

A Note on Approximating Weighted Nash Social Welfare with Additive Valuations

Received Oct 30, 2024
 Revised May 27, 2025
 Accepted Jun 4, 2025
 Published Aug 14, 2025

Key words and phrases
 Nash Social Welfare, Configuration LP, Approximation Algorithms

Yuda Feng^{a,b} ✉ 

Shi Li^{a,b} ✉ 

a School of Computer Science,
 Nanjing University, Nanjing,
 Jiangsu, China

b New Cornerstone Science
 Laboratory

ABSTRACT. We give the first $O(1)$ -approximation for the weighted Nash Social Welfare problem with additive valuations. The approximation ratio we obtain is $e^{1/e} + \epsilon \approx 1.445 + \epsilon$, which matches the best known approximation ratio for the unweighted case [3].

Both our algorithm and analysis are simple. We solve a natural configuration LP for the problem, and obtain the allocation of items to agents using the Shmoys-Tardos rounding algorithm developed for unrelated machine scheduling problems [32]. In the analysis, we show that the approximation ratio of the algorithm is at most the worst gap between the Nash social welfare of the optimum allocation and that of an EF1 allocation, for an unweighted Nash Social Welfare instance with identical additive valuations. This was shown to be at most $e^{1/e} \approx 1.445$ by Barman, Krishnamurthy and Vaish [3], leading to our approximation ratio.

1. Introduction

In the weighted (or asymmetric) Nash Social Welfare problem with additive valuations, we are given a set \mathcal{A} of n agents, and a set \mathcal{G} of m indivisible items. Every agent $i \in \mathcal{A}$ has a weight $w_i \geq 0$ such that $\sum_{i \in \mathcal{A}} w_i = 1$. There is a value $v_{ij} \in \mathbb{R}_{\geq 0}$ for every $i \in \mathcal{A}$ and $j \in \mathcal{G}$. The goal of the problem is to find an allocation $\sigma : \mathcal{G} \rightarrow \mathcal{A}$ of items to agents so as to maximize the

This paper was invited from ICALP 2024 [13]. The work of YF and SL was supported by the State Key Laboratory for Novel Software Technology, and the New Cornerstone Science Laboratory.

following weighted Nash Social Welfare of σ :

$$\prod_{i \in \mathcal{A}} \left(\sum_{j \in \sigma^{-1}(i)} v_{ij} \right)^{w_i}.$$

In the case where all w_i 's are equal to $\frac{1}{n}$, we call the problem the unweighted (or symmetric) Nash Social Welfare problem.

Allocating resources in a fair and efficient manner among multiple agents is a fundamental problem in computer science, game theory, and economics, with applications across diverse domains [21, 35, 4, 30, 27, 2, 31, 5]. The weighted Nash Social Welfare function is a notable objective that balances efficiency and fairness. The unweighted (or symmetric) objective was independently proposed by different communities [28, 22, 34], and later the study has been extended to the weighted case [18, 20]. Since then it has been used in a wide range of applications, including bargaining theory [23, 7, 33], water allocation [19, 10], and climate agreements [36].

The unweighted Nash Social Welfare problem with additive valuations is proved to be NP-hard by Nguyen, Nguyen, Roos and Rothe [29], and APX-hard by Lee [24]. Later the hardness of approximation was improved to $\sqrt{8/7} \approx 1.069$ by Garg, Hoefer and Mehlhorn [14], via a reduction from Max-E3-Lin-2.

On the positive side, Cole and Gkatzelis [9] gave a $(2e^{1/e} + \epsilon \approx 2.889 + \epsilon)$ -approximation using a spending restricted market equilibrium relaxation. The ratio was improved by Cole, Devanur, Gkatzelis, Jain, Mai, Vazirani and Yazdanbod [8] to 2 using a tight analysis, and by Anari, Oveis Gharan, Saberi and Singh [1] to e via a connection of the problem to real stable polynomials. Both papers formulated convex programming (CP) relaxations of the problem. In particular, [8] showed that the optimum solution to their CP corresponds to the spending-restricted market equilibrium defined in [9]. The state-of-the-art result for the problem is a combinatorial $(e^{1/e} + \epsilon \approx 1.445 + \epsilon)$ -approximation algorithm due to Barman, Krishnamurthy and Vaish [3]. They showed that when all the valuations of agents are identical, any allocation that is envy-free up to one item (EF1) is $e^{1/e}$ -approximate. Their approximation result then follows from a connection between the non-identical and identical valuation settings they established.

All the results discussed above are for the unweighted case. For the weighted case with agent weights $w \in [0, 1]^{\mathcal{A}}$, $|w|_1 = 1$, Brown, Laddha, Pittu and Singh [6] presented a $5 \cdot \exp(2 \cdot D_{\text{KL}}(w || \frac{\vec{1}}{n})) = 5 \cdot \exp(2 \log n + 2 \sum_{i \in \mathcal{A}} w_i \log w_i)$ approximation algorithm, where D_{KL} denotes the KL divergence of two distributions. Their result is based on the CP from [8], generalized to the weighted setting.

The additive valuation setting is a special case of the submodular valuation setting, which is another important setting studied in the literature. In this setting, instead of a v_{ij} value for every ij pair, we are given a monotone submodular function $v_i : 2^{\mathcal{G}} \rightarrow \mathbb{R}_{\geq 0}$ for every agent $i \in \mathcal{A}$. Till the end of the section, our goal is to find an allocation $\sigma : \mathcal{G} \rightarrow \mathcal{A}$ so as to maximize

$\prod_{i \in \mathcal{A}} (v_i(\sigma^{-1}(i)))^{w_i}$. A bulk of the previous work has focused on the unweighted case; that is, $w_i = \frac{1}{n}$ for all $i \in \mathcal{A}$. For this case, [17] proved a hardness of $e/(e-1) \approx 1.5819$ using a reduction from Max-3-Coloring; this is better than the 1.069 hardness for the additive valuation case.

