

Approximating Nash Social Welfare by Matching and Local Search

Jugal Garg¹, Edin Husic², Wenzheng Li³, László A. Végh⁴, and Jan Vondrák³

¹University of Illinois at Urbana-Champaign, IL

²IDSIA, USI-SUPSI, Switzerland

³Stanford University, CA

⁴London School of Economics, UK

Abstract

For any $\varepsilon > 0$, we give a simple, deterministic $(4 + \varepsilon)$ -approximation algorithm for the Nash social welfare (NSW) problem under submodular valuations. We also consider the asymmetric variant of the problem, where the objective is to maximize the weighted geometric mean of agents' valuations, and give an $(\omega + 2 + \varepsilon)e$ -approximation if the ratio between the largest weight and the average weight is at most ω .

We also show that the $1/2$ -EFX envy-freeness property can be attained simultaneously with a constant-factor approximation. More precisely, we can find an allocation in polynomial time that is both $1/2$ -EFX and a $(8 + \varepsilon)$ -approximation to the symmetric NSW problem under submodular valuations.

1 Introduction

We consider the problem of allocating a set G of m indivisible items among a set A of n agents, where each agent $i \in A$ has a valuation function $v_i : 2^G \rightarrow \mathbb{R}_{\geq 0}$ and a weight (entitlement) $w_i > 0$ such that $\sum_{i \in A} w_i = 1$. The Nash social welfare (NSW) problem asks for an allocation $\mathcal{S} = (S_i)_{i \in A}$ that maximizes the weighted geometric mean of the agents' valuations,

$$\text{NSW}(\mathcal{S}) = \prod_{i \in A} v_i(S_i)^{w_i}.$$

We refer to the special case when all agents have equal weight (i.e., $w_i = 1/n$) as the *symmetric* NSW problem, and call the general case the *asymmetric* NSW problem. Throughout, we let $w_{\max} := \max_{i \in A} w_i$. For $\alpha > 1$, an α -*approximate solution* to the NSW problem is an allocation \mathcal{S} with $\text{NSW}(\mathcal{S}) \geq \text{OPT}/\alpha$, where OPT denotes the optimum value of the NSW-maximization problem.

Allocating resources among agents in a fair and efficient manner is a fundamental problem in computer science, economics, and social choice theory; we refer the reader to the monographs [5, 11, 12, 47, 52, 53, 55] on the background. A common measure of efficiency is *utilitarian social welfare*,

*JV and WL were supported by NSF Award 2127781. LV received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement no. ScaleOpt-757481). EH was supported by SNSF Grant 200021 200731/1. JG was supported by NSF Grants CCF-1942321 and CCF-2334461.

i.e., the sum of the utilities $\sum_{i \in A} v_i(S_i)$ for an allocation $(S_i)_{i \in A}$. In contrast, fairness is often measured by *max-min fairness*, i.e., $\min_{i \in A} v_i(S_i)$; maximizing this objective is also known as the *Santa Claus problem* [4].

Symmetric NSW provides a balanced tradeoff between the often conflicting requirements of fairness and efficiency. It has been introduced independently in a variety of contexts. It is a discrete analogue of the Nash bargaining game [39, 48]; it corresponds to the notion of competitive equilibrium with equal incomes in economics [54]; and arises as a proportional fairness notion in networking [40]. The more general asymmetric objective has also been well-studied since the seventies [37, 38]. It has found applications in different areas, such as bargaining theory [17, 41], water resource allocation [24, 36], and climate agreements [56].

A distinctive feature of the NSW problem is invariance under scaling of the valuation functions v_i by independent factors λ_i , i.e., each agent can express their preference in a “different currency” without changing the optimization problem (see [47] for additional characteristics).

1/2-EFX allocations Envy-freeness up to any item (EFX) is considered the most compelling fairness notion in the discrete setting with equal entitlements [16], where an allocation $\mathcal{S} = (S_i)_{i \in A}$ is said to be EFX if

$$v_i(S_i) \geq v_i(S_k \setminus \{j\}), \quad \forall i, k \in A, \forall j \in S_k.$$

That is, no agent envies another agent’s bundle after the removal of any single item from that agent’s bundle. It is not known whether EFX allocations always exists or not, and it is regarded as “fair division’s biggest open question” [51]. This motivated the study of its relaxation α -EFX for an $\alpha \in (0, 1)$, where an allocation \mathcal{S} is said to be α -EFX if

$$v_i(S_i) \geq \alpha \cdot v_i(S_k \setminus \{j\}), \quad \forall i, k \in A, \forall j \in S_k.$$

The best-known α , for which the existence is known, is $1/2$ for submodular valuations, albeit with the efficiency guarantee of $O(n)$ -approximation to the symmetric NSW problem [20, 50].

Submodular and subadditive valuation functions A set function $v : 2^G \rightarrow \mathbb{R}$ is *monotone* if $v(S) \leq v(T)$ whenever $S \subseteq T$. A monotone set function with $v(\emptyset) = 0$ is also called a *valuation function* or simply *valuation*. The function $v : 2^G \rightarrow \mathbb{R}$ is *submodular* if

$$v(S) + v(T) \geq v(S \cap T) + v(S \cup T) \quad \forall S, T \subseteq G,$$

and *subadditive* if

$$v(S) + v(T) \geq v(S \cup T) \quad \forall S, T \subseteq G.$$

We assume the valuation functions are given by value oracles that return $v(S)$ for any $S \subseteq G$ in $O(1)$ time.

Our contributions Our main theorem on NSW is the following.

Theorem 1.1. *For any $\varepsilon > 0$, there is a deterministic polynomial-time $(nw_{\max} + 2 + \varepsilon)$ -approximation algorithm for the asymmetric Nash social welfare problem with submodular valuations. For symmetric instances, the algorithm returns a $(4 + \varepsilon)$ -approximation. The number of arithmetic operations and value oracle calls is polynomial in n , m , and $1/\varepsilon$.*

Algorithm 1 in Section 2 presents the algorithm asserted in the theorem. Note that nw_{\max} is the ratio between the maximum weight w_{\max} and the average weight $(1/n)$. In the symmetric case, when all weights are $w_i = 1/n$, this bound gives $(3 + \varepsilon)e < 8.2$. In this case, we can improve the analysis to obtain a $(4 + \varepsilon)$ -approximation algorithm.

As our second main result, we show that a $1/2$ -EFX allocation with high NSW value exists and can also be efficiently found. We give a general reduction for subadditive valuations. An allocation $\mathcal{S} = (S_i)_{i \in A}$ is called *complete* if every item is included, i.e., $\cup_{i \in A} S_i = G$; otherwise, it is called *partial*. In the context of $1/2$ -EFX allocations, $\text{NSW}(\mathcal{S})$ will always refer to the NSW value of allocation \mathcal{S} in the symmetric case ($w_i = 1/n$ for all $i \in A$).

Theorem 1.2. *There is a deterministic strongly polynomial-time algorithm that given a symmetric NSW instance with subadditive valuations and given a (complete or partial) allocation \mathcal{S} of the items, it returns a complete allocation \mathcal{T} that is $1/2$ -EFX and $\text{NSW}(\mathcal{T}) \geq \text{NSW}(\mathcal{S})/2$.*

The above algorithm is strongly polynomial in the value oracle model: number of basic arithmetic operations and oracle calls is polynomially bounded in n and m . Together with Theorem 1.1, we obtain the following corollary.

Corollary 1.3. *For any $\varepsilon > 0$, there is a deterministic polynomial algorithm that returns a $1/2$ -EFX complete allocation that is $(8 + \varepsilon)$ -approximation to the symmetric NSW problem under submodular valuations. The number of arithmetic operations and value oracle calls is polynomial in n , m , and $1/\varepsilon$.*

Finally, we show that the local search techniques used in proving Theorem 1.1, combined with a guessing framework, can lead to a fixed parameter tractability result. The approximation factor is independent of the weights w_i , however, the running time is exponential in n .

Theorem 1.4. *For any $\varepsilon > 0$, there is a deterministic $(6\varepsilon + \varepsilon)$ -approximation algorithm for the asymmetric Nash social welfare problem with submodular valuations, where the number of arithmetic operations and value oracle calls is $2^{O(n \log n)} \text{poly}(m, 1/\varepsilon)$.*

1.1 Related work

Prior work on approximating NSW Let us first consider *additive valuations*, i.e., when $v_i(S) = \sum_{j \in S} v_{ij}$ for nonnegative values v_{ij} . Maximizing symmetric NSW is NP-hard already in the case of two agents with identical additive valuations, by a reduction from the Subset-Sum problem. It is NP-hard to approximate within a factor better than 1.069 for additive valuations [32], and better than 1.5819 for submodular valuations [35].

On the positive side, a number of remarkably different constant-factor approximations are known for additive valuations. The first such algorithm with the factor of $2 \cdot e^{1/e} \approx 2.889$ was given by Cole and Gkatzelis [23] using a continuous relaxation based on a particular market equilibrium concept. Later, [22] improved the analysis of this algorithm to achieve the factor of 2. Anari, Oveis Gharan, Saberi, and Singh [2] used a convex relaxation that relies on properties of real stable polynomials. The current best factor is $e^{1/e} + \varepsilon \simeq 1.45$ by Barman, Krishnamurthy, and Vaish [8]; the algorithm uses a different market equilibrium based approach and also guarantees EF1 fairness, a weaker version of the EFX property.

For the general class of subadditive valuations [6, 20, 35], $O(n)$ -approximations are known. This is the best one can hope for in the value oracle model [6], for the same reasons that this is impossible for the utilitarian social welfare problem [26]. Sublinear approximation $O(n^{53/54})$ is possible for XOS

valuations if we are given access to both demand and XOS oracles [7]. Recall that all submodular valuations are XOS, and all XOS valuations are subadditive.

Constant-factor approximations were also obtained beyond additive valuation functions: capped-additive [31], separable piecewise-linear concave (SPLC) [3], and their common generalization, capped-SPLC [18] valuations; the approximation factor for capped-SPLC valuations matches the $e^{1/e} + \varepsilon$ factor for additive valuations. All these valuations are special classes of submodular. Subsequently, Li and Vondrák [43] designed an algorithm that estimates the optimal value within a factor of $\frac{e^3}{(e-1)^2} \simeq 6.8$ for a broad class of submodular valuations, such as coverage and summations of matroid rank functions, by extending the techniques of [2] using real stable polynomials. However, this algorithm only estimates the optimum value but does not find a corresponding allocation in polynomial time.

In [34], Garg, Husić, and Végő developed a constant-factor approximation for a broader subclass of submodular valuations called *Rado-valuations*. These include weighted matroid rank functions and many others that can be obtained using operations such as induction by network and contractions. An important example outside this class is the coverage valuation. They attained an approximation ratio 772 for the symmetric case and $772(w_{\max}/w_{\min})^3$ for the asymmetric case. Most recently, Li and Vondrák [44] obtained a randomized 380-approximation for symmetric NSW under submodular valuations by extending the the approach of [34].

We significantly improve and simplify the approach used in [34] and [44]; we give a comparison to these works in Section 2.5.

Subsequent work on approximating NSW Subsequent to the conference version of this work [33], Dobzinski, Li, Rubinfeld, and Vondrák [25] obtained a constant factor approximation for symmetric NSW for subadditive valuations, assuming a demand oracle. They extend the approach of [34] and [44], using also a configuration LP rounding.

Further, Brown, Laddha, Pittu, and Singh [13] obtained an approximation factor of $5 \exp(2 \log n + 2 \sum_{i \in A} w_i \log w_i)$ for additive valuations. The term in $\exp(\cdot)$ is twice the Bregman divergence between the distribution $(w_i)_{i \in A}$ and the uniform distribution. As pointed out by Bento Natura [49], a simple change in our analysis leads to the improved bound $\exp(\log(3n) + \sum_{i \in A} w_i \log w_i) \cdot (e + \varepsilon)$ in Theorem 1.1 for general submodular valuations; this is explained in Remark 2.15. Very recently, Feng and Li [29] obtained an $e^{1/e} + \varepsilon$ -approximation for general asymmetric NSW with additive valuations using a novel rounding of the configuration LP. However, the general asymmetric case with submodular valuations remains open.

Prior work on EFX and related notions The existence of EFX allocations has not been settled despite significant efforts [16, 19, 50, 51]. This problem is open for more than two agents with general monotone valuations (including submodular), and for more than three agents with additive valuations. This necessitated the study of its relaxations α -EFX for $\alpha \in (0, 1)$ and partial EFX allocations. For the notion of α -EFX, the best-known α is 0.618 for additive [1] and 0.5 for subadditive valuations (including submodular) [50].

For the notion of partial EFX allocations, the existence is known for general monotone valuations if we do not allocate at most $n - 2$ items [10, 21, 46], albeit without any efficiency guarantees. For additive valuations, although $n - 2$ is still the best bound known, there exist partial EFX allocations with 2-approximation to the NSW problem [15].

A well-studied weaker notion is envy-freeness up to one item (EF1), where no agent envies another agent after the removal of *some* item from the envied agent’s bundle. EF1 allocations are known to exist for general monotone valuations and can also be computed in polynomial-time [45].

However, an EF1 allocation alone is not desirable because it might be highly inefficient in terms of any welfare objective. For additive valuations, the allocations maximizing NSW are EF1 [16]. Although the NSW problem is APX-hard [42], there exists a pseudopolynomial time algorithm to find an allocation that is EF1 and 1.45-approximation to the NSW problem under additive valuations [8]. For capped-SPLC valuations, [18] shows the existence of an allocation that is $1/2$ -EF1 and 1.45-approximation to the NSW problem.

Subsequent to the conference version of this work, [28] improved Theorem 1.2 to show the existence of an allocation \mathcal{T} that is $1/2$ -EFX and $\text{NSW}(\mathcal{T}) \geq 2/3 \text{NSW}(\mathcal{S})$ for a given allocation \mathcal{S} . For subadditive valuations, [9] recently showed the existence of an allocation that is EF1 and provides a 2-approximation to the NSW problem.

Notation We also use monotone set functions with $v(\emptyset) > 0$; we refer to these as *endowed valuation functions*. We use $\log(x)$ for the natural logarithm throughout. For set $S \subseteq G$ and $j \in G$, we use $S + j$ to denote $S \cup \{j\}$ and $S - j$ for $S \setminus \{j\}$ and we write $v(j)$ for $v(\{j\})$. For a vector $p \in \mathbb{R}^G$ and $S \subseteq G$, we denote $p(S) = \sum_{i \in S} p_i$.

By a *matching* from A to G we mean a mapping $\tau : A \rightarrow G \cup \{\perp\}$ where $\tau(i) \neq \tau(j)$ if $\tau(i) \neq \perp$; \perp is a special symbol representing unmatched agents. A matching is *A-perfect*, if $\tau(i) \neq \perp$ for every $i \in A$. An *A-perfect matching* is simply a perfect matching when every node on both sides is matched, which is possible when $|A| = |G|$.

Overview of the paper Section 2 describes the NSW approximation algorithm. A comparison of our technique with previous approaches is given in Section 2.5. The exposition of the fairness guarantee is given in Section 3, and the FPT algorithm is analyzed in Section 4. We conclude the paper with some remarks in Section 5.

2 Approximation algorithm for Nash social welfare

The NSW algorithm asserted in Theorem 1.1 is shown in Algorithm 1. The subroutine `LocalSearch` will be described in Algorithm 2. The algorithm proceeds in the following three phases.

Algorithm 1: Approximating the submodular NSW problem

Input: Valuations $(v_i)_{i \in A}$ over G , weights $w \in \mathbb{R}_{>0}^A$ such that $\sum_{i \in A} w_i = 1$, and $\varepsilon > 0$.

Output: Allocation $\mathcal{S} = (S_i)_{i \in A}$.

1 Find an A -perfect matching $\tau : A \rightarrow G$ maximizing $\prod_{i \in A} v_i(\tau(i))^{w_i}$ and set

$$H := \tau(A), J := G \setminus H$$

2 **for** $i \in \bar{A}$ **do**

3 $\ell(i) \leftarrow \operatorname{argmax}\{v_i(\ell) : \ell \in J\}$

4 Define $\bar{v}_i(S) := v_i(\ell(i)) + v_i(S)$

5 $\mathcal{R} = (R_i)_{i \in A} := \text{LocalSearch}(J, (\bar{v}_i)_{i \in A})$

6 Find a matching $\sigma : A \rightarrow H$ maximizing $\prod_{i=1}^n v_i(R_i + \sigma(i))^{w_i}$

7 **return** $\mathcal{S} = (R_i + \sigma(i))_{i \in A}$

Phase 1: Initial matching We find an optimal assignment of one item to each agent, i.e., a matching $\tau : A \rightarrow G$ maximizing $\prod_{i \in A} v_i(\tau(i))^{w_i}$. This can be done using a maximum-weight matching algorithm with weights $w_i \log v_i(j)$ in the bipartite graph between A and G with edge set

$\{(i, j) : v_i(j) > 0\}$. If no matching of size n exists, then we can conclude that there is no allocation with positive NSW value, and return an arbitrary allocation. For the rest of the paper, we assume there is a matching covering A , and let $H := \tau(A)$ be the set of matched items.

Phase 2: Local search In the second phase, we let $J := G \setminus H$ denote the set of items not assigned in the first phase. We let $\bar{A} := \{i \in A : v_i(J) > 0\}$ denote the set of agents that have a positive value on the items in J . For every $i \in \bar{A}$, we select

$$\ell(i) \in \operatorname{argmax}_{j \in J} v_i(j)$$

as a *favorite* item of agent i in J . By submodularity, $v_i(\ell(i)) > 0$. For each $i \in \bar{A}$, we define the endowed valuation function $\bar{v}_i : 2^J \rightarrow \mathbb{R}_{>0}$ as

$$\bar{v}_i(S) := v_i(\ell(i)) + v_i(S) \quad \forall S \subseteq J.$$

Thus, $\bar{v}_i(\emptyset) = v_i(\ell(i))$, and $\bar{v}_i(j) \leq 2\bar{v}_i(\emptyset)$ for any $j \in J$. Further, we set the accuracy parameter

$$\bar{\varepsilon} := -1 + \sqrt[3]{1 + \varepsilon}.$$

(Instead of this exact value, we can set a lower value within a constant factor range.)

Our local search starts with allocating all items to a single agent in \bar{A} . As long as moving one item to a different agent increases the potential function

$$\prod_{i \in \bar{A}} \bar{v}_i(R_i)^{w_i}$$

by at least a factor $(1 + \bar{\varepsilon})$, we perform such an exchange. Phase 2 terminates when no more such exchanges are possible, and returns the current allocation. For all agents $i \in A \setminus \bar{A}$, we let $R_i = \emptyset$.

Algorithm 2: LocalSearch($J, (\bar{v}_i)_{i \in A}$)

- 1 $\bar{A} \leftarrow \{i \in A : \bar{v}_i(J) > 0\}$
 - 2 $R_k \leftarrow J$ for some $k \in \bar{A}$ and $R_i \leftarrow \emptyset$ for $i \in A - k$
 - 3 **while** $\exists i, k \in \bar{A}$ and $j \in R_i$ such that $\left(\frac{\bar{v}_i(R_i - j)}{\bar{v}_i(R_i)}\right)^{w_i} \cdot \left(\frac{\bar{v}_k(R_k + j)}{\bar{v}_k(R_k)}\right)^{w_k} > 1 + \bar{\varepsilon}$ **do**
 - 4 $R_i \leftarrow R_i - j$ and $R_k \leftarrow R_k + j$
 - 5 **return** $\mathcal{R} := (R_i)_{i \in A}$
-

Phase 3: Rematching In the final phase, we match the items in H to the agents optimally, considering allocation $\mathcal{R} = (R_i)_{i \in A}$ of J . This can be done by again solving a maximum-weight matching problem, now with weights $w_{ij} = w_i \log v_i(R_i + j)$.

In the remainder of this section, we prove Theorem 1.1. In Section 2.1, we formulate simple properties of approximate local optimal solution found in Phase 2. This is followed by a technical bound comparing the approximate local optimal solution to the optimal solution. In this step, we present two different analyses: in Section 2.2 for the asymmetric case, and in Section 2.3 for the symmetric case. Section 2.4 gives a lower bound on the weight of the final matching found in Phase 3 of the algorithm; this argument is the same for the asymmetric and symmetric cases. This completes the proof of Theorem 1.1.

2.1 Local optima

Throughout this section, we work with the item set J , set of agents \bar{A} , favourite items $\ell(i)$, endowed valuations $\bar{v}_i(S) = v_i(\ell(i)) + v_i(S)$, and we let

$$\bar{\varepsilon} := -1 + \sqrt[m]{1 + \varepsilon}. \quad (1)$$

Thus, $(\bar{\varepsilon} + 1)^m = 1 + \varepsilon$.

Definition 2.1 ($\bar{\varepsilon}$ -local optimum). A complete allocation $\mathcal{R} = (R_i)_{i \in \bar{A}}$ is an $\bar{\varepsilon}$ -local optimum with respect to valuations \bar{v}_i , if for all pairs of different agents $i, k \in \bar{A}$ and all $j \in R_i$ it holds that

$$\left(\frac{\bar{v}_i(R_i - j)}{\bar{v}_i(R_i)} \right)^{w_i} \cdot \left(\frac{\bar{v}_k(R_k + j)}{\bar{v}_k(R_k)} \right)^{w_k} \leq (1 + \bar{\varepsilon}).$$

A 0-local optimum will be simply called *local optimum*.

Lemma 2.2. Consider an NSW instance with submodular valuations, and let $\varepsilon > 0$. Then, `LocalSearch`(J, v_1, \dots, v_n) returns an $\bar{\varepsilon}$ -local maximum with respect to the endowed valuations \bar{v}_i in $O\left(\frac{m}{\varepsilon} \log m\right)$ exchange steps.

Proof. It is immediate that the algorithm terminates with an $\bar{\varepsilon}$ -local maximum. Recalling that $\bar{v}_i(j) \leq 2\bar{v}_i(\emptyset)$ for any $j \in J$, submodularity implies $v_i(J) < (|J| + 1)\bar{v}_i(\emptyset) \leq m\bar{v}_i(\emptyset)$ for every $i \in \bar{A}$. Hence,

$$\prod_{i \in \bar{A}} \bar{v}_i(J)^{w_i} \leq m \prod_{i \in \bar{A}} \bar{v}_i(\emptyset)^{w_i},$$

and therefore the product $\prod_{i \in \bar{A}} \bar{v}_i(R_i)^{w_i}$ may grow by at most a factor m throughout all exchange steps. Every swap increases this product by at least a factor $(1 + \bar{\varepsilon})$. Thus, the total number of swaps is bounded by $\log_{(1+\bar{\varepsilon})} m = m \log_{1+\bar{\varepsilon}} m = O\left(\frac{m}{\varepsilon} \log m\right)$. \square

We need the following two properties of submodular functions.

Proposition 2.3. Let $\bar{v} : 2^J \rightarrow \mathbb{R}_{>0}$ be a monotone submodular function. Let $S \subseteq T \subseteq J$ and $j \in J$. Then,

$$\frac{\bar{v}(T + j)}{\bar{v}(T)} \leq \frac{\bar{v}(S + j)}{\bar{v}(S)}.$$

Proof. By the monotonicity and submodularity of v we have

$$\begin{aligned} \frac{\bar{v}(T + j)}{\bar{v}(T)} &= \frac{\bar{v}(T) + \bar{v}(T + j) - \bar{v}(T)}{\bar{v}(T)} \leq \frac{\bar{v}(S) + \bar{v}(T + j) - \bar{v}(T)}{\bar{v}(S)} \\ &\leq \frac{\bar{v}(S) + \bar{v}(S + j) - \bar{v}(S)}{\bar{v}(S)} = \frac{\bar{v}(S + j)}{\bar{v}(S)}. \end{aligned} \quad \square$$

Proposition 2.4. Let $\bar{v} : 2^J \rightarrow \mathbb{R}_{>0}$ be a submodular endowed valuation. For any $j \in R$,

$$\bar{v}(R - j) \geq \sum_{k \in R} (\bar{v}(R) - \bar{v}(R - k)).$$

Proof. Let us denote $R - j := \{r_1, \dots, r_s\}$. By submodularity, we have

$$\begin{aligned} \bar{v}(R - j) &= \bar{v}(\emptyset) + \sum_{k=1}^s (\bar{v}(\{r_1, \dots, r_k\}) - \bar{v}(\{r_1, \dots, r_{k-1}\})) \\ &\geq \bar{v}(\emptyset) + \sum_{k=1}^s (\bar{v}(R) - \bar{v}(R - r_k)) \geq \sum_{k \in R} (\bar{v}(R) - \bar{v}(R - r_k)) \end{aligned}$$

where in the last step, we used the fact that $\bar{v}(\emptyset) = v(\ell(i)) \geq v(j) \geq \bar{v}(R) - \bar{v}(R - j)$. \square

We analyze our local search in slightly different ways in the symmetric case (where $w_1 = \dots = w_n = 1/n$) and the general asymmetric case. We consider the asymmetric case first.

2.2 Local equilibrium analysis for asymmetric NSW

For $\bar{\varepsilon}$ as in (1), let $\mathcal{R} = (R_i)_{i \in A}$ be an $\bar{\varepsilon}$ -local optimum with respect to the endowed valuations \bar{v}_i . Let $j \in J$ and let $i \in \bar{A}$ be the agent such that $j \in R_i$. We define the *price* of j as

$$p_j := w_i \log \frac{\bar{v}_i(R_i)}{\bar{v}_i(R_i - j)}.$$

Lemma 2.5. *For an $\bar{\varepsilon}$ -local optimum $\mathcal{R} = (R_i)_{i \in \bar{A}}$ and prices p_j defined as above, for every item $j \in R_i$ and every agent $k \in \bar{A}$, we have*

$$\frac{\bar{v}_k(R_k + j)}{\bar{v}_k(R_k)} \leq (1 + \bar{\varepsilon})^{1/w_k} e^{p_j/w_k}.$$

Moreover, if the valuation \bar{v}_k is submodular, then for all $T \subseteq J$, we have

$$\frac{\bar{v}_k(R_k \cup T)}{\bar{v}_k(R_k)} \leq (1 + \bar{\varepsilon})^{|T|/w_k} \cdot e^{\sum_{j \in T} p_j/w_k}.$$

Proof. By definition, $e^{p_j/w_i} = \frac{\bar{v}_i(R_i)}{\bar{v}_i(R_i - j)}$. If $k = i$ the first statement is trivial. Otherwise, for $k \neq i$, the first statement follows from the $\bar{\varepsilon}$ -optimality of \mathcal{R} ; if false, we would swap item j to agent k .

For the second statement, let us denote $T = \{t_1, t_2, \dots, t_{|T|}\} \subseteq J$. Since \bar{v}_k is submodular, by Proposition 2.3 we have

$$\begin{aligned} \frac{\bar{v}_k(R_k \cup T)}{\bar{v}_k(R_k)} &= \prod_{a=1}^{|T|} \frac{\bar{v}_k(R_k \cup \{t_1, \dots, t_a\})}{\bar{v}_k(R_k \cup \{t_1, \dots, t_{a-1}\})} \leq \prod_{a=1}^{|T|} \frac{\bar{v}_k(R_k + t_a)}{\bar{v}_k(R_k)} \\ &\leq (1 + \bar{\varepsilon})^{|T|/w_k} e^{\sum_{j \in T} p_j/w_k}. \end{aligned} \quad \square$$

The following lemma shows that the ‘spending’ of agent i , $p(R_i)$, is at most their weight w_i .

Lemma 2.6 (Bounded spending). *For an $\bar{\varepsilon}$ -local optimum $\mathcal{R} = (R_i)_{i \in \bar{A}}$ and prices p_j defined as above, $p(R_i) \leq w_i$ for every agent $i \in \bar{A}$, and consequently, $p(J) \leq 1$.*

Proof. From the definition of p_j , we have

$$p(R_i) = w_i \sum_{j \in R_i} \log \frac{\bar{v}_i(R_i)}{\bar{v}_i(R_i - j)} \leq w_i \sum_{j \in R_i} \frac{\bar{v}_i(R_i) - \bar{v}_i(R_i - j)}{\bar{v}_i(R_i - j)} \leq w_i$$

due to the elementary inequality $\log x \leq x - 1$, and by Proposition 2.4 we know that $\sum_{j \in R_i} (\bar{v}_i(R_i) - \bar{v}_i(R_i - j)) \leq \bar{v}_i(R_i - j')$ for $j' \in \operatorname{argmin}_{j \in R_i} \bar{v}_i(R_i - j)$.

Adding up the prices over all the sets R_i , whose union is J , we obtain $p(J) = \sum_{i \in \bar{A}} p(R_i) \leq \sum_{i \in \bar{A}} w_i \leq 1$. \square

We recall the First Welfare Theorem: any Walrasian equilibrium allocation maximizes the utilitarian social welfare. For conditional equilibrium, [30, Proposition 1] give an approximate version of the first welfare theorem: the utilitarian social welfare in any conditional equilibrium is at least half of the maximal welfare. Analogously, if we interpret local optimum as equilibrium, then the following proposition states that such an equilibrium gives an e -approximation of the optimal *Nash* social welfare with respect to the *endowed valuations*. Recall that, by definition of \bar{A} , $\bar{v}_i(S) = 0$ for any $i \in A \setminus \bar{A}$ and any $S \subseteq J$.

Proposition 2.7. Let $\mathcal{R} = (R_i)_{i \in A}$ be a local optimum and $\mathcal{S} = (S_i)_{i \in A}$ be an optimal NSW allocation with respect to the endowed submodular valuations \bar{v}_i . Then

$$\prod_{i \in \bar{A}} \bar{v}_i(R_i)^{w_i} \geq \frac{1}{e} \cdot \prod_{i \in \bar{A}} \bar{v}_i(S_i)^{w_i}.$$

Proof. By Lemma 2.6, $\sum_{i \in \bar{A}} p(S_i) \leq p(J) \leq 1$. Then, by Lemma 2.5,

$$\prod_{i \in \bar{A}} \bar{v}_i(S_i)^{w_i} \leq \prod_{i \in \bar{A}} \bar{v}_i(R_i \cup S_i)^{w_i} \leq \prod_{i \in \bar{A}} \bar{v}_i(R_i)^{w_i} \cdot e^{p(S_i)} = e^{\sum_{i \in \bar{A}} p(S_i)} \cdot \prod_{i \in \bar{A}} \bar{v}_i(R_i)^{w_i} \leq e \cdot \prod_{i \in \bar{A}} \bar{v}_i(R_i)^{w_i}. \quad \square$$

Proposition 2.7 is included solely for the intuition. We cannot really use it as such, because it does not deal with the allocation of items in H . For this, we need the final rematching phase (Section 2.4). We will need a bound in the following form. The parameters h_i will represent the number of items that agent i takes from the set H in the optimum solution.

Lemma 2.8. Let $\mathcal{R} = (R_i)_{i \in \bar{A}}$ be an $\bar{\varepsilon}$ -local optimum with respect to the endowed valuations \bar{v}_i that are submodular. Let (S_1, S_2, \dots, S_n) denote any partition of the set J , and let $h_i \geq 0$ such that $\sum_{i \in A} h_i \leq n$. Then,

$$\prod_{i \in A \setminus \bar{A}} h_i^{w_i} \prod_{i \in \bar{A}} \left(\frac{v_i(S_i)}{\max\{v_i(\ell(i)), v_i(R_i)\}} + h_i \right)^{w_i} \leq (1 + \varepsilon)(2 + nw_{\max})e.$$

Proof. By Lemma 2.5, for each $i \in \bar{A}$ we can bound

$$\frac{v_i(S_i)}{\max\{v_i(\ell(i)), v_i(R_i)\}} \leq \frac{v_i(R_i \cup S_i)}{\frac{1}{2}[v_i(\ell(i)) + v_i(R_i)]} \leq \frac{2\bar{v}_i(R_i \cup S_i)}{\bar{v}_i(R_i)} \leq 2(1 + \bar{\varepsilon})^{|S_i|/w_i} e^{p(S_i)/w_i}.$$

Thus,

$$\begin{aligned} \prod_{i \in A \setminus \bar{A}} h_i^{w_i} \prod_{i \in \bar{A}} \left(\frac{v_i(S_i)}{\max\{v_i(\ell(i)), v_i(R_i)\}} + h_i \right)^{w_i} &\leq \prod_{i \in A \setminus \bar{A}} h_i^{w_i} \prod_{i \in \bar{A}} \left(2(1 + \bar{\varepsilon})^{|S_i|/w_i} e^{p(S_i)/w_i} + h_i \right)^{w_i} \\ &\leq \prod_{i \in A \setminus \bar{A}} h_i^{w_i} \prod_{i \in \bar{A}} \left((2 + h_i)(1 + \bar{\varepsilon})^{|S_i|/w_i} \cdot e^{p(S_i)/w_i} \right)^{w_i} \\ &\leq (1 + \bar{\varepsilon})^m e^{p(J)} \prod_{i \in A} (2 + h_i)^{w_i}. \end{aligned}$$

By the choice of $\bar{\varepsilon}$, $(1 + \bar{\varepsilon})^m = 1 + \varepsilon$. From Lemma 2.6, we get $p(J) \leq 1$. The proof of the lemma is complete by showing that the last product is at most $(2 + nw_{\max})$. This follows by the AM-GM inequality:

$$\prod_{i \in A} (2 + h_i)^{w_i} \leq \sum_{i \in A} w_i (2 + h_i) \leq 2 + w_{\max} \sum_{i \in A} h_i \leq 2 + nw_{\max}. \quad (2)$$

\square

2.3 Local equilibrium analysis for symmetric NSW

Let $\varepsilon, \bar{\varepsilon} \geq 0$, as in (1), and let further

$$\hat{\varepsilon} := (1 + \bar{\varepsilon})^n - 1. \quad (3)$$

Thus, $1 + \hat{\varepsilon} = (1 + \bar{\varepsilon})^n \leq (1 + \bar{\varepsilon})^m = 1 + \varepsilon$ since $n \leq m$. In particular, $0 \leq \bar{\varepsilon} \leq \hat{\varepsilon} \leq \varepsilon \leq 1$. Let $\mathcal{R} = (R_i)_{i \in A}$ be an $\bar{\varepsilon}$ -local optimum with respect to the endowed valuations \bar{v}_i , in the symmetric case.

Let $j \in J$ and let $i \in \bar{A}$ be the agent such that $j \in R_i$. We define the *price* of j as

$$p_j := \frac{\bar{v}_i(R_i)}{\bar{v}_i(R_i - j)} - 1 = \frac{\bar{v}_i(R_i) - \bar{v}_i(R_i - j)}{\bar{v}_i(R_i - j)}.$$

The following lemma gives the basic properties of these prices that we will need in the following.

Lemma 2.9. *Given an $\bar{\varepsilon}$ -local optimum $\mathcal{R} = (R_i)_{i \in A}$, and the prices p_j defined as above, we have*

(i) For every item $j \in J$,

$$p_j \leq 1.$$

(ii) For every item $j \in J \setminus R_k$,

$$\frac{\bar{v}_k(R_k + j)}{\bar{v}_k(R_k)} \leq (1 + \hat{\varepsilon})(1 + p_j).$$

(iii) For every $T \subseteq J$,

$$\frac{\bar{v}_i(R_k \cup T)}{\bar{v}_k(R_k)} \leq p(T) + 1 + 2\hat{\varepsilon}|T|.$$

Proof. (i). By construction of \bar{v}_i , $\bar{v}_i(R_i) - \bar{v}_i(R_i - j) \leq \bar{v}_i(\emptyset) \leq \bar{v}_i(R_i - j)$. Hence, $p_j = \frac{\bar{v}_i(R_i) - \bar{v}_i(R_i - j)}{\bar{v}_i(R_i - j)} \leq 1$.

(ii). Let $i \in A$ such that $j \in R_i$. From the $\bar{\varepsilon}$ -optimality of \mathcal{R} , we get

$$\frac{\bar{v}_k(R_k + j)}{\bar{v}_k(R_k)} \leq (1 + \bar{\varepsilon})^n \frac{\bar{v}_i(R_i)}{\bar{v}_i(R_i - j)} = (1 + \hat{\varepsilon})(1 + p_j),$$

because otherwise we could swap item j to agent k .

(iii). By submodularity, we have

$$\begin{aligned} \frac{\bar{v}_k(R_k \cup T)}{\bar{v}_k(R_k)} &\leq \frac{\bar{v}_k(R_k) + \sum_{j \in T} (\bar{v}_k(R_k + j) - \bar{v}_k(R_k))}{\bar{v}_k(R_k)} \\ &\leq 1 + \sum_{j \in T} ((1 + \hat{\varepsilon})(1 + p_j) - 1) \leq 1 + \sum_{j \in T} (2\hat{\varepsilon} + p_j) \end{aligned}$$

using parts (i) and (ii). □

The following lemma shows that the *spending* of each agent i , $p(R_i) = \sum_{j \in R_i} p_j$, is at most 1.

Lemma 2.10 (Bounded spending). *Let $\mathcal{R} = (R_i)_{i \in \bar{A}}$ be an $\bar{\varepsilon}$ -local optimum with respect to the endowed valuations \bar{v}_i . Then, $p(R_i) \leq 1$ for every agent $i \in \bar{A}$, and consequently, $p(J) \leq |\bar{A}|$.*

Proof. From the definition of the prices p_j , and by Proposition 2.4, we have

$$p(R_i) = \sum_{j \in R_i} \frac{\bar{v}_i(R_i) - \bar{v}_i(R_i - j)}{\bar{v}_i(R_i - j)} \leq \frac{\sum_{j \in R_i} (\bar{v}_i(R_i) - \bar{v}_i(R_i - j))}{\min_{k \in R_i} \bar{v}_i(R_i - k)} \leq 1.$$

Since (R_1, \dots, R_n) is a partition of J (every item is allocated throughout the local search), we have

$$p(J) = \sum_{j \in J} p_j = \sum_{i \in \bar{A}} \sum_{j \in R_i} p_j \leq |\bar{A}|. \quad \square$$

The next lemma bounds the value of any set relative to our local optimum in terms of prices.

Proposition 2.11. *Let $\mathcal{R} = (R_i)_{i \in A}$ be an $\bar{\varepsilon}$ -local optimum and $S \subseteq J$ any set of items. Then,*

$$\frac{v_i(S)}{\max\{v_i(R_i), v_i(\ell(i))\}} \leq 2p(S) + 1 + 4\hat{\varepsilon}|S|.$$

Proof. By Lemma 2.9,

$$\frac{v_i(\ell(i)) + v_i(S)}{v_i(\ell(i)) + v_i(R_i)} = \frac{\bar{v}_i(S)}{\bar{v}_i(R_i)} \leq \frac{\bar{v}_i(R_i \cup S)}{\bar{v}_i(R_i)} \leq p(S) + 1 + 2\hat{\varepsilon}|S|.$$

Let $\lambda := \frac{v_i(R_i)}{v_i(\ell(i))}$. We can rewrite the inequality above as follows:

$$\frac{1 + \frac{v_i(S)}{v_i(\ell(i))}}{1 + \lambda} \leq p(S) + 1 + 2\hat{\varepsilon}|S|.$$

From here,

$$\frac{v_i(S)}{v_i(\ell(i))} \leq (1 + \lambda)(p(S) + 1 + 2\hat{\varepsilon}|S|) - 1 = \lambda + (1 + \lambda)(p(S) + 2\hat{\varepsilon}|S|).$$

We use this inequality if $0 \leq \lambda \leq 1$. If $\lambda > 1$, we divide by λ to obtain:

$$\frac{v_i(S)}{v_i(R_i)} \leq 1 + (1/\lambda + 1)(p(S) + 2\hat{\varepsilon}|S|).$$

Either way, the worst case is $\lambda = 1$, which gives

$$\frac{v_i(S)}{\max\{v_i(\ell(i)), v_i(R_i)\}} \leq 1 + 2(p(S) + 2\hat{\varepsilon}|S|). \quad \square$$

The next lemma is the analogue of Lemma 2.8 in the symmetric case.

Lemma 2.12. *Let $\mathcal{R} = (R_i)_{i \in A}$ be an $\bar{\varepsilon}$ -local optimum with respect to the endowed valuations \bar{v}_i . Let (S_1, S_2, \dots, S_n) denote any allocation of the set J , and let $h_i \geq 0$ be such that $\sum_{i \in A} h_i \leq n$. Then,*

$$\prod_{i \in A \setminus \bar{A}} h_i \prod_{i \in \bar{A}} \left(\frac{v_i(S_i)}{\max\{v_i(\ell(i)), v_i(R_i)\}} + h_i \right) \leq (1 + \varepsilon)^n 4^n.$$

Proof. Using Proposition 2.11 and the AM-GM inequality,

$$\begin{aligned} \prod_{i \in A \setminus \bar{A}} h_i \prod_{i \in \bar{A}} \left(\frac{v_i(S_i)}{\max\{v_i(\ell(i)), v_i(R_i)\}} + h_i \right) &\leq \prod_{i \in A \setminus \bar{A}} h_i \prod_{i \in \bar{A}} (2p(S_i) + 1 + 4\hat{\varepsilon}|S_i| + h_i) \\ &\leq \frac{1}{n^n} \left(\sum_{i \in A \setminus \bar{A}} h_i + \sum_{i \in \bar{A}} (2p(S_i) + 1 + 4\hat{\varepsilon}|S_i| + h_i) \right)^n \\ &= \left(\frac{\sum_{i \in A} h_i}{n} + \frac{|\bar{A}|}{n} + \frac{4\hat{\varepsilon}|J|}{n} + \frac{2p(J)}{n} \right)^n. \end{aligned}$$

We upper-bound each of these two summands. First, using $\sum_{i \in A} h_i \leq n$. Second, using $|\bar{A}| \leq n$. Third, using $|J| \leq m$. Fourth, using $p(J) \leq |\bar{A}| \leq n$ from Lemma 2.10. We obtain

$$\left(\frac{\sum_{i \in A} h_i}{n} + \frac{|\bar{A}|}{n} + \frac{4\hat{\varepsilon}|J|}{n} + \frac{2p(J)}{n} \right)^n \leq \left(1 + 1 + \frac{4m\hat{\varepsilon}}{n} + 2 \right)^n = 4^n \left(1 + \frac{m\hat{\varepsilon}}{n} \right)^n.$$

Since $m \geq n$, by Bernoulli's inequality $4^n \left(1 + \frac{m\hat{\varepsilon}}{n} \right)^n \leq 4^n (1 + \hat{\varepsilon})^{\frac{mn}{n}} = (1 + \varepsilon)^n 4^n$. \square

2.4 Rematching

Throughout, let OPT denote the optimum NSW value of the instance. For sets $\mathcal{R} = (R_i)_{i \in A}$, and a matching $\pi : A \rightarrow H \cup \{\perp\}$, we let

$$\text{NSW}(\mathcal{R}, \pi) := \prod_{i \in A} v_i(R_i + \pi(i))^{w_i}.$$

In Phase 3, we select a matching $\rho : A \rightarrow H$ that maximizes $\text{NSW}(\mathcal{R}, \rho)$, where $\mathcal{R} = (R_i)_{i \in A}$ denotes the $\bar{\varepsilon}$ -local optimum with respect to the endowed valuations \bar{v}_i from Phase 2. The following lemma completes the proof of Theorem 1.1.¹

Lemma 2.13. *Let $\varepsilon \geq 0$, $\bar{\varepsilon}$ as in (1), and let $\mathcal{R} = (R_i)_{i \in A}$ be an $\bar{\varepsilon}$ -local optimum with respect to the endowed valuations \bar{v}_i . Then, there exists a matching $\rho : A \rightarrow H$ such that*

$$\text{NSW}(\mathcal{R}, \rho) \geq \frac{\text{OPT}}{(2 + nw_{\max})e(1 + \varepsilon)}.$$

Further, in the symmetric case, the following stronger bound holds:

$$\text{NSW}(\mathcal{R}, \rho) \geq \frac{\text{OPT}}{4(1 + \varepsilon)}.$$

The proof of Lemma 2.13 relies on the following elementary statement on bipartite matchings. For $k = 1$, this is a very natural statement about matchings, but we are not aware of a previous explicit occurrence in the literature.

Lemma 2.14. *Let $\mathcal{G} = (A, B; c)$ be a complete bipartite graph with edge weights $c \in (\mathbb{R} \cup \{-\infty\})^{A \times B}$. Let $\tau : A \rightarrow B$ be a maximum-weight A -perfect matching, and let $H := \tau(A) \subseteq B$ denote the set of matched nodes in B . For some integer $k \geq 1$, let us obtain from \mathcal{G} the bipartite graph $\mathcal{G}' = (A, B'; c')$ by creating k copies of every node in $B \setminus H$; thus, $|B'| = k|B| - (k - 1)|A|$. Then, for any $X \subseteq A$, there is a maximum-weight X -perfect matching $\rho : X \rightarrow B'$ such that $\rho(X) \subseteq H$.*

Proof. Let us start with the case $k = 1$, i.e., $B' = B$. Take any maximum-weight X -perfect matching $\rho : X \rightarrow B$ with minimum size of $\rho(X) \setminus H$. For the sake of contradiction assume $\rho(X) \setminus H \neq \emptyset$ and let $j \in \rho(X) \setminus H$. Let us consider the symmetric difference $\tau \Delta \rho$; this consists of cycles and paths. Since $j \notin H$, j is an endpoint of an alternating path P between ρ and τ that starts in j with a ρ -edge and ends in a τ -edge (since τ is an A -perfect matching). Let $Y \subseteq A$ be the set of nodes incident to P . Clearly, $Y \subseteq X$. We get another A -perfect matching $\tau' : A \rightarrow B$ by flipping τ along P , i.e., setting $\tau'(i) = \rho(i)$ for $i \in Y$ and $\tau'(i) = \tau(i)$ for $i \in A \setminus Y$. Similarly, we obtain another X -perfect matching $\rho' : X \rightarrow B$ by flipping ρ along P , i.e., setting $\rho'(i) = \tau(i)$ for $i \in Y$ and $\rho'(i) = \rho(i)$ for $i \in X \setminus Y$.

By the optimality of τ , the weight of τ -edges in P is at least the weight of ρ -edges in P . This implies that the weight of ρ' is at least the weight of ρ . This contradicts the choice of ρ since $\rho'(X) \setminus H = (\rho(X) \setminus H) \setminus \{j\}$.

The proof of $k > 1$ follows similarly: we consider the symmetric difference of τ (naturally embedded into the larger graph \mathcal{G}') and ρ . We take an alternating path P starting from a node $j' \in \rho(X) \setminus H$. Assume j' is one of the k copies of a node $j \in B \setminus H$. The A -perfect matching $\tau' : A \rightarrow B'$ obtained from τ by flipping along P will assign $\tau'(i') = j'$ for some $i' \in A$. This naturally maps back to an A -perfect matching $\bar{\tau} : A \rightarrow B$ of the same weight in the original graph,

¹One needs to select a smaller parameter ε to obtain the bounds in Theorem 1.1.

by setting $\bar{\tau}(i') = j$, and $\bar{\tau}(k) = \tau'(k)$ for $k \neq i'$. This again shows that the weight of τ -edges in P is at least the weight of ρ -edges in P , and we obtain a contradiction by constructing a better X -perfect matching $\rho' : A \rightarrow B'$ as in the $k = 1$ case. \square

Proof of Lemma 2.13. Consider an optimal solution $(S_1 \cup H_1, \dots, S_n \cup H_n)$ to the NSW problem where S_i is the set of items allocated to i from $J = G \setminus H$, and H_i is the set of items allocated to i from H . For $i \in A \setminus \bar{A}$, we must have $H_i \neq \emptyset$, and we can assume $S_i = \emptyset$. Let $h_i := |H_i|$. We define a matching $g : A \rightarrow H \cup \{\perp\}$ as follows. If $h_i > 0$, let $g(i) \in \operatorname{argmax}_{j \in H_i} v_i(S_i + j)$ be one of the items in H_i providing the largest marginal gain to agent i . Otherwise, let $g(i) := \perp$. Submodularity implies

$$v_i(S_i \cup H_i) \leq v_i(S_i) + h_i v_i(g(i)) \quad \forall i \in A. \quad (4)$$

Let us partition the set of agents A as

$$\begin{aligned} A_\pi &:= \{i \in A : v_i(g(i)) \geq \max\{v_i(R_i), v_i(\ell(i))\}\}, \\ A_R &:= \{i \in A \setminus A_\pi : v_i(R_i) \geq \max\{v_i(g(i)), v_i(\ell(i))\}\}, \\ A_\ell &:= \{i \in A \setminus (A_\pi \cup A_R) : v_i(\ell(i)) \geq \max\{v_i(R_i), v_i(g(i))\}\}. \end{aligned}$$

Note that $A \setminus \bar{A} \subseteq A_\pi$. As an intermediate step in the construction of the claimed matching ρ , we first define an allocation $\mathcal{T} = (T_i)_{i \in A}$ and matching $\pi : A \rightarrow H \cup \{\perp\}$ as follows.

- For $i \in A_\pi$, let $T_i := \emptyset$ and $\pi(i) := g(i)$.
- For $i \in A_R$, let $T_i := R_i$ and $\pi(i) := \perp$.
- For $i \in A_\ell$, let $T_i := \{\ell(i)\}$ and $\pi(i) := \perp$.

Note that this allocation is not feasible: $\ell(i) = \ell(i')$ is possible for different agents, and the same item may even be contained in R_i for some $i \in A_R$.

Let $X := A_\pi \cup A_\ell$. By taking $k = n$ copies of all goods in $G \setminus H$, mapping $i \in A_\pi$ to $g(i)$ and $i \in A_\ell$ to a copy of $\ell(i)$ gives a matching between X and the extended set of items. Let us now apply Lemma 2.14 for the cost functions $c_{ij} = \log v_i(j)$ and $k = n$. This guarantees the existence of a matching $\rho : X \rightarrow H$ with at least the same cost. Let us extend ρ to A by $\rho(i) = \perp$ for $i \in A \setminus X = A_R$. Now, we have $\text{NSW}(\mathcal{R}, \rho) \geq \text{NSW}(\mathcal{T}, \pi)$. The following claim completes the proof.

Claim.

$$\text{NSW}(\mathcal{T}, \pi) \geq \frac{\text{OPT}}{(2 + nw_{\max})e(1 + \varepsilon)},$$

and for the symmetric setting we have

$$\text{NSW}(\mathcal{T}, \pi) \geq \frac{\text{OPT}}{4(1 + \varepsilon)}.$$

Proof. Our goal is to upper bound

$$\frac{\text{OPT}}{\text{NSW}(\mathcal{T}, \pi)} = \prod_{i \in A_\pi} \left(\frac{v_i(S_i \cup H_i)}{v_i(\pi(i))} \right)^{w_i} \prod_{i \in A_R} \left(\frac{v_i(S_i \cup H_i)}{v_i(R_i)} \right)^{w_i} \prod_{i \in A_\ell} \left(\frac{v_i(S_i \cup H_i)}{v_i(\ell(i))} \right)^{w_i}.$$

In order to do so, we first upper bound the loss of each agent depending in which set they belong. If $i \in A \setminus \bar{A}$ then $i \in A_\pi$, by (4) and submodularity, we have

$$\frac{v_i(S_i \cup H_i)}{v_i(\pi(i))} \leq \frac{h_i v_i(g(i))}{v_i(\pi(i))} = h_i.$$

If $i \in A_\pi \cap \bar{A}$, by (4), as well as using the definition of A_π and submodularity, we can bound

$$\frac{v_i(S_i \cup H_i)}{v_i(\pi(i))} \leq \frac{v_i(S_i) + h_i v_i(g(i))}{v_i(\pi(i))} = \frac{v_i(S_i)}{v_i(\pi(i))} + h_i \leq \frac{v_i(S_i)}{\max\{v_i(\ell(i)), v_i(R_i)\}} + h_i.$$

Similarly, if $i \in A_R$, we get

$$\frac{v_i(S_i \cup H_i)}{v_i(R_i)} \leq \frac{v_i(S_i) + h_i v_i(g(i))}{v_i(R_i)} \leq \frac{v_i(S_i)}{v_i(R_i)} + h_i = \frac{v_i(S_i)}{\max\{v_i(\ell(i)), v_i(R_i)\}} + h_i.$$

Finally, if $i \in A_\ell$, the bound is

$$\frac{v_i(S_i \cup H_i)}{v_i(\ell(i))} \leq \frac{v_i(S_i) + h_i v_i(g(i))}{v_i(\ell(i))} \leq \frac{v_i(S_i)}{v_i(\ell(i))} + h_i = \frac{v_i(S_i)}{\max\{v_i(\ell(i)), v_i(R_i)\}} + h_i.$$

Consequently,

$$\frac{\text{OPT}}{\text{NSW}(\mathcal{T}, \pi)} \leq \prod_{i \in A \setminus \bar{A}} h_i^{w_i} \prod_{i \in A} \left(\frac{v_i(S_i)}{\max\{v_i(\ell(i)), v_i(R_i)\}} + h_i \right)^{w_i}.$$

The proof of the claim is complete by Lemmas 2.8 for the general case and by Lemma 2.12 in the symmetric case. \blacksquare

\square

Remark 2.15 (Natura [49]). The bound in Theorem 1.1 can be in fact strengthened to $\exp(\log(2n) + \sum_{i=1}^n w_i \log w_i) \cdot (e + \varepsilon)$, a slight improvement over the bound in [13] for the additive case. This follows by strengthening the bound $(1 + \varepsilon)(2 + n w_{\max})$ in Lemma 2.8 to $(1 + \varepsilon)2n \prod_{i \in A} w_i^{w_i}$. To see this, we only need to change the final estimate (2) in the proof. Here, we are given $h_i \geq 0$ such that $\sum_{i \in A} h_i \leq n$, and need to upper bound $\prod_{i \in A} (2 + h_i)^{w_i}$. Instead of simply using AM-GM as in (2), we use AM-GM for a reweighted product:

$$\prod_{i \in A} (2 + h_i)^{w_i} = \prod_{i \in A} w_i^{w_i} \prod_{i \in A} \left(\frac{2 + h_i}{w_i} \right)^{w_i} \leq \prod_{i \in A} w_i^{w_i} \sum_{i \in A} (2 + h_i) \leq 2n \prod_{i \in A} w_i^{w_i}.$$

2.5 Our techniques and comparison with previous approaches

We now compare our algorithm to those in [34] and in [44]. At a high level, all three algorithms proceed in three phases, with Phases 1 and 3 being the same as outlined above. However, they largely differ in how the allocation \mathcal{R} of $J = G \setminus H$ is obtained in Phase 2.

Garg, Husić, and Végő [34] use a rational convex relaxation, based on the concave extension of Rado valuations. After solving the relaxation exactly, they use combinatorial arguments to sparsify the support of the solution and construct an integral allocation.

Li and Vondrák [44] allow arbitrary submodular valuations. For submodular functions, the concave extension is NP-hard to evaluate. Instead, they work with the multilinear extension. This can be evaluated with random sampling, but it is not convex. To solve the relaxation (approximately), they use an iterated continuous greedy algorithm. The allocation \mathcal{R} is obtained by independent randomized rounding of this fractional solution. Whereas the algorithm is simple, the analysis is somewhat involved. The main tool to analyze the rounding is the Efron–Stein concentration inequality; but this only works well if every item in the support of the fractional solution has bounded value. This is not true in general, and the argument instead analyzes a two-stage randomized rounding that gives a lower bound on the performance of the actual algorithm. First, a set of ‘large’ fractional items is preserved, and a careful combinatorial argument is needed to complete the allocation.

Our approach for the second part is radically different and much simpler. We do not use any continuous relaxation, but \mathcal{R} is obtained by a simple local search with respect to the modified valuation functions. Because of using these modified valuations, we can first guarantee a high NSW value of the infeasible allocation $(R_i + \ell(i))_{i \in A}$ of J in the analysis. Our analysis of the local search is inspired by the *conditional equilibrium* notion introduced by Fu, Kleinberg, and Lavi [30]. They show that any conditional equilibrium 2-approximates the utilitarian social welfare and give an auction algorithm for finding such an equilibrium under submodular valuations.

We note that local search applied directly to the NSW problem cannot yield a constant factor approximation algorithm even if we allow changing an arbitrary fixed number k of items. This can be seen already when $m = n$, i.e., every allocation with positive NSW value is a matching. Also, some other natural variants of local search do not work, or the analysis is not clear; for example, our analysis does not seem to extend when local search is applied to the (seemingly more natural) choice of $\bar{v}_i(S) = v_i(S + \tau(i))$. The idea of defining $\ell(i)$ and using the modified valuation functions is inspired by rounding of the fractional solution from previous approaches; the role of the $\ell(i)$ items is similar to the large items in [44], but we obtain much better guarantees using a more direct deterministic approach.

The last part of the analysis concerns the rematching in Phase 3. Here, we convert the infeasible allocation $(R_i + \ell(i))_{i \in A}$ to a feasible allocation by an alternating path argument, combining the initial matching τ and an (unknown) optimal matching g . While the rematching phase was already present (and essentially identical) in [34] and [44], it is implemented and analyzed differently here. We show the existence of a matching ρ that together with \mathcal{R} gives good approximation of the optimum. The papers [34] and [44] find such ρ by first showing that there is matching π that has high NSW together with \mathcal{R} and the items $\ell(i)$. Then, they show in a convoluted way that we can remove the items $\ell(i)$ and find a matching ρ (as a combination of π and the initial matching τ) while only losing only a constant in objective when compared to the solution consisting of π, \mathcal{R} and the $\ell(i)$'s.

We prove the existence of a good matching ρ by carefully analyzing the alternating cycles in the union of the optimal allocation of H and the initial matching τ of Phase 1. Our proof is radically simpler than the previous analysis of [34] and [44]. The key is a self-contained lemma on a monotonicity property of maximum-weight bipartite matchings on subsets of nodes (Lemma 2.14).

The new rematching procedure is also responsible for much of the approximation factor improvement. