  $v : 2^{[m]} \mapsto \mathbb{R}_+$ , is said to be XOS if it is obtained by evaluating the maximum over a collection of additive functions  $\{f_r\}_{r \in [L]}$ , i.e.,  $v(S) := \max_{1 \leq j \leq L} \{f_r(S)\}$ , for each subset  $S \subseteq [m]$ .<sup>3</sup>

We use  $\Pi_n([m])$  to denote the collection of all  $n$  partitions of the indivisible goods  $[m]$ . An *allocation* is an  $n$ -partition  $\mathcal{A} = (A_1, \dots, A_n) \in \Pi_n([m])$  of the  $m$  goods. Here,  $A_i$  denotes the subset of goods allocated to agent  $i \in [n]$  and will be referred to as a *bundle*.

<sup>3</sup> Here,  $L$  can be exponentially large in  $m$ .

Given a fair division instance  $\mathcal{I} = \langle [m], [n], \{v_i\}_i \rangle$ , the *Nash social welfare* of allocation  $\mathcal{A}$  is defined as the geometric mean of the agents' valuations under  $\mathcal{A}$ :  $\text{NSW}(\mathcal{A}) := (\prod_{i=1}^n v_i(A_i))^{\frac{1}{n}}$ .

We will throughout use  $\mathcal{N}^* = (N_1^*, \dots, N_n^*)$  to denote an allocation that maximizes the Nash social welfare for a given fair division instance. We refer to  $\mathcal{N}^*$  as a Nash optimal allocation. An allocation  $\mathcal{P} = (P_1, \dots, P_n)$  is an  $\alpha$ -approximate solution (with  $\alpha \geq 1$ ) of the Nash social welfare maximization problem if  $\text{NSW}(\mathcal{P}) \geq \frac{1}{\alpha} \text{NSW}(\mathcal{N}^*)$ .

Besides the Nash social welfare, we address a family of objectives defined by considering the *generalized means* of agents' valuations. In particular, for parameter  $p \in \mathbb{R}$ , the  $p^{\text{th}}$  generalized (Hölder) mean  $M_p(\cdot)$  of  $n$  nonnegative numbers  $x_1, \dots, x_n \in \mathbb{R}_+$  is defined as

$$M_p(x_1, \dots, x_n) := \left( \frac{1}{n} \sum_{i=1}^n x_i^p \right)^{\frac{1}{p}}.$$

Parameterized by  $p$ , this family of functions captures multiple fairness and efficiency measures. In particular, when  $p = 1$ ,  $M_p$  reduces to the arithmetic mean. In the limit,  $M_p$  is equal to the geometric mean as  $p$  tends to zero. In addition,  $\lim_{p \rightarrow -\infty} M_p(x_1, \dots, x_n) = \min\{x_1, x_2, \dots, x_n\}$ .

We define the *p-mean welfare*,  $M_p(\mathcal{A})$ , of an allocation  $\mathcal{A} = (A_1, A_2, \dots, A_n)$  as

$$M_p(\mathcal{A}) := M_p(v_1(A_1), \dots, v_n(A_n)) = \left( \frac{1}{n} \sum_{i=1}^n v_i(A_i)^p \right)^{1/p}.$$

With  $p$  equal to one, zero, and  $-\infty$ , the  $p$ -mean welfare corresponds to the (average) social welfare, Nash social welfare, and egalitarian welfare, respectively.

The following proposition implies that for any  $p \leq -n \log n$ , if instead of the  $p$ -mean welfare, we maximize the egalitarian welfare, then the resulting allocation loses a negligible factor in the approximation ratio.

► **Proposition 1.** *For any  $n$  nonnegative numbers  $x_1, \dots, x_n \in \mathbb{R}_+$  and  $p \leq -n \log n$ , we have*

$$M_{-\infty}(x_1, \dots, x_n) \leq M_p(x_1, \dots, x_n) \leq 2^{1/n} M_{-\infty}(x_1, \dots, x_n).$$

### 3 An $8n$ -Approximation for Nash Social Welfare

This section presents an efficient  $8n$ -approximation algorithm for the Nash social welfare maximization problem, under subadditive valuations. Our algorithm, Algorithm 1 (ALG), requires access to the valuation functions through basic value queries, i.e., it only requires an oracle which, when queried with a subset of goods  $S \subseteq [m]$  and an agent  $i \in [n]$ , returns  $v_i(S) \in \mathbb{R}_+$ .

We first describe the ideas behind our algorithm. Write  $\mathcal{N}^* = \{N_1^*, \dots, N_n^*\}$  denote a Nash optimal allocation in the given instance and let us, for now, assume that the agents have additive valuations, i.e., for all agents  $i \in [n]$  and subset of goods  $S \subseteq [m]$ , we have  $v_i(S) = \sum_{g \in S} v_i(g)$ . In the following two cases, we can readily obtain an  $O(n)$  approximation. In the first case, each agent has a few “high-value” goods, i.e., each agent  $i$  has a good  $g'_i \in N_i^*$  with the property that  $v_i(g'_i) \geq v_i(N_i^*)/n$ . In such a setting, we can construct a complete bipartite graph with agents  $[n]$  on one side and all the goods  $[m]$  on the other. Here, the weight of edge  $(i, g) \in [n] \times [m]$  is set to be  $\log(v_i(g))$ . In this bipartite graph, the matching  $(i, g'_i)_{i \in [n]}$  has Nash social welfare at least  $1/n$  times the optimal and, hence, this also holds for a left-perfect maximum-weight matching in this graph.

**Algorithm 1** ALG.

---

**Input:** Instance  $\mathcal{I} = \langle [m], [n], \{v_i\}_{i=1}^n \rangle$  with value oracle access to the valuation functions  $v_i$ s.

**Output:** An allocation  $\mathcal{B} = (B_1, B_2, \dots, B_n)$

- 1: Initialize iteration count  $t = 0$  and define  $\text{SAT}_t = \emptyset$  and  $\text{UNSAT}_t = [n]$
- 2: **for**  $i \in [n]$  **do**
- 3:     Sort the goods in  $[m]$  in descending order of value such that  $v_i(g_1) \geq \dots \geq v_i(g_m)$
- 4:     **if**  $v_i([m] \setminus \{g_1, \dots, g_{2n}\}) = 0$  **then**
- 5:         Set  $\gamma_i^t = 0$
- 6:     **else**
- 7:          $\gamma_i^t = v_i([m])$
- 8:     **end if**
- 9: **end for**
- 10: **while**  $\text{UNSAT}_t \neq \emptyset$  **do**
- 11:     Consider the bipartite graph  $([n] \cup [m], [n] \times [m], \{w(i, g)\}_{i \in [n], g \in [m]})$  with weight of edge  $(i, g) \in [n] \times [m]$  set as  $w(i, g) = \log(v_i(g) + \gamma_i^t)$
- 12:     Compute a left-perfect maximum-weight matching,  $\pi^t$ , in this bipartite graph
- 13:     Set  $G = [m] \setminus \{\pi^t(i)\}_{i \in [n]}$  and  $A = [n]$
- 14:     **while** there exists  $a' \in A$  and  $g' \in G$  such that  $v_{a'}(g') \geq \frac{1}{2n} v_{a'}(G)$  **do**
- 15:         Set  $B_{a'}^t = \{g'\}$  and update  $G \leftarrow G \setminus \{g'\}$  along with  $A \leftarrow A \setminus \{a'\}$
- 16:     **end while**
- 17:     Set  $(B_i^t)_{i \in A} = \text{MOVINGKNIFE}(G, A, \{v_i\}_{i \in A})$
- 18:     Define  $\text{SAT}_{t+1} = \{i \in [n] \mid v_i(B_i^t) \geq \gamma_i^t\}$  and set  $\gamma_i^{t+1} = \gamma_i^t$  for each  $i \in \text{SAT}_{t+1}$
- 19:     Define  $\text{UNSAT}_{t+1} = \{i \in [n] \mid v_i(B_i^t) < \gamma_i^t\}$  and set  $\gamma_i^{t+1} = (1 - \frac{1}{m}) \gamma_i^t$  for each  $i \in \text{UNSAT}_{t+1}$
- 20:     Update  $t \leftarrow t + 1$
- 21: **end while**
- 22: **return** allocation  $(B_1^{t-1} \cup \{\pi^{t-1}(1)\}, B_2^{t-1} \cup \{\pi^{t-1}(2)\}, \dots, B_n^{t-1} \cup \{\pi^{t-1}(n)\})$

---

In the second case, all goods are of “low-value”, i.e., for all  $i \in [n]$  and  $g \in [m]$  we have  $v_i(g) \leq v_i(N_i^*)/(2n)$ . Here again an  $O(n)$  approximation can be obtained via a simple round-robin algorithm, wherein the agents (in an arbitrary order) repeatedly pick their highest valued good from those remaining. At a high level, our algorithm stitches together these two extreme cases by first matching high-value goods and then allocating the low-value ones.

We connect the two cases by considering the following quantity for each agent  $i \in [n]$

$$\ell_i := \min_{S \subseteq [m]: |S| \leq 2n} \frac{1}{2n} v_i([m] \setminus S). \quad (1)$$

That is,  $\ell_i$  is the value that each agent is guaranteed to achieve in a (near) proportional allocation, even after the removal of any  $2n$ -size subset of goods. Our algorithm leverages the following existential guarantee (Lemma 3): there necessarily exists a good  $\hat{g}_i \in N_i^*$  with the property that

$$v_i(\hat{g}_i) + \ell_i \geq \frac{1}{4n} v_i(N_i^*). \quad (2)$$

This result ensures that, a single high-value good (in particular,  $\hat{g}_i$ ) coupled with a  $2n$ -approximation to all the low-value goods (i.e.,  $\ell_i$ ), is sufficient to ensure a  $4n$ -approximation for each agent. At this point, if we could (i) explicitly compute  $\ell_i$  for each agent  $i$  and (ii)

for any size- $n$  subset of goods  $S$ , assign the remaining goods  $[m] \setminus S$  such that each agent gets a bundle of value at least  $\ell_i$ , then we would be done. This follows from the observation that in the complete bipartite graph  $([n] \cup [m], [n] \times [m])$  with weight of edge  $(i, g)$  set to  $\log(v_i(g) + \ell_i)$ , the weight of the matching  $(i, \hat{g}_i)_i$  is a  $4n$  approximation to the optimal Nash social welfare by equation (2) and, hence, the same guarantee holds for a maximum-weight matching in the graph. Condition (ii) ensures that each agent also receives at least  $\ell_i$  after the initial assignment of the  $n$  matched goods.

For additive valuations, both conditions (i) and (ii) can be satisfied. This template was employed in the SMatch algorithm (for additive valuations) of Garg et al. [20]. However, for submodular (and subadditive) valuations, the quantity  $\ell_i$  is hard to approximate within a sub-linear factor [28].

Therefore, instead of satisfying condition (i) explicitly, we maintain an upper bound  $\gamma_i \geq \ell_i$  for each agent  $i$ . Our algorithm first obtains a maximum weight matching in the bipartite graph between agents and goods with the weight of edge  $(i, g) \in [n] \times [m]$  set to  $\log(v_i(g) + \gamma_i)$ . It assigns all the matched goods to the respective agents, removes these goods from further consideration in this iteration, and then carries out a procedure (described below) to ensure condition (ii). If, for agent  $i$ , the bundle obtained in this procedure (i.e., the bundle obtained for  $i$  after removing the matched goods) has value less than  $\gamma_i$ , then we multiplicatively reduce the (over) estimate  $\gamma_i$  for  $i$  and repeat the algorithm.

The procedure towards satisfying condition (ii) consists of two steps. Let  $G$  be the set of goods that remain once we remove the matched  $n$  goods from  $[m]$ . In the first step, if there exists an agent  $i$  and a good  $g \in G$  such that  $v_i(g) \geq v_i(G)/(2n)$ , we assign  $g$  to  $i$  and remove both from further consideration. An agent thus removed has value  $\ell_i$  from the assigned good; note that, by definition,  $\ell_i \leq v_i(G)/(2n)$ . After this step, we observe that  $v_i(g) \leq v_i(G)/(2n)$  for each remaining agent  $i$  and good  $g$ . In the second step, we run a *moving knife* subroutine (Algorithm 2) on the goods that are still unassigned. In this subroutine, the goods are initially ordered in an arbitrary fashion. A hypothetical knife is then moved across the goods from one side until an agent  $i$  (who has yet to receive a bundle) calls out that the goods covered so far have a collective value of at least  $v_i(G)/(2n)$  for her. These covered goods are then allocated to said agent  $i$  and both the agent as well as this bundle is removed from further consideration. We show that this allocation satisfies condition (ii), i.e., the bundle assigned to each agent in this procedure has value at least  $\ell_i$  (but it may be lower than the overestimate  $\gamma_i$ ).

Since we can guarantee  $\ell_i$  for each agent  $i$ , irrespective of which goods are removed in the matching step,  $\gamma_i$  never goes below  $\ell_i$ , for any agent. Hence, at some point, every agent  $i$  receives a bundle of value at least  $\gamma_i$  in the above two steps. We show that these bundles, with the goods matched with each agent, provide an  $8n$  approximation to the optimal Nash social welfare.

It is relevant to note we use  $\ell_i$  solely for the purposes of analysis. Our algorithm executes with the overestimate  $\gamma_i$  and keeps reducing this value till it is realized (in the two-step procedure) for all the agents.

As mentioned previously, the SMatch algorithm (developed for additive valuations) of Garg et al. [20] relies of conditions (i) and (ii). However, for submodular valuations their work diverges considerably from the current approach. In particular, the RepReMatch algorithm (developed for submodular valuations) in [20] first finds a set of goods  $\mathcal{G}$  with the property that in the bipartite graph between all the agents and  $\mathcal{G}$ , there is a matching wherein every agent is matched to a good with value at least as much as her highest valued good in  $N_i^*$ . To ensure this property the cardinality of  $\mathcal{G}$  needs to be  $n \log n$ . Intuitively,

■ **Algorithm 2** MOVINGKNIFE.

**Input:** Instance  $\mathcal{J} = \langle G, A, \{v_i\}_{i \in A} \rangle$  with value oracle access to the valuation functions  $v_i$

**Output:** An allocation  $\mathcal{P} = (P_1, P_2, \dots, P_{|A|})$

---

```

1: Initialize  $S = \emptyset$ ,  $\widehat{G} = G$ ,  $\widehat{A} = A$ , and bundle  $P_i = \emptyset$  for all  $i \in A$ .
2: while  $\widehat{G} \neq \emptyset$  and  $\widehat{A} \neq \emptyset$  do
3:   Select any arbitrary good  $g \in \widehat{G}$  and update  $S \leftarrow S \cup \{g\}$  along with  $\widehat{G} \leftarrow \widehat{G} \setminus \{g\}$ .
4:   if for some agent  $\widehat{a} \in \widehat{A}$  we have  $v_{\widehat{a}}(S) \geq \frac{1}{2n} v_{\widehat{a}}(\widehat{G})$  then
5:     Set  $P_{\widehat{a}} = S$  and update  $\widehat{A} \leftarrow \widehat{A} \setminus \{\widehat{a}\}$  along with  $S \leftarrow \emptyset$ .
6:   end if
7: end while
8: if  $\widehat{G} \neq \emptyset$  then
9:    $P_{|A|} \leftarrow P_{|A|} \cup \widehat{G}$ 
10: end if
11: return allocation  $\mathcal{P} = (P_1, \dots, P_{|A|})$ .

```

---

this requirement leads to a lower bound of  $\Omega(n \log n)$  on the approximation ratio obtained in [20]. Furthermore, the steps in their algorithm to ensure condition (ii) do not extend to subadditive valuations either. Specifically, Garg et al. [20] note that their algorithm gives an approximation ratio of  $\Omega(m)$  for the case of subadditive valuations. The  $2n$ -approximation of Khot and Ponnuswami for egalitarian welfare [23] first guesses the optimal egalitarian welfare  $b$ , and uses this to partition the goods into “large” ones (those with value higher than  $b/n$ ) and “small” ones, for each agent. It then tries to ensure every agent receives a bundle with valuation at least  $b/n$ . For Nash social welfare, guessing just a single value does not appear to help, since the Nash social welfare depends on the valuation of each agent.

The following theorem constitutes our main result for Nash social welfare.

► **Theorem 2.** *Let  $\mathcal{I} = \langle [m], [n], \{v_i\}_{i=1}^n \rangle$  be a fair division instance in which the valuation function  $v_i$ , of each agent  $i \in [n]$ , is nonnegative, monotone, and subadditive. Given value oracle access to  $v_i$ s, the algorithm ALG computes an  $8n$  approximation to the Nash optimal allocation in polynomial time.*

The following lemma proves inequality (2). We state and prove it for an arbitrary allocation  $\mathcal{A}^* = (A_1^*, \dots, A_n^*)$ , rather than just for the Nash optimal allocation.

► **Lemma 3.** *Let  $\mathcal{I} = \langle [m], [n], \{v_i\}_{i=1}^n \rangle$  be a fair division instance with monotone, subadditive valuations and let  $\mathcal{A}^* = (A_1^*, \dots, A_n^*)$  be any allocation in  $\mathcal{I}$ . Let  $\widehat{g}_i$  be the most valued (by  $i$ ) good in  $A_i^*$  (i.e.,  $\widehat{g}_i := \arg \max_{g \in A_i^*} v_i(g)$ ) and  $\ell_i$  be as defined in (1). Then, for each agent  $i \in [n]$*

$$v_i(\widehat{g}_i) + \ell_i \geq \frac{1}{4n} v_i(A_i^*).$$

**Proof.** Consider any agent  $i \in [n]$  and note that  $\ell_i \geq 0$ . We will establish the lemma by considering two complementary cases.

**Case 1.** There exists a good  $g_i \in A_i^*$  with the property that  $v_i(g_i) \geq \frac{1}{4n} v_i(A_i^*)$ . Since  $\widehat{g}_i$  is the most valued good in  $A_i^*$ , we have  $v_i(\widehat{g}_i) \geq v_i(g_i)$  and the desired inequality follows.



**Case 2.** For all goods  $g \in A_i^*$ ,  $v_i(g) < \frac{1}{4n}v_i(A_i^*)$ . Recall that  $\ell_i := \min_{S \subseteq [m], |S| \leq 2n} \frac{1}{2n}v_i([m] \setminus S)$ . Let  $S^*$  be the set  $S$  that induces  $\ell_i$ , i.e.,  $\ell_i = \frac{1}{2n}v_i([m] \setminus S^*)$ . Monotonicity of  $v_i$  ensures that  $|S^*| = 2n$  and

$$\ell_i = \frac{1}{2n}v_i([m] \setminus S^*) \geq \frac{1}{2n}v_i(A_i^* \setminus S^*). \quad (3)$$

Furthermore, given that in the current case  $v_i(g) < \frac{1}{4n}v_i(A_i^*)$  for all  $g \in A_i^*$ , we have

$$v_i(A_i^* \cap S^*) \leq \sum_{g \in A_i^* \cap S^*} v_i(g) < \sum_{g \in A_i^* \cap S^*} \frac{1}{4n}v_i(A_i^*) \leq \frac{|S^*|}{4n}v_i(A_i^*) = \frac{1}{2}v_i(A_i^*). \quad (4)$$

Here, the first inequality follows from the fact that  $v_i$  is subadditive and the last since  $|S^*| = 2n$ .

Therefore, we obtain the desired bound in terms of  $\ell_i$ :

$$\begin{aligned} \ell_i &\geq \frac{1}{2n}v_i(A_i^* \setminus S^*) && \text{(via inequality (3))} \\ &\geq \frac{1}{2n}(v_i(A_i^*) - v_i(A_i^* \cap S^*)) && \text{(} v_i \text{ is subadditive)} \\ &\geq \frac{1}{4n}v_i(A_i^*) && \text{(via inequality (4))} \end{aligned}$$

Thus, the the stated inequality  $v_i(\hat{g}_i) + \ell_i \geq \frac{1}{4n}v_i(A_i^*)$  holds even in this case.  $\blacktriangleleft$

The next lemma establishes the key property of Algorithm 2 (MOVINGKNIFE): if all the goods have low value for every agent, then MOVINGKNIFE returns a near-proportional allocation.

► **Lemma 4.** *Consider a fair division instance  $\langle G, A, \{v_i\}_{i \in A} \rangle$  wherein the agents have monotone, subadditive valuations. In addition, suppose for each agent  $i \in A$  and good  $g \in G$  we have  $v_i(g) < \frac{1}{2n}v_i(G)$ , where  $n \geq |A|$ . Then the allocation  $(P_1, \dots, P_{|A|})$  returned by Algorithm 2 (MOVINGKNIFE) satisfies  $v_i(P_i) \geq \frac{1}{2n}v_i(G)$  for all  $i \in A$ .*

**Proof.** Given instance  $\langle G, A, \{v_i\}_{i \in A} \rangle$ , the MOVINGKNIFE algorithm (Algorithm 2) considers the goods in an arbitrary order and adds these goods one by one into a bundle  $S$  until an agent  $\hat{a}$  calls out that its value for  $S$  is at least  $\frac{1}{2n}v_{\hat{a}}(G)$ . We assign these goods to agent  $\hat{a}$  and remove them – along with  $\hat{a}$  – from consideration. The algorithm iterates over the remaining set of agents and goods. We will show that the while loop in the MOVINGKNIFE algorithm terminates with  $\hat{A} = \emptyset$  and, hence, assigns to each agent a bundle of desired value.

Consider an integer (count)  $k \in \mathbb{N}$ . Let  $\hat{G}$  and  $\hat{A}$  denote the set of goods and agents, respectively, that are left unassigned after  $k$  agents are assigned bundles in MOVINGKNIFE; note that  $|\hat{A}| = |A| - k$ . The arguments below establish that for each remaining agent  $i \in \hat{A}$ ,

$$v_i(\hat{G}) \geq \left(1 - \frac{k}{n}\right)v_i(G) \quad (5)$$

Therefore, for any  $k < |A| \leq n$ , the set of unassigned goods  $\hat{G}$  is nonempty and even the last agent (i.e., with  $k = |A| - 1$ ) receives a bundle of sufficiently high value.

To prove (5), consider any agent  $i \in \hat{A}$ . Indeed, agent  $i$  has not received any goods yet, but the  $k$  agents in  $A \setminus \hat{A}$  have been assigned bundles. Let  $S$  be a bundle assigned to some agent in  $A \setminus \hat{A}$  (i.e.,  $P_j = S$  for some  $j \in A \setminus \hat{A}$ ) and  $g'$  be the last good included in  $S$ . Step 4 of the algorithm ensures that  $v_i(S \setminus \{g'\}) < \frac{1}{2n}v_i(G)$ ; otherwise,  $S \setminus \{g'\}$  would have

## 11:10 Tight Approximation Algorithms for $p$ -Mean Welfare Under Subadditive Valuations

been assigned to agent  $i$ . Furthermore, the assumption (in the Lemma statement) gives us  $v_i(g') \leq \frac{1}{2n}v_i(G)$ . Hence, using these inequalities and the subadditivity of  $v_i$ , we get  $v_i(S) \leq v_i(S \setminus \{g'\}) + v_i(g') \leq \frac{1}{n}v_i(G)$ .

This inequality provides an upper bound on  $v_i(G \setminus \widehat{G}) = v_i\left(\bigcup_{j \in A \setminus \widehat{A}} P_j\right)$ , the total value of the set of goods assigned among the  $k$  agents in  $A \setminus \widehat{A}$ . Specifically, by the subadditivity of  $v_i$ ,  $v_i(G \setminus \widehat{G}) \leq \frac{k}{n}v_i(G)$ . Therefore,  $v_i(\widehat{G}) \geq v_i(G) - v_i(G \setminus \widehat{G}) \geq \left(1 - \frac{k}{n}\right)v_i(G)$ .

Overall, every agent  $i \in A$  is eventually assigned a bundle of value at least  $\frac{1}{2n}v_i(G)$  in the while loop.  $\blacktriangleleft$

Next we show that in each iteration of the while loop in ALG (Algorithm 1), the value of the assigned bundle  $B_i^t$  is at least as large as  $\ell_i$ .

► **Lemma 5.** *Given a fair division instance  $\mathcal{I} = \langle [m], [n], \{v_i\}_{i=1}^n \rangle$  with subadditive valuations, let  $B_i^t$  be the bundle assigned to agent  $i \in [n]$  in the  $t^{\text{th}}$  iteration (for  $t \in \mathbb{N}$ ) of the outer while loop (Step 10) in ALG. Then, for all agents  $i \in [n]$  and each iteration count  $t$ , we have  $v_i(B_i^t) \geq \ell_i$ .*

**Proof.** During any iteration  $t$  of the outer while loop (Step 10) in ALG and for any agent  $i \in [n]$ , the bundle  $B_i^t$  either consists of a single good of high value (Step 15), or of the set of goods assigned to agent  $i$  obtained after executing the MOVINGKNIFE subroutine (Step 17). We will show that in both cases the stated inequality holds.

Recall that  $\ell_i := \min_{S \subseteq [m]: |S| \leq 2n} \frac{1}{2n} v_i([m] \setminus S)$ . Equivalently,  $\ell_i = \min_{T \subseteq [m]: |T| \geq m-2n} \frac{1}{2n} v_i(T)$ . Therefore, we have

$$\frac{1}{2n}v_i(T) \geq \ell_i \quad \text{for any subset } T \subseteq [m] \text{ of size at least } (m-2n) \quad (6)$$

The relevant observation here is that, in any iteration  $t$ , the set of goods  $G$  from which the bundles  $B_i^t$ s are populated satisfies  $|G| \geq m-2n$ . Specifically, in the  $t^{\text{th}}$  iteration, we start with  $|G| = m-n$  (Step 13). Subsequently, the inner while loop (Step 14) assigns at most  $n$  goods and, hence, the number of goods passed on to the MOVINGKNIFE subroutine satisfies  $|G| \geq m-2n$ .

First, we note that the lemma holds for any agent  $a'$  that receive a singleton bundle  $B_{a'}^t = \{g'\}$  in Step 15:  $v_{a'}(g') \geq \frac{1}{2n}v_{a'}(G) \geq \ell_{a'}$ . Here, the first inequality follows from the selection criterion applied to  $g'$  and the second inequality from equation (6) and the fact that  $|G| \geq m-2n$ .

Finally, we note that the bound also holds for the remaining agents  $i$  that receive a bundle  $B_i^t$  through the MOVINGKNIFE subroutine. As mentioned previously, at least  $m-2n$  goods are passed on as input to the subroutine, i.e., if MOVINGKNIFE is executed on instance  $\mathcal{J} = \langle G, A, \{v_i\}_{i \in A} \rangle$ , then we have  $|G| \geq m-2n$ . Inequality (6) ensures that  $\frac{1}{2n}v_i(G) \geq \ell_i$  for all  $i \in A$ . Finally, using Lemma 4, we get that the bundle assigned to agent  $i \in A$  satisfies the stated inequality:  $v_i(B_i^t) = v_i(P_i) \geq \frac{1}{2n}v_i(G) \geq \ell_i$ .

Hence, the stated claim follows.  $\blacktriangleleft$

We now show that the estimates  $\gamma_i^t$ s used in ALG also satisfy a lower bound similar to that in Lemma 5.

► **Lemma 6.** *Given a fair division instance  $\mathcal{I} = \langle [m], [n], \{v_i\}_{i=1}^n \rangle$  with subadditive valuations, let  $\gamma_i^t \in \mathbb{R}_+$  be the estimate associated with agent  $i \in [n]$  in the  $t^{\text{th}}$  iteration (for  $t \in \mathbb{N}$ ) of the outer while loop (Step 10) in ALG. Then, for all agents  $i \in [n]$  and each iteration count  $t$ , we have  $\gamma_i^t \geq \left(1 - \frac{1}{m}\right)\ell_i$ .*

**Proof.** Note that for any agent  $i \in [n]$ , the quantity  $\ell_i = 0$  iff  $i$  has positive value for at most  $2n$  goods. This observation implies that the initial for loop in ALG correctly identifies agents  $i$  that have  $\ell_i = 0$ , and sets  $\gamma_i^0 = 0$ . For such agents  $\gamma_i^t = 0$  for all  $t$ . Hence, the lemma holds for any agent  $i$  with  $\ell_i = 0$ .

We now consider agents  $i \in [n]$  with  $\ell_i > 0$ . For such an agent  $i$ , the algorithm initially sets  $\gamma_i^0 = v_i([m])$ . Hence, for  $t = 0$  we have  $\gamma_i^t \geq (1 - \frac{1}{m}) \ell_i$ . An inductive argument shows that this inequality continues to hold as the algorithm progresses. In particular, if in the  $t^{\text{th}}$  iteration the algorithm does not decrement the estimate (i.e., if  $i \in \text{SAT}_{t+1}$ ), then  $\gamma_i^{t+1} = \gamma_i^t \geq (1 - \frac{1}{m}) \ell_i$ .

Even otherwise, if the algorithm multiplicatively decrements the estimate (in particular, sets  $\gamma_i^{t+1} = (1 - 1/m) \gamma_i^t$ ), then it must be the case that  $\gamma_i^t > v_i(B_i^t)$  (i.e.,  $i \in \text{UNSAT}_{t+1}$ ). That is, after the decrement we have  $\gamma_i^{t+1} \geq (1 - \frac{1}{m}) v_i(B_i^t) \geq (1 - \frac{1}{m}) \ell_i$ ; the last inequality follows from Lemma 5. This completes the proof.  $\blacktriangleleft$

### 3.1 Proof of Theorem 2

In this section we prove Theorem 2 by showing that ALG runs in polynomial time (Lemma 7) and the computed allocation achieves the stated approximation ratio of  $8n$  (Lemma 8).

► **Lemma 7 (Runtime Analysis).** *Given any fair division instance  $\mathcal{I} = \langle [m], [n], \{v_i\}_{i=1}^n \rangle$  in which the agents have monotone, subadditive valuations, ALG (Algorithm 1) terminates after  $T = O(nm \log(nmV))$  iterations of its outer while loop (Step 10); here,*

$$V = \max_{i \in [n]} \left( \frac{\max_{g \in [m]} v_i(g)}{\min_{g \in [m]: v_i(g) > 0} v_i(g)} \right).$$

**Proof.** By design, ALG iterates as long as  $\text{UNSAT}_t \neq \emptyset$ . We will bound the number of times (i.e., the distinct values of  $t$  for which) any agent  $i \in [n]$  is contained in  $\text{UNSAT}_t$  and, hence, establish the stated runtime bound.

Recall that for any agent  $i \in [n]$ , the quantity  $\ell_i = 0$  iff  $i$  has positive value for at most  $2n$  goods. For such agents ALG sets  $\gamma_i^0 = 0$ . Therefore, these agents are contained in  $\text{SAT}_t$ , for all iterations  $t \geq 1$ , and do not contribute to the repetitions of the outer while loop.

For the remaining agents, with  $\ell_i > 0$ , the algorithm initially sets  $\gamma_i^0 = v_i([m])$  and we have

$$\ell_i \geq \frac{1}{2n} \min_{g \in [m]: v_i(g) > 0} v_i(g). \quad (7)$$

Using Lemma 6 and the fact that the algorithm decrements  $\gamma_i^t$  by a multiplicative factor of  $(1 - 1/m)$  whenever  $i \in \text{UNSAT}_t$ , we get that the number of times agent  $i$  can be in the  $\text{UNSAT}_t$  is at most

$$\begin{aligned} m \log \left( \frac{v_i([m])}{\ell_i} \right) &\leq m \log \left( \frac{m \max_{g \in [m]} v_i(g)}{\ell_i} \right) \\ &\quad \text{(since } v_i \text{ is subadditive, } v_i([m]) \leq m \max_{g \in [m]} v_i(g)\text{)} \\ &\leq m \log \left( \frac{2nm \max_{g \in [m]} v_i(g)}{\min_{g \in [m]: v_i(g) > 0} v_i(g)} \right) \quad \text{(via inequality (7))} \\ &\leq m \log(2nmV). \end{aligned}$$

Summing over all agents, we get that the number of times  $\text{UNSAT}_t \neq \emptyset$  is at most  $T = O(nm \log(nmV))$ . Hence, the stated lemma follows.  $\blacktriangleleft$

## 11:12 Tight Approximation Algorithms for $p$ -Mean Welfare Under Subadditive Valuations

We now show that the allocation computed by ALG achieves the required approximation guarantee.

► **Lemma 8 (Approximation Guarantee).** *For any fair division instance  $\mathcal{I} = \langle [m], [n], \{v_i\}_{i=1}^n \rangle$  with subadditive valuations, let  $\mathcal{B} = (B_1, \dots, B_n)$  denote the allocation computed by ALG. Then,  $\text{NSW}(\mathcal{B}) \geq \frac{1}{8n} \text{NSW}(\mathcal{N}^*)$ ; here,  $\mathcal{N}^*$  denotes the Nash optimal allocation in  $\mathcal{I}$ .*

**Proof.** For the given instance  $\mathcal{I}$ , say ALG terminates after  $T + 1$  iterations of the outer while loop. That is, we have  $\text{UNSAT}_{T+1} = \emptyset$  and, for each agent  $i \in [n]$ , the returned bundle  $B_i = B_i^T \cup \{\pi^T(i)\}$ . Here,  $\pi^T(i)$  is the good assigned to agent  $i$  under the maximum weight matching  $\pi^T$  (considered in the last iteration) and  $B_i^T$  is the bundle populated for  $i$  (either in Step 15 or in Step 17).

The fact that  $\text{UNSAT}_{T+1} = \emptyset$  (i.e.,  $\text{SAT}_{T+1} = [n]$ ) gives us

$$v_i(B_i^T) \geq \gamma_i^T \quad \text{for all } i \in [n]. \quad (8)$$

Lemma 3 (instantiated with  $\mathcal{A}^* = \mathcal{N}^*$ ) implies that there exists a matching  $\sigma(i) := \hat{g}_i \in \mathcal{N}_i^*$ , for all  $i \in [n]$  – with the property that  $v_i(\sigma(i)) + \ell_i \geq \frac{1}{4n} v_i(\mathcal{N}_i^*)$ . Using this inequality and Lemma 6 we get, for all  $i \in [n]$ :

$$v_i(\sigma(i)) + \gamma_i^T \geq \left(1 - \frac{1}{m}\right) \frac{1}{4n} v_i(\mathcal{N}_i^*). \quad (9)$$

Recall that  $\pi^T$  is a maximum weight matching in the bipartite graph (considered in Step 11 of ALG) with edge weights  $\log(v_i(g) + \gamma_i^T)$ . Given that  $\sigma(\cdot)$  is some matching in the graph and  $\pi^T$  is a maximum weight matching, we get  $\sum_{i=1}^n \log(v_i(\pi^T(i)) + \gamma_i^T) \geq \sum_{i=1}^n \log(v_i(\sigma(i)) + \gamma_i^T)$ . That is,

$$\left(\prod_{i=1}^n (v_i(\pi^T(i)) + \gamma_i^T)\right)^{\frac{1}{n}} \geq \left(\prod_{i=1}^n (v_i(\sigma(i)) + \gamma_i^T)\right)^{\frac{1}{n}} \geq \left(1 - \frac{1}{m}\right) \frac{1}{4n} \text{NSW}(\mathcal{N}^*). \quad (10)$$

The last inequality follows from equation (9). Also, as defined previously, the optimal Nash social welfare  $\text{NSW}(\mathcal{N}^*) = (\prod_{i=1}^n v_i(\mathcal{N}_i^*))^{1/n}$ .

The monotonicity of the valuation function  $v_i$  implies  $v_i\{\pi^T(i)\} \cup B_i^T \geq 1/2(v_i(\pi^T(i)) + v_i(B_i^T))$  for each  $i \in [n]$ . Using these observations we can lower bound the Nash social welfare of the computed allocation  $(B_i = \{\pi^T(i)\} \cup B_i^T)_i$  as follows

$$\begin{aligned} \left(\prod_{i=1}^n v_i(B_i)\right)^{\frac{1}{n}} &\geq \frac{1}{2} \left(\prod_{i=1}^n (v_i(\pi^T(i)) + v_i(B_i^T))\right)^{\frac{1}{n}} \\ &\geq \frac{1}{2} \left(\prod_{i=1}^n (v_i(\pi^T(i)) + \gamma_i^T)\right)^{\frac{1}{n}} && \text{(via inequality (8))} \\ &\geq \left(1 - \frac{1}{m}\right) \frac{1}{8n} \text{NSW}(\mathcal{N}^*). && \text{(via inequality (10))} \end{aligned}$$

This establishes the stated approximation guarantee and completes the proof of the lemma. ◀

► **Remark.** The result of Garg et al. [20] also holds for an asymmetric version of Nash social welfare maximization, in which each agent  $i$  has an associated weight  $\eta_i \geq 0$  and the goal is to find an allocation  $(A_1, \dots, A_n)$  that maximizes  $\left(\prod_{i \in [n]} (v_i(A_i))^{\eta_i}\right)^{\frac{1}{\sum_{i \in [n]} \eta_i}}$ . Our

approximation guarantee extends to this formulation. In particular, in Step 11 of ALG we can set the edges weights to be  $\eta_i \log(v_i(g) + \gamma_i)$  (instead of  $\log(v_i(g) + \gamma_i)$ ) and note that the subsequent arguments follow through to provide an  $8n$ -approximation ratio for maximizing Nash social welfare with asymmetric agents and subadditive valuations.

Also, one can use Theorem 2, in conjunction with the  $m/n$  approximation guarantee of Nguyen and Rothe [26],<sup>4</sup> to obtain an  $O(\sqrt{m})$ -approximation algorithm for maximizing Nash social welfare under subadditive valuations: for instances in which  $m \geq n^2$ , the  $8n$  approximation suffices. Otherwise, if  $m < n^2$  (i.e.,  $m/n < \sqrt{m}$ ), then we can invoke the result of Nguyen and Rothe [26].

#### 4 An $8n$ -Approximation for $p$ -Mean Welfare

This section shows that we can extend Algorithm 1 and obtain an  $8n$  approximation for maximizing the  $p$ -mean welfare as well.

For maximizing  $p$ -mean welfare, ALG (Algorithm 1) is modified as follows: In Step 11, the weight  $w(i, g)$  of edge  $(i, g) \in [n] \times [m]$  is set as  $(v_i(g) + \gamma_i^t)^p$  (instead of  $\log(v_i(g) + \gamma_i^t)$ ).<sup>5</sup> Furthermore,

- (i) For  $p \in (0, 1]$ , in Step 12 we compute a left-perfect *maximum*-weight matching,  $\pi^t$ , otherwise
- (ii) For finite  $p < 0$ , we compute a left-perfect *minimum*-weight matching,  $\pi^t$ , in Step 12
- (iii) For maximizing egalitarian welfare (the  $p = -\infty$  case), we set edge weights to be  $(v_i(g) + \gamma_i^t)$  and compute a max-min matching<sup>6</sup>  $\pi^t$  with respect to these weights.

Theorem 9 below establishes that, with these changes in ALG (Algorithm 1), we can efficiently compute an allocation with  $p$ -mean welfare at least  $\frac{1}{8n}$  times the optimal ( $p$ -mean welfare). Note that by Proposition 1, for  $p \leq -n \log n$ , we can maximize the egalitarian welfare, instead of the  $p$ -mean welfare, and the allocation thus obtained is an  $8n$ -approximation to the optimal  $p$ -mean welfare allocation.

► **Theorem 9.** *Let  $\mathcal{I} = \langle [m], [n], \{v_i\}_{i=1}^n \rangle$  be a fair division instance in which the valuation function  $v_i$ , of each agent  $i \in [n]$ , is nonnegative, monotone, and subadditive. Then, given value oracle access to  $v_i$ s, one can efficiently compute an  $8n$  approximation to the optimal  $p$ -mean welfare for any  $p \in (-\infty, 1]$ .*

**Proof.** We first note that Lemmas 3, 4, 5, 6, and 7 hold as is for  $p$ -mean welfare. In particular, using Lemma 7 we get that, even with the above-mentioned changes, the algorithm runs in polynomial time.

To complete the proof of the theorem, we will next show that the computed allocation  $\mathcal{B} = (B_1, \dots, B_n)$  satisfies  $M_p(\mathcal{B}) \geq \frac{1}{8n} M_p(\mathcal{A}^*(p))$ , where  $\mathcal{A}^*(p)$  is a  $p$ -mean welfare maximizing allocation.

For the given instance  $\mathcal{I}$ , say the modified algorithm terminates after  $T + 1$  iterations of the outer while loop. That is, we have  $\text{UNSAT}_{T+1} = \emptyset$  and, for each agent  $i \in [n]$ , the returned bundle  $B_i = B_i^T \cup \{\pi^T(i)\}$ . Here,  $\pi^T(i)$  is the good assigned to agent  $i$  under the matching  $\pi^T$  (considered in the last iteration) and  $B_i^T$  is the bundle populated for  $i$  in the final iteration.

<sup>4</sup> While Theorem 4 in [26] provides the above-mentioned approximation guarantee of  $(m - n + 1)$ , its proof can in fact be easily modified to obtain an approximation ratio of  $m/n$ .

<sup>5</sup> Recall that the  $p = 0$  case corresponds to Nash social welfare. Since we already have the desired approximation guarantee for this case, it is not explicitly addressed in this section.

<sup>6</sup> In particular, via binary search (over edge weights), we find a matching wherein the minimum edge weight (across agents) is as high as possible.

## 11:14 Tight Approximation Algorithms for $p$ -Mean Welfare Under Subadditive Valuations

The fact that  $\text{UNSAT}_{T+1} = \emptyset$  (i.e.,  $\text{SAT}_{T+1} = [n]$ ) gives us

$$v_i(B_i^T) \geq \gamma_i^T \quad \text{for all } i \in [n] \quad (11)$$

Lemma 3 implies that there exists a matching  $\sigma(i) := \widehat{g}_i \in A_i^*(p)$ , for all  $i \in [n]$  – with the property that  $v_i(\sigma(i)) + \ell_i \geq \frac{1}{4n}v_i(A_i^*(p))$ . Using this inequality and Lemma 6 we get, for all  $i \in [n]$ :

$$v_i(\sigma(i)) + \gamma_i^T \geq \left(1 - \frac{1}{m}\right) \frac{1}{4n}v_i(A_i^*(p)) \quad (12)$$

Recall that, given  $p$ , the modified algorithm computes  $\pi^T$  based on the sign of  $p$ . Hence, we split the proof of Theorem 9 into three cases depending on whether  $p > 0$ ,  $p < 0$ , or  $p = -\infty$ .

**Case (i).**  $p > 0$ . In this case,  $\pi^T$  is a left-perfect maximum-weight matching in the bipartite graph  $([n] \cup [m], [n] \times [m])$  with edge weights  $(v_i(g) + \gamma_i^T)^p$ . Given that  $\sigma(\cdot)$  is some (left-perfect) matching in the graph and  $\pi^T$  is a maximum-weight matching, we get  $\sum_{i=1}^n (v_i(\pi^T(i)) + \gamma_i^T)^p \geq \sum_{i=1}^n (v_i(\sigma(i)) + \gamma_i^T)^p$ . Therefore, with  $p > 0$ , the following inequality holds

$$\left(\frac{1}{n} \sum_{i=1}^n (v_i(\pi^T(i)) + \gamma_i^T)^p\right)^{\frac{1}{p}} \geq \left(\frac{1}{n} \sum_{i=1}^n (v_i(\sigma(i)) + \gamma_i^T)^p\right)^{\frac{1}{p}} \geq \left(1 - \frac{1}{m}\right) \frac{1}{4n}M_p(\mathcal{A}^*(p)). \quad (13)$$

The last inequality follows from (12).

**Case (ii).** Finite  $p < 0$ . By design, in this case,  $\pi^T$  is a left-perfect minimum-weight matching in the bipartite graph  $([n] \cup [m], [n] \times [m])$  with edge weights  $(v_i(g) + \gamma_i^T)^p$ . Given that  $\sigma(\cdot)$  is some left-perfect matching in the graph and  $\pi^T$  is a minimum-weight matching, we get  $\sum_{i=1}^n (v_i(\pi^T(i)) + \gamma_i^T)^p \leq \sum_{i=1}^n (v_i(\sigma(i)) + \gamma_i^T)^p$ . The fact that  $p$  is negative gives us

$$\left(\frac{1}{n} \sum_{i=1}^n (v_i(\pi^T(i)) + \gamma_i^T)^p\right)^{\frac{1}{p}} \geq \left(\frac{1}{n} \sum_{i=1}^n (v_i(\sigma(i)) + \gamma_i^T)^p\right)^{\frac{1}{p}} \geq \left(1 - \frac{1}{m}\right) \frac{1}{4n}M_p(\mathcal{A}^*(p)) \quad (14)$$

The last inequality follows from (12).

**Case (iii).**  $p = -\infty$ . In this case,  $\pi^T$  is a max-min matching computed with edge weights  $(v_i(\pi^T(i)) + \gamma_i^T)$ . Given that  $\sigma(\cdot)$  is some matching in the graph and matching  $\pi^T$  maximizes the value of the minimum matched edge, we get  $\min_{i \in [n]} (v_i(\pi^T(i)) + \gamma_i^T) \geq \min_{i \in [n]} (v_i(\sigma^T(i)) + \gamma_i^T)$ . Therefore,

$$\min_{i \in [n]} (v_i(\pi^T(i)) + \gamma_i^T) \geq \min_{i \in [n]} (v_i(\sigma^T(i)) + \gamma_i^T) \geq \left(1 - \frac{1}{m}\right) \frac{1}{4n}M_p(\mathcal{A}^*(p)) \quad (15)$$

The last inequality follows from (12).

The monotonicity of the valuation function  $v_i$  implies  $v_i(\{\pi^T(i)\} \cup B_i^T) \geq 1/2(v_i(\pi^T(i)) + v_i(B_i^T))$  for each  $i \in [n]$ . Using these observations we can lower bound the  $p$ -mean welfare of the computed allocation  $(B_i = \{\pi^T(i)\} \cup B_i^T)_i$  as follows

$$\begin{aligned}
\left(\sum_{i=1}^n (v_i(B_i))^p\right)^{\frac{1}{p}} &\geq \frac{1}{2} \left(\sum_{i=1}^n (v_i(\pi^T(i)) + v_i(B_i^T))^p\right)^{\frac{1}{p}} \\
&\geq \frac{1}{2} \left(\sum_{i=1}^n (v_i(\pi^T(i)) + \gamma_i^T)^p\right)^{\frac{1}{p}} && \text{(via inequality (11))} \\
&\geq \left(1 - \frac{1}{m}\right) \frac{1}{8n} M_p(\mathcal{A}^*(p)) && \text{(via inequality (13), (14), or (15))}
\end{aligned}$$

This establishes the stated approximation guarantee and completes the proof of the theorem.  $\blacktriangleleft$

## 5 Lower Bound on Approximating $p$ -Mean Welfare

This section shows that, under XOS valuations, maximizing the  $p$ -mean welfare for  $p \in (-\infty, 1]$  within a sub-linear (in  $n$ ) approximation factor necessarily requires an exponential number of value queries (Theorem 10). This result directly implies that the approximation ratio obtained in Theorems 2 and 9 (via polynomially many value queries) is essentially tight. We note that this query lower bound is unconditional, i.e., it does not depend on any complexity theoretic assumption.

We establish Theorem 10 by directly adapting a result of Dobzinski et al. [16], which provides a similar lower bound for social welfare. The impossibility result here holds under XOS valuations;<sup>7</sup> recall that XOS valuations constitute a special class of subadditive functions.

**► Theorem 10.** *For fair division instances  $\mathcal{I} = \langle [m], [n], \{v_i\}_{i=1}^n \rangle$  with XOS valuations and  $p \in (-\infty, 1]$ , finding an allocation with  $p$ -mean welfare at least  $1/n^{1-\varepsilon}$  times the optimal requires exponentially many value queries; here  $\varepsilon > 0$  is any fixed constant.*

Here, we briefly explain the salient points of the proof of this lower bound and provide the details in the full version of the current paper [4]. Dobzinski et al. [16] construct two (families of) instances, both with  $n$  agents,  $m = n^2$  goods, and XOS valuations for the agents. In the first instance, each agent has the same valuation function  $f : 2^{[m]} \mapsto \mathbb{R}_+$  and maximum average social welfare (1-mean welfare) is  $n^{4\delta}$ , for a fixed constant  $\delta > 0$ . In the second instance, each agent has her own (non-identical) valuation function  $v_i : 2^{[m]} \mapsto \mathbb{R}_+$  and there exists an allocation in which each agent has value  $n$  for her bundle. For any  $p \leq 1$ , it follows that in the first instance the optimal  $p$ -mean welfare is at most  $n^{4\delta}$  (via the generalized mean inequality), while for the second instance, the optimal  $p$ -mean welfare is at least  $n$  (since there exists an allocation where every agent achieves value  $n$ ). The proof of Dobzinski et al. [16] goes on to show that it takes an exponential number of value queries to distinguish between the two instances. However, given an  $O(n^{1-\varepsilon})$ -approximation algorithm for the  $p$ -mean welfare, one can readily distinguish between the two instances (by choosing  $\delta < \varepsilon/4$ ). Hence such an algorithm must make an exponential number of value queries.

<sup>7</sup> Our results work under the value oracle model and do not require an explicit description of the underlying additive functions that define the XOS function at hand.

**6**  $(m - n + 1)$ -Approximation Guarantees

This section provides two extensions of the result of Nguyen and Rothe [26], which shows the Nash social welfare maximization problem (under subadditive valuations) admits an  $(m - n + 1)$ -approximation algorithm. First, we show that an  $(m - n + 1)$ -approximation for the  $p$ -mean welfare can be obtained for all  $p \leq 0$  and with subadditive valuations. Then, we establish that it is NP-hard to extend this positive result to any  $p \in (0, 1)$ , even under additive valuations, i.e., it is NP-hard to obtain an  $(m - n + 1)$ -approximation for  $0 < p < 1$ . The proofs of these two results are delegated to the full version of this paper [4].

► **Theorem 11.** *Let  $\mathcal{I} = \langle [m], [n], \{v_i\}_{i=1}^n \rangle$  be a fair division instance in which the valuation function  $v_i$ , of each agent  $i \in [n]$ , is nonnegative, monotone, and subadditive. Then, given value oracle access to  $v_i$ s, one can efficiently compute an  $(m - n + 1)$  approximation to the  $p$ -mean welfare maximization problem for any  $p \in (-\infty, 0]$ .*

The next theorem asserts that it is unlikely that Theorem 11 extends to  $p \in (0, 1)$ .

► **Theorem 12.** *For fair division instances  $\mathcal{I} = \langle [m], [n], \{v_i\}_{i=1}^n \rangle$  with additive valuations and for any fixed  $p \in (0, 1)$ , computing an allocation with  $p$ -mean welfare at least  $1/(m - n + 1)$ -times the optimal (for all  $m$  and  $n$ ) is NP-hard.*

Note that this hardness result (in light of Theorem 9) is relevant for instances in which  $m < 2n$ .

---

**References**

- 1 Nima Anari, Shayan Oveis Gharan, Tung Mai, and Vijay V Vazirani. Nash Social Welfare for Indivisible Items under Separable, Piecewise-Linear Concave Utilities. In *Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2274–2290, 2018.
- 2 Nima Anari, Shayan Oveis Gharan, Amin Saberi, and Mohit Singh. Nash Social Welfare, Matrix Permanent, and Stable Polynomials. In *Proceedings of the 8th Conference on Innovations in Theoretical Computer Science (ITCS)*, 2017.
- 3 Haris Aziz. Developments in multi-agent fair allocation. In *AAAI*, pages 13563–13568, 2020.
- 4 Siddharth Barman, Umang Bhaskar, Anand Krishna, and Ranjani G. Sundaram. Tight approximation algorithms for  $p$ -mean welfare under subadditive valuations. *CoRR*, abs/2005.07370, 2020. [arXiv:2005.07370](https://arxiv.org/abs/2005.07370).
- 5 Siddharth Barman, Sanath Kumar Krishnamurthy, and Rohit Vaish. Finding fair and efficient allocations. In *Proceedings of the 2018 ACM Conference on Economics and Computation, Ithaca, NY, USA, June 18-22, 2018*, pages 557–574. ACM, 2018.
- 6 Xiaohui Bei, Jugal Garg, Martin Hoefer, and Kurt Mehlhorn. Earning Limits in Fisher Markets with Spending-Constraint Utilities. In *Proceedings of the International Symposium on Algorithmic Game Theory (SAGT)*, pages 67–79, 2017.
- 7 Ivona Bezáková and Varsha Dani. Allocating indivisible goods. *SIGecom Exchanges*, 5(3):11–18, 2005.
- 8 Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D Procaccia. *Handbook of computational social choice*. Cambridge University Press, 2016.
- 9 Ioannis Caragiannis, David Kurokawa, Hervé Moulin, Ariel D. Procaccia, Nisarg Shah, and Junxing Wang. The unreasonable fairness of maximum nash welfare. *ACM Trans. Economics and Comput.*, 7(3):12:1–12:32, 2019.
- 10 Deeparnab Chakrabarty, Julia Chuzhoy, and Sanjeev Khanna. On allocating goods to maximize fairness. In *50th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2009, October 25-27, 2009, Atlanta, Georgia, USA*, pages 107–116. IEEE Computer Society, 2009.



- 11 Bhaskar Ray Chaudhury, Yun Kuen Cheung, Jugal Garg, Naveen Garg, Martin Hoefer, and Kurt Mehlhorn. On fair division for indivisible items. In *38th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2018, December 11-13, 2018, Ahmedabad, India*, volume 122 of *LIPICs*, pages 25:1–25:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.
- 12 Bhaskar Ray Chaudhury, Jugal Garg, and Ruta Mehta. Fair and efficient allocations under subadditive valuations, 2020. [arXiv:2005.06511](https://arxiv.org/abs/2005.06511).
- 13 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 *Proceedings of the 2017 ACM Conference on Economics and Computation, EC '17, Cambridge, MA, USA, June 26-30, 2017*, pages 459–460. ACM, 2017.
- 14 Richard Cole and Vasilis Gkatzelis. Approximating the Nash Social Welfare with Indivisible Items. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing (STOC)*, pages 371–380, 2015.
- 15 Vincent Conitzer, Rupert Freeman, and Nisarg Shah. Fair public decision making. In *Proceedings of the 2017 ACM Conference on Economics and Computation, EC '17, Cambridge, MA, USA, June 26-30, 2017*, pages 629–646. ACM, 2017.
- 16 Shahar Dobzinski, Noam Nisan, and Michael Schapira. Approximation algorithms for combinatorial auctions with complement-free bidders. *Math. Oper. Res.*, 35(1):1–13, 2010.
- 17 Ulle Endriss. *Trends in Computational Social Choice*. Lulu. com, 2017.
- 18 Uriel Feige. On maximizing welfare when utility functions are subadditive. *SIAM J. Comput.*, 39(1):122–142, 2009.
- 19 Jugal Garg, Martin Hoefer, and Kurt Mehlhorn. Approximating the Nash Social Welfare with Budget-Additive Valuations. In *Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2326–2340, 2018.
- 20 Jugal Garg, Pooja Kulkarni, and Rucha Kulkarni. Approximating nash social welfare under submodular valuations through (un)matchings. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 2673–2687. SIAM, 2020.
- 21 Michel X. Goemans, Nicholas J. A. Harvey, Satoru Iwata, and Vahab S. Mirrokni. Approximating submodular functions everywhere. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2009, New York, NY, USA, January 4-6, 2009*, pages 535–544. SIAM, 2009.
- 22 Jonathan Goldman and Ariel D Procaccia. Spliddit: Unleashing fair division algorithms. *ACM SIGecom Exchanges*, 13(2):41–46, 2015.
- 23 Subhash Khot and Ashok Kumar Ponnuswami. Approximation algorithms for the max-min allocation problem. In *APPROX-RANDOM*, 2007.
- 24 Euiwoong Lee. Apx-hardness of maximizing nash social welfare with indivisible items. *Inf. Process. Lett.*, 122:17–20, 2017.
- 25 Hervé Moulin. *Fair division and collective welfare*. MIT Press, 2003.
- 26 Trung Thanh Nguyen and Jörg Rothe. Minimizing envy and maximizing average nash social welfare in the allocation of indivisible goods. *Discret. Appl. Math.*, 179:54–68, 2014.
- 27 Noam Nisan, T Roughgarden, E Tardos, and Vijay V Vazirani. *Algorithmic game theory* - Cambridge University Press, 2007.
- 28 Zoya Svitkina and Lisa Fleischer. Submodular approximation: Sampling-based algorithms and lower bounds. *SIAM Journal on Computing*, 40(6):1715–1737, 2011.
- 29 Jan Vondrák. Optimal approximation for the submodular welfare problem in the value oracle model. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 67–74. ACM, 2008.