On the positive side, Li and Vondrák [26] extended the techniques of [1], to obtain an $e^3/(e-1)^2$ -approximation algorithm for the unweighted Nash Social Welfare problem for a large family of submodular valuations, including coverage functions and linear combinations of matroid rank functions. Later, Garg, Husić, and Véggh [16] considered a family of submodular functions called Rado functions, and gave an $O(1)$ -approximation for this family using the matching theory and convex program techniques. Li and Vondrák [25] developed the first $O(1)$ -approximation for general submodular functions, with an approximation ratio of 380. Recently, Garg, Husić, Li, Véggh and Vondrák [15] presented an elegant 4-approximation local search algorithm for the problem, which is the current best approximation result for the problem. All the results discussed above are for the unweighted case. For the weighted case, [15] gave an $O(nw_{\max})$ -approximation, where $w_{\max} = \max_{i \in \mathcal{A}} w_i$.

Using our configuration LP idea, Feng, Hu, Li and Zhang [12] recently developed a $(233 + \epsilon)$ -approximation for the weighted case, which is the first $O(1)$ -approximation for the weighted Nash Social Welfare problem with submodular valuations.

Recently, the problem has been studied in an even more general setting, namely, the subadditive valuation setting. Dobzinski, Li, Rubinstein and Vondrák [11] gave an $O(1)$ -approximation for the unweighted Nash Social Welfare problem in this setting under the demand oracle model.

1.1 Our Result and Techniques

In this note, we give the first $O(1)$ -approximation algorithm for the weighted Nash Social Welfare problem with additive valuations:

THEOREM 1.1. *For any $\epsilon > 0$, there is a deterministic $(e^{1/\epsilon} + \epsilon \approx 1.445 + \epsilon)$ -approximation algorithm for the weighted Nash Social Welfare problem with additive valuations, with running time polynomial in the size of the input and $\frac{1}{\epsilon}$.*

Our approximation ratio of $e^{1/\epsilon} + \epsilon$ matches the best ratio for the unweighted case due to Barman, Krishnamurthy and Vaish [3]. In contrast, the ratio given by [6] is $5 \cdot \exp(2 \cdot D_{\text{KL}}(w || \frac{\vec{1}}{n}))$, which could be polynomial in n .

Our algorithm is based on a natural configuration LP for the problem, which has not been studied before to the best of our knowledge. The configuration LP contains a $y_{i,S}$ variable for every agent i and subset S of items, indicating if the set of items i gets is S or not. We show that the configuration LP can be solved in polynomial time to any precision, despite having an exponential number of variables. Once we obtain the LP solution, we define x_{ij} for every $i \in \mathcal{A}$ and $j \in \mathcal{G}$ to be the fraction of j assigned to i .

To round x into an integral solution, it is more convenient to focus on a randomized algorithm first, which uses the Shmoys-Tardos rounding algorithm [32] developed for unrelated machine scheduling problems. For every agent i , we break the fractional items assigned to i into groups from the most valuable to the least, each containing 1 fractional item. The rounding algorithm maintains marginal probabilities, and the requirement that i gets exactly one item from each group (except for the last one, from which i gets at most one item). In the analysis for each agent i , we construct an instance of the unweighted Nash Social Welfare problem with *identical* additive valuations, that involves many copies of the agent i , along with two allocations \mathcal{S} and \mathcal{S}' to the instance. \mathcal{S} corresponds to the LP solution, and \mathcal{S}' corresponds to the randomized solution given by the rounding algorithm. Thanks to the condition that every group contains one item, the solution \mathcal{S}' is envy-free up to one item (EF1), defined as follows.

DEFINITION 1.2. Given an instance of the unweighted Nash Social Welfare problem with agents \mathcal{A} , items \mathcal{G} , and identical additive valuation $v : \mathcal{G} \rightarrow \mathbb{R}_{\geq 0}$ for all agents, an allocation $\sigma : \mathcal{G} \rightarrow \mathcal{A}$ is said to be envy-free up to one item (EF1), if for every two distinct agents i, i' with $\sigma^{-1}(i') \neq \emptyset$, there exists some $j \in \sigma^{-1}(i')$, such that $v(\sigma^{-1}(i') \setminus j) \leq v(\sigma^{-1}(i))$.

Barman, Krishnamurthy and Vaish [3] showed the following result on the quality of EF1 allocations:

THEOREM 1.3 ([3]). *For the unweighted Nash Social Welfare problem with identical additive valuations, any EF1-allocation is an $e^{1/e}$ -approximate solution.*

The result implies that the Nash social welfare of \mathcal{S}' is at least $e^{-1/e}$ times that of \mathcal{S} . This in turn proves that the expected weighted Nash Social Welfare of the solution returned by our rounding algorithm is at least the exponential of the value of the LP solution. To de-randomize the algorithm, we break the fractional matching between groups and items into a convex combination of polynomial number of integral matchings, and choose the best matching in the combination.

2. $(e^{1/e} + \varepsilon)$ -Approximation Using Configuration LP

At the beginning of the algorithm, we check if there is an allocation with positive weighted Nash Social Welfare, using the bipartite matching algorithm. If not, we terminate the algorithm immediately by returning any allocation. From now on, we assume the optimum weighted Nash Social Welfare value is positive. We describe the configuration LP in Section 2.1 and the rounding algorithm in Section 2.2. The analysis is given in Section 2.3.

2.1 The Configuration LP

For convenience, for any value function $v : \mathcal{G} \rightarrow \mathbb{R}_{\geq 0}$, we define $v(S) := \sum_{j \in S} v_j$ for every $S \subseteq \mathcal{G}$ to be the total value of items in S according to the value function v . In the integer program corresponding to the configuration LP, for every $i \in \mathcal{A}$ and $S \subseteq \mathcal{G}$, we have a variable $y_{i,S} \in \{0, 1\}$ indicating if the set of items assigned to i is S or not. We relax the integer constraint to obtain the following configuration LP:

$$\max \sum_{i \in \mathcal{A}, S \subseteq \mathcal{G}} w_i \cdot y_{i,S} \cdot \ln v_i(S) \quad \text{s.t.} \quad (\text{Conf-LP})$$

$$\sum_{i \in \mathcal{A}, S \ni j} y_{i,S} \leq 1 \quad \forall j \in \mathcal{G} \quad (1)$$

$$\sum_{S \subseteq \mathcal{G}} y_{i,S} = 1 \quad \forall i \in \mathcal{A} \quad (2)$$

$$y_{i,S} \geq 0 \quad \forall i \in \mathcal{A}, S \subseteq \mathcal{G} \quad (3)$$

It is convenient for us to consider the natural logarithm of the Nash social welfare function as the objective, which is $\sum_{i \in \mathcal{A}} w_i \cdot \ln v_i(\sigma^{-1}(i))$. This leads to the objective in (Conf-LP). (1) requires that every item j is assigned to at most one agent, and (2) requires that every agent i is assigned one set of items.

The configuration LP has an exponential number of variables, but it can be solved within an additive error of $\ln(1 + \epsilon)$ for any $\epsilon > 0$, in time polynomial in the size of the instance and $\frac{1}{\epsilon}$.

THEOREM 2.1. *For any $\epsilon > 0$, there is an algorithm that outputs a valid solution $(y_{i,S} \in \mathbb{Q}_{\geq 0})_{i \in \mathcal{A}, S \subseteq \mathcal{G}}$ to (Conf-LP) whose value is at least the optimum value of the LP minus $\ln(1 + \epsilon)$, represented using a list of the non-zero entries. The running time of the algorithm is polynomial in the input size and $\frac{1}{\epsilon}$.*

We defer the proof of Theorem 2.1 to Section 3. Notice that we are considering the logarithm of Nash social welfare, and the typical $(1 + \epsilon)$ -multiplicative factor becomes an additive error of $\ln(1 + \epsilon)$.

2.2 The Rounding Algorithm

From now on, we assume we have obtained a vector y from solving the LP, described using a list of non-zero coordinates; the value of y to (Conf-LP) is at least the optimum value minus $\ln(1 + \epsilon)$. We can assume (1) holds with equalities: $\sum_{i \in \mathcal{A}, S \ni j} y_{i,S} = 1$ for every $j \in \mathcal{G}$. Then we let $x_{ij} = \sum_{S \ni j} y_{i,S}$ for every $i \in \mathcal{A}$ and $j \in \mathcal{G}$. So $\sum_{i \in \mathcal{A}} x_{ij} = 1$ for every $j \in \mathcal{G}$.

In this paragraph, we fix an agent i and break the fractional items assigned to i into a set G_i of groups, each containing 1 fractional item. They are created in non-increasing order of values, as in the Shmoys-Tardos algorithm for unrelated machine scheduling problems. That is,

the first group contains the 1 fractional most valuable items assigned to i , the second group contains the 1 fractional most valuable items assigned to i after removing the first group, and so on. Formally, we sort the items in \mathcal{G} in non-increasing order of v_{ij} values, breaking ties arbitrarily. Let $p_i = \lceil \sum_{j \in \mathcal{G}} x_{ij} \rceil$. Then we can find vectors $g^1, g^2, \dots, g^{p_i} \in [0, 1]^{\mathcal{G}}$ satisfying the following properties:

- (P1) For every $t \in [1, p_i - 1]$, we have $|g^t|_1 = 1$; for $t = p_i$, we have $|g^t|_1 = \sum_{j \in \mathcal{G}} x_{ij} - (p_i - 1) \in (0, 1]$.
- (P2) $\sum_{t=1}^{p_i} g_j^t = x_{ij}$ for every $j \in \mathcal{G}$.
- (P3) For every $1 \leq t < t' \leq p_i$, and two items j, j' such that j appears before j' in the ordering, it cannot happen that $g_{j'}^t > 0$ and $g_j^{t'} > 0$.

It is easy to see that g^1, g^2, \dots, g^{p_i} are uniquely determined by the three conditions. We say each g^t is a group. Let $G_i = \{g^1, g^2, \dots, g^{p_i}\}$ be the set of all groups constructed for this agent i .

Now we take all agents i into consideration and let $G = \cup_{i \in \mathcal{A}} G_i$ be the set of all groups constructed.¹ The representations of groups give a fractional matching between the groups G and items \mathcal{G} : an item j is matched to a group $g \in [0, 1]^{\mathcal{G}}$ with a fraction of g_j . Then each item is matched to an extent of 1, and every group g is matched to an extent of $|g|_1$. So a group is matched to an extent of 1 if it is not the last group for an agent, and at most 1 otherwise.

We can efficiently partition the fractional matching between G and \mathcal{G} into a convex combination of polynomial number of (partial-)matchings. If we randomly choose a matching from the convex combination, the following property holds.

- (★) For every group $g \in G$ and item $j \in \mathcal{G}$, we have $\Pr[j \text{ is matched to } g] = g_j$.

Each matching in the combination naturally gives us an allocation of items to agents: If an item $j \in \mathcal{G}$ is matched to some group $g \in G_i$, then we assign j to i . Our algorithm simply outputs the best allocation from all matchings in the combination.

2.3 The Analysis

It is more convenient to analyze the following randomized rounding algorithm: randomly choose a matching from the convex combination, and output the allocation corresponding to the matching. Clearly, the deterministic algorithm can only perform better.

Let S_i be the set of items assigned to i by the randomized rounding algorithm. By (★) we know that the probability that j is assigned to i is precisely x_{ij} . (★) implies that an item $j \in \mathcal{G}$ is matched with probability 1. If a group g has $|g|_1 = 1$, then it is matched with probability 1.

With Theorem 1.3 on the quality of EF1 allocations in the setting of identical agents, we prove the following key lemma:

¹ It is possible that two groups from different sets G_i and $G_{i'}$ have the same vector representation. So we treat G as a multi-set and we assume we know which set G_i each group $g \in G$ belongs to.

LEMMA 2.2. For every $i \in \mathcal{A}$, we have

$$\mathbb{E} \left[\ln v_i(S_i) \right] \geq \sum_{S \subseteq \mathcal{G}} y_{i,S} \cdot \ln v_i(S) - \frac{1}{e}.$$

PROOF. Throughout the proof, we fix the agent i . Let $\Delta > 0$ be an integer, so that every $y_{i,S}$ is an integer multiple of $1/\Delta$, and the probability that $S_i = S$ for any S is also an integer multiple of $1/\Delta$.² We consider an instance of the unweighted Nash Social Welfare problem with identical additive valuations. In the instance, there are Δ copies of the agent i , and Δx_{ij} copies of every item $j \in \mathcal{G}$; so all the agents are identical. The $y = (y_{i,S})_{S \subseteq \mathcal{G}}$ vector gives us an allocation \mathcal{S} to the instance: For every $S \subseteq \mathcal{G}$, there are exactly $\Delta y_{i,S}$ agents who get a copy of S . Notice that this is a valid solution, as $\sum_S y_{i,S} = 1$ and $\sum_{S \ni j} y_{i,S} = x_{ij}$ for every item j .

The Nash Social Welfare of the allocation \mathcal{S} is

$$\left(\prod_{S \subseteq \mathcal{G}} v_i(S)^{\Delta y_{i,S}} \right)^{1/\Delta} = \prod_{S \subseteq \mathcal{G}} v_i(S)^{y_{i,S}}.$$

The distribution for S_i also corresponds to an allocation \mathcal{S}' of items to agents: For every $S \subseteq \mathcal{G}$, there are $\Delta \cdot \Pr[S_i = S]$ agents who get a copy of S . Again, this is a valid solution as $\sum_S \Pr[S_i = S] = 1$ and $\sum_{S \ni j} \Pr[S_i = S] = \mathbb{E}[S_i \ni j] = x_{ij}$.

The Nash Social Welfare of the allocation \mathcal{S}' is

$$\left(\prod_{S \subseteq \mathcal{G}} v_i(S)^{\Delta \Pr[S_i = S]} \right)^{1/\Delta} = \prod_{S \subseteq \mathcal{G}} v_i(S)^{\Pr[S_i = S]}.$$

A crucial property for the solution \mathcal{S}' is that it is EF1. Indeed, if $\Pr[S_i = S] > 0$ for some S , then S contains exactly one item from each group in G_i except for the last one, from which S contains at most one item. Also, the items in the groups G_i are sorted by (P3). So if there are two sets S and S' in the support of the distribution for S_i , and we remove the most valuable item from S' , then S beats S' item by item.

Therefore, by Theorem 1.3, we know that the Nash Social Welfare of \mathcal{S}' is at least $e^{-1/e}$ times that of the optimum allocation for the instance, which is at least that of \mathcal{S} . That is,

$$\prod_{S \subseteq \mathcal{G}} v_i(S)^{\Pr[S_i = S]} \geq e^{-1/e} \cdot \prod_{S \subseteq \mathcal{G}} v_i(S)^{y_{i,S}}.$$

Taking logarithm on both sides gives the lemma. ■

Applying the lemma for every $i \in \mathcal{A}$ and using linearity of expectation, we have

$$\mathbb{E} \left[\sum_{i \in \mathcal{A}} w_i \cdot \ln v_i(S_i) \right] \geq \sum_{i \in \mathcal{A}, S \subseteq \mathcal{G}} w_i \cdot y_{i,S} \cdot \ln v_i(S) - \frac{1}{e}.$$

2 We have that all $y_{i,S}$ values are rational numbers. Under this condition, it is easy to guarantee that the probabilities are rational numbers.

We used that $\sum_{i \in \mathcal{A}} w_i = 1$. By the convexity of exponential function, we have

$$\mathbb{E} \left[\prod_{i \in \mathcal{A}} v_i(S_i)^{w_i} \right] \geq e^{-1/e} \cdot \exp \left(\sum_{i \in \mathcal{A}, S \subseteq \mathcal{G}} w_i \cdot y_{i,S} \cdot \ln v_i(S_i) \right) \geq e^{-1/e} \cdot \frac{\text{opt}}{1 + \epsilon},$$

where opt is the weighted Nash Social Welfare of the optimum allocation, and the second inequality used that the value of our solution y to (Conf-LP) is at least its optimum value minus $\ln(1 + \epsilon)$. By scaling ϵ down by an absolute constant at the beginning, we can make the right side to be at least $\frac{\text{opt}}{e^{1/e} + \epsilon}$. As we argued, the deterministic algorithm will output an allocation whose weighted Nash Social Welfare is at least $\frac{\text{opt}}{e^{1/e} + \epsilon}$. This finishes the proof of Theorem 1.1.

3. Solving Configuration LP within an Additive Error of $\ln(1 + \epsilon)$: Proof of Theorem 2.1

In this section, we prove Theorem 2.1. By scaling valuation functions, we assume $v_{ij} = 0$ or $v_{ij} \geq 1$ for every $i \in \mathcal{A}, j \in \mathcal{G}$; this does not change the instance. Let lp be the value of the (Conf-LP). As we assumed there is an allocation with positive value, we have $\text{lp} \neq -\infty$.

We assume $\epsilon > 0$ is upper bounded by a sufficiently small constant; otherwise, we take ϵ to be the constant. We say an algorithm is efficient if its running time is polynomial in the size of the input instance and $\frac{1}{\epsilon}$. We assume that we are given a number $o < \text{lp}$. Our goal is to find a valid solution to (Conf-LP) with rational coordinates, whose value is at least $o - \frac{\epsilon}{4} - \ln(1 + \frac{\epsilon}{4})$. At the end of the section, we show that this is sufficient.

We consider the dual of (Conf-LP), with the objective replaced by a constraint.

$$\sum_{j \in \mathcal{G}} \alpha_j + \sum_{i \in \mathcal{A}} \beta_i \leq o \tag{4}$$

$$\sum_{j \in S} \alpha_j + \beta_i \geq w_i \cdot \ln v_i(S) \quad \forall i \in \mathcal{A}, S \subseteq \mathcal{G} \tag{5}$$

$$\alpha_j \geq 0 \quad \forall j \in \mathcal{G} \tag{6}$$

As we assumed $o < \text{lp}$, the dual LP (4-6) is infeasible. Abusing the terminology slightly, we say a vector $(\alpha \in \mathbb{R}_{\geq 0}^{\mathcal{G}}, \beta \in \mathbb{R}^{\mathcal{A}})$ is an optimum solution to the dual LP if it minimizes the left-side of (4), subject to (5) and (6). By linear programming duality, an optimum solution (α, β) has $\sum_{j \in \mathcal{G}} \alpha_j + \sum_{i \in \mathcal{A}} \beta_i = \text{lp}$.

To check the feasibility of dual LP, we need an approximate separation oracle for (5). This can be done using dynamic programming:

LEMMA 3.1. Given $(\alpha \in \mathbb{R}_{\geq 0}^{\mathcal{G}}, \beta \in \mathbb{R}^{\mathcal{A}})$ that does not satisfy (5), there is an efficient oracle that finds some $i \in \mathcal{A}$ and $S' \subseteq \mathcal{G}$ satisfying

$$\sum_{j \in S'} \alpha_j + \beta_i < w_i \cdot \ln \left(\left(1 + \frac{\epsilon}{2}\right) v_i(S') \right). \quad (7)$$

PROOF. Suppose (5) is not satisfied for $i \in \mathcal{A}$ and $S \subseteq \mathcal{G}$. We can guess the agent i , and the item $j^* \in S$ with the largest v_{ij^*} . Then we discard the items j with $v_{ij} > v_{ij^*}$. Let \mathcal{G}' be the set of remaining items. For every $j \in \mathcal{G}'$, we round down v_{ij} to the nearest integer multiple of $\frac{\epsilon \cdot v_{ij^*}}{2m}$; let \bar{v}_{ij} be the rounded value. As \bar{v}_{ij} values are integer multiples of $\frac{\epsilon \cdot v_{ij^*}}{2m}$ not exceeding v_{ij^*} , we can afford to guess $\bar{V} := \bar{v}_i(S)$ as it has only $O(\frac{m^2}{\epsilon})$ possibilities.

We solve the following knapsack covering problem: find a set $S' \subseteq \mathcal{G}'$ such that $\bar{v}_i(S') \geq \bar{V}$ so as to minimize $\sum_{j \in S'} \alpha_j$. As the \bar{v}_{ij} values are of the form $\frac{z \cdot \epsilon \cdot v_{ij^*}}{2m}$ with $z \in \left[0, \frac{2m}{\epsilon}\right]$ being integers, the problem can be solved efficiently and exactly using dynamic programming. As the set S is a valid solution to the knapsack covering instance, the solution S' returned satisfies $\bar{v}_i(S') \geq \bar{V} = \bar{v}_i(S)$ and $\sum_{j \in S'} \alpha_j \leq \sum_{j \in S} \alpha_j$. So,

$$v_i(S') \geq \bar{v}_i(S') \geq \bar{v}_i(S) \geq v_i(S) - m \cdot \frac{\epsilon \cdot v_{ij^*}}{2m} = v_i(S) - \frac{\epsilon \cdot v_{ij^*}}{2} \geq v_i(S) - \frac{\epsilon}{2} \cdot v_i(S').$$

Therefore $v_i(S) \leq \left(1 + \frac{\epsilon}{2}\right) v_i(S')$. Then (7) follows from that (5) is not satisfied for i, S .

We need to guess i, j^* and \bar{V} ; there are $O(\frac{nm^3}{\epsilon})$ possibilities for the combination. For a fixed i, j^* and \bar{V} , the dynamic programming runs in $O(\frac{m^3}{\epsilon})$ -time. So, overall, the running time of the oracle is $\text{poly}(n, m, \frac{1}{\epsilon})$.³ Finally as (7) can be verified easily for any given i and S' , incorrect guesses will not cause any issue. ■

To run the ellipsoid method, we need some bounds on α_j and β_i values. This is not straightforward as the β_i values can be negative. We define $v_{\max} := \max_{i \in \mathcal{A}, j \in \mathcal{G}} v_{ij}$.

LEMMA 3.2. There is an optimum solution $(\alpha \in \mathbb{R}_{\geq 0}^{\mathcal{G}}, \beta \in \mathbb{R}^{\mathcal{A}})$ to the dual LP such that $\alpha_j \in [0, \ln(mv_{\max}^2)]$ for every $j \in \mathcal{G}$ and $|\beta_i| \leq \ln(mv_{\max}^2)$ for every $i \in \mathcal{A}$.

PROOF. We modify constraint (5) in the dual LP (4-6), so that we only consider the constraint for pairs $(i \in \mathcal{A}, S \subseteq \mathcal{G})$ satisfying $v_{ij} > 0$ for every $j \in S$. This does not change the dual LP as constraint (5) for a pair (i, S) is implied by the constraint for the pair $(i, \{j \in S : v_{ij} > 0\})$. Let $(\alpha \in \mathbb{R}_{\geq 0}^{\mathcal{G}}, \beta \in \mathbb{R}^{\mathcal{A}})$ an optimum solution to the dual LP with the smallest $\sum_{j \in \mathcal{G}} \alpha_j$; notice that $\sum_{j \in \mathcal{G}} \alpha_j + \sum_{i \in \mathcal{A}} \beta_i = \text{lp}$. Let Π be the set of (i, S) pairs for which (5) is tight for the given (α, β) , after the modification to constraint (5).

We argue that if $\beta_i < 0$ and $|S| \geq 2$, then $(i, S) \notin \Pi$. Assume towards the contradiction that some $(i, S) \in \Pi$ has $\beta_i < 0$ and $|S| \geq 2$.

3 We remark that we did not try to optimize the running time.

Every $j \in S$ has $v_{ij} \geq 1$ by the assumption made at the beginning of the section. We have $\alpha_j + \beta_i \geq w_i \cdot \ln v_{ij}$ for every $j \in S$ as (5) holds for $(i, \{j\})$. Summing up the inequalities over all $j \in S$ gives us $\sum_{j \in S} \alpha_j + |S|\beta_i \geq w_i \cdot \sum_{j \in S} \ln v_{ij} \geq w_i \ln v_i(S)$. Therefore $\sum_{j \in S} \alpha_j + \beta_i > w_i \ln v_i(S)$ as $|S| \geq 2$ and $\beta_i < 0$, a contradiction.

Let \mathcal{A}' be the set of agents i with $\beta_i < 0$. Let $E = \{(i, j) : i \in \mathcal{A}', (i, \{j\}) \in \Pi\}$. We prove that there is a matching in the bipartite graph $H := (\mathcal{A}' \cup \mathcal{G}, E)$ covering all agents \mathcal{A}' . Otherwise, there is a set $\mathcal{A}'' \subseteq \mathcal{A}'$ of agents with $|N_H(\mathcal{A}'')| < |\mathcal{A}''|$, where $N_H(\mathcal{A}'')$ is the set of neighbors of \mathcal{A}'' in H . Then consider the following operation: decreasing β_i for every $i \in \mathcal{A}''$ by δ and increasing α_j for every $j \in N_H(\mathcal{A}'')$ by δ , for a small enough $\delta > 0$. This decreases $\sum_{j \in \mathcal{G}} \alpha_j + \sum_{i \in \mathcal{A}} \beta_i$ as $|C \cap \mathcal{G}| \geq |C \cap \mathcal{A}|$ without violating (5) and (6), contradicting that (α, β) is optimum.

Focus on any connected component C of H . We prove that some $j \in C \cap \mathcal{G}$ has $\alpha_j \leq \ln(mv_{\max})$. Assume this is not the case; that is, every $j \in C \cap \mathcal{G}$ has $\alpha_j > \ln(mv_{\max})$. Consider the following operation: decreasing α_j by δ for every $j \in C \cap \mathcal{G}$ and increasing β_i by δ for every $i \in C \cap \mathcal{A}$, for a small enough $\delta > 0$. This operation does not break constraint (5) for any i with $\beta_i \geq 0$, as the right side of (5) is at most $\ln(mv_{\max})$. It does not break the constraint for any i with $\beta_i < 0$ as δ is sufficiently small. Moreover, the operation decreases $\sum_{j \in \mathcal{G}} \alpha_j$ without increasing $\sum_{j \in \mathcal{G}} \alpha_j + \sum_{i \in \mathcal{A}} \beta_i$ as $|C \cap \mathcal{G}| \geq |C \cap \mathcal{A}|$, contradicting the way we choose (α, β) .

Therefore, every connected component C of H contains some $j \in C \cap \mathcal{G}$ with $\alpha_j \leq \ln(mv_{\max})$. Recall that every $(i, j) \in E$ has $\beta_i < 0$ and $\beta_i + \alpha_j = w_i \cdot \ln v_{ij}$. This is equivalent to $\alpha_j = -\beta_i + w_i \cdot \ln v_{ij}$. There is a simple path in H connecting every vertex $\mathcal{A}' \cup \mathcal{G}$ to an item j with $\alpha_j \leq \ln(mv_{\max})$. So, α_j for every $j \in \mathcal{G}$ (respectively, $-\beta_i$ for every $i \in \mathcal{A}'$) is at most $\ln(mv_{\max}) + \sum_{i \in \mathcal{A}'} w_i \ln v_{\max} \leq \ln(mv_{\max}) + \ln v_{\max} = \ln(mv_{\max}^2)$. Finally, for $i \in \mathcal{A} \setminus \mathcal{A}'$, we have $\beta_i \geq 0$. Clearly we have $\beta_i \leq w_i \ln(mv_{\max}) \leq \ln(mv_{\max})$ since other decreasing β_i to $w_i \ln(mv_{\max})$ will not break any constraint, violating the optimality of (α, β) . ■

With the lemma, we start the ellipsoid method with the smallest ellipsoid containing the cuboid $[0, mv_{\max}^2]^{\mathcal{G}} \times [-mv_{\max}^2, mv_{\max}^2]^{\mathcal{A}}$. We terminate the procedure when the volume of the ellipsoid is less than that of the cuboid $\left[0, \frac{\epsilon}{4(n+m)}\right]^{n+m}$. The algorithm terminates in $\text{poly}(n, m, \log \frac{1}{\epsilon}, \log v_{\max})$ iterations and is thus efficient.

We claim the following LP is infeasible:

$$\sum_{j \in \mathcal{G}} \alpha_j + \sum_{i \in \mathcal{A}} \beta_i \leq o - \frac{\epsilon}{4} \quad (8)$$

$$\sum_{j \in S'} \alpha_j + \beta_i \geq w_i \cdot \ln \left(\left(1 + \frac{\epsilon}{2}\right) v_i(S') \right) \quad \forall (i, S') \text{ used by the ellipsoid method} \quad (9)$$

$$\alpha_j \geq 0 \quad \forall j \in \mathcal{G} \quad (10)$$

Otherwise, assume some (α, β) is a feasible solution to the above LP. Then the cuboid $\left\{(\alpha, \beta) + \delta : \delta \in \left[0, \frac{\epsilon}{4(n+m)}\right]^{\mathcal{G} \cup \mathcal{A}}\right\}$ would be still contained in the final ellipsoid, contradicting our termination condition.

We consider the weighted Nash Social Welfare instance where all v_{ij} values are scaled up by $1 + \frac{\epsilon}{2}$, and (Conf-LP) to the instance. By solving the LP restricted to the variables $y_{i,S'}$ corresponding to constraints (9) (that is, we fix the other variables to 0), we obtain a solution y whose value is at least $o - \frac{\epsilon}{4}$ w.r.t the scaled instance. The LP solver runs in polynomial time as the number of variables $y_{i,S'}$ now is polynomial. So, the value of the solution y to (Conf-LP) w.r.t the original instance is at least $o - \frac{\epsilon}{4} - \sum_{i \in \mathcal{A}, S \subseteq \mathcal{G}} y_{i,S} w_i \ln\left(1 + \frac{\epsilon}{4}\right) = o - \sum_i w_i \ln\left(1 + \frac{\epsilon}{4}\right) = o - \frac{\epsilon}{4} - \ln\left(1 + \frac{\epsilon}{4}\right)$. Moreover, if y is a vertex solution, then it is rational as all the coefficients in the LP constraints are in $\{0, 1\}$.

Finally, we show how to handle the assumption that we are given the parameter o . By only allowing every agent to get one item, we can obtain an m -approximation for the weighted Nash Social Welfare instance. Therefore, we can construct a list of $O\left(\frac{\log m}{\epsilon}\right)$ values of o , such that for some value o in the list, we have $lp \in (o, o + \frac{\epsilon}{4}]$. For every o in the list, we run the above algorithm and return the best solution constructed. In particular, for the o with $lp \in (o, o + \frac{\epsilon}{4}]$, the solution has value at least $o - \frac{\epsilon}{4} - \ln\left(1 + \frac{\epsilon}{4}\right) \geq lp - \frac{\epsilon}{4} - \frac{\epsilon}{4} - \ln\left(1 + \frac{\epsilon}{4}\right) \geq lp - \ln(1 + \epsilon)$ for a sufficiently small ϵ . Therefore, the best solution returned has value at least $lp - \ln(1 + \epsilon)$; this finishes the proof of Theorem 2.1.

References

- [1] Nima Anari, Shayan Oveis Gharan, Amin Saberi, and Mohit Singh. Nash Social Welfare, Matrix Permanent, and Stable Polynomials. *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*, volume 67 of *LIPIcs*, 36:1–36:12, 2017. [DOI](#) (2, 3)
- [2] Julius B. Barbanel and Alan D. Taylor. The geometry of efficient fair division. Cambridge Univ. Press, 2005. [DOI](#) (2)
- [3] Siddharth Barman, Sanath Kumar Krishnamurthy, and Rohit Vaish. Finding Fair and Efficient Allocations. *Proceedings of the 2018 ACM Conference on Economics and Computation, EC '18*, pages 557–574, 2018. [DOI](#) (1–4)
- [4] Steven J. Brams and Alan D. Taylor. Fair Division: From Cake-Cutting to Dispute Resolution. Cambridge University Press, 1996. [DOI](#) (2)
- [5] Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia. Handbook of Computational Social Choice. Cambridge University Press, 1st edition, 2016. [DOI](#) (2)
- [6] Adam Brown, Aditi Laddha, Madhusudhan Reddy Pittu, and Mohit Singh. Approximation algorithms for the weighted Nash social welfare via convex and non-convex programs. *Proceedings of the 2024 ACM-SIAM Symposium on Discrete Algorithms, SODA 2024*, pages 1307–1327, 2024. [DOI](#) (2, 3)
- [7] Suchan Chae and Hervé Moulin. Bargaining among groups: an axiomatic viewpoint. *Int. J. Game Theory*, 39(1-2):71–88, 2010. [DOI](#) (2)
- [8] Richard Cole, Nikhil Devanur, Vasilis Gkatzelis, Kamal Jain, Tung Mai, Vijay V. Vazirani, and Sadra Yazdanbod. Convex Program Duality, Fisher Markets, and Nash Social Welfare. *Proceedings of the 2017 ACM Conference on Economics and Computation, EC '17*, pages 459–460, 2017. [DOI](#) (2)
- [9] Richard Cole and Vasilis Gkatzelis. Approximating the Nash social welfare with indivisible items. *SIAM Journal on Computing*, 47(3):1211–1236, 2018. [DOI](#) (2)

- [10] Dagmawi Mulugeta Degefu, Weijun He, Liang Yuan, An Min, and Qi Zhang. Bankruptcy to Surplus: Sharing Transboundary River Basin's Water under Scarcity. *Water Resources Management: An International Journal, Published for the European Water Resources Association (EWRA)*, 32(8):2735–2751, 2018. DOI (2)
- [11] Shahar Dobzinski, Wenzheng Li, Aviad Rubinstein, and Jan Vondrák. A constant-factor approximation for Nash social welfare with subadditive valuations. *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024*, pages 467–478, 2024. DOI (3)
- [12] Yuda Feng, Yang Hu, Shi Li, and Ruilong Zhang. Constant approximation for weighted Nash social welfare with submodular valuations. *Proceedings of the 57th Annual ACM Symposium on Theory of Computing, STOC 2025*, pages 1395–1405, 2025. DOI (3)
- [13] Yuda Feng and Shi Li. A note on approximating weighted Nash social welfare with additive valuations. *51st International Colloquium on Automata, Languages, and Programming, ICALP 2024*, volume 297 of *LIPIcs*, 63:1–63:9, 2024. DOI (1)
- [14] Jugal Garg, Martin Hoefer, and Kurt Mehlhorn. Satiation in fisher markets and approximation of Nash social welfare. *Mathematics of Operations Research*, 49, 2023. DOI (2)
- [15] Jugal Garg, Edin Husić, Wenzheng Li, László A. Végh, and Jan Vondrák. Approximating Nash Social Welfare by Matching and Local Search. *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023*, pages 1298–1310, 2023. DOI (3)
- [16] Jugal Garg, Edin Husić, and László A. Végh. Approximating Nash Social Welfare under Rado Valuations. *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2021*, pages 1412–1425, 2021. DOI (3)
- [17] Jugal Garg, Pooja Kulkarni, and Rucha Kulkarni. Approximating Nash Social Welfare under Submodular Valuations through (Un)Matchings. *ACM Trans. Algorithms*, 19(4), 2023. DOI (3)
- [18] John C. Harsanyi and Reinhard Selten. A generalized Nash solution for two-person bargaining games with incomplete information. *Management Science*, 18(5):P80–P106, 1972. DOI (2)
- [19] Harold Houba, Gerard van der Laan, and Yuyu Zeng. Asymmetric Nash Solutions in the River Sharing Problem. *Game Theory & Bargaining Theory eJournal*, 2013. DOI (2)
- [20] Ehud Kalai. Nonsymmetric Nash solutions and replication of 2-person bargaining. *International Journal of Games Theory*, 6:129–133, January 1977. DOI (2)
- [21] Mamoru Kaneko and Kenjiro Nakamura. The Nash social welfare function. *Econometrica*, 47(2):423–435, 1979. DOI (2)
- [22] Frank Kelly. Charging and Rate Control for Elastic Traffic. *European Transactions on Telecommunications*, 8(1):33–37, 1997. DOI (2)
- [23] Annick Laruelle and Federico Valenciano. Bargaining in Committees as An Extension of Nash's Bargaining Theory. *Journal of Economic Theory*, 132(1):291–305, 2007. DOI (2)
- [24] Euiwoong Lee. Apx-hardness of maximizing Nash social welfare with indivisible items. *Information Processing Letters*, 122:17–20, 2017. DOI (2)
- [25] Wenzheng Li and Jan Vondrák. A Constant-Factor Approximation Algorithm for Nash Social Welfare with Submodular Valuations. *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 25–36, 2022. DOI (3)
- [26] Wenzheng Li and Jan Vondrák. Estimating the Nash social welfare for coverage and other submodular valuations. *Proceedings of the Thirty-Second Annual ACM-SIAM Symposium on Discrete Algorithms. SODA '21*. 2021., pages 1119–1130. DOI (3)
- [27] Herve Moulin. Fair Division and Collective Welfare. The MIT Press, 2004. DOI (2)
- [28] John F. Nash. The Bargaining Problem. *Econometrica*, 18(2):155–162, 1950. DOI (2)
- [29] Nhan-Tam Nguyen, Trung Thanh Nguyen, Magnus Roos, and Jörg Rothe. Complexity and approximability of social welfare optimization in multiagent resource allocation. *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 3, AAMAS '12*, pages 1287–1288, 2012. DOI (2)
- [30] Jack Robertson and William Webb. Cake-Cutting Algorithms: Be Fair if You Can. A K Peters/CRC Press, 1998. DOI (2)
- [31] Jörg Rothe, editor. Economics and Computation, An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division. Springer texts in business and economics. Springer, 2016. DOI (2)
- [32] David B. Shmoyes and Éva Tardos. An approximation algorithm for the generalized assignment problem. *Math. Program.* 62:461–474, 1993. DOI (1, 4)
- [33] William Thomson. Replication Invariance of Bargaining Solutions. *Int. J. Game Theory*, 15(1):59–63, 1986. DOI (2)
- [34] Hal R Varian. Equity, Envy, and Efficiency. *Journal of Economic Theory*, 9(1):63–91, 1974. DOI (2)
- [35] H. Peyton Young. Equity: In Theory and Practice. Princeton University Press, 1994. DOI (2)
- [36] S. Yu, E. C. van Ierland, H. -P. Weikard, and X. Zhu. Nash Bargaining Solutions for International Climate Agreements under Different Sets of Bargaining Weights. *International Environmental Agreements: Politics, Law and Economics*, 17(5):709–729, 2017. DOI (2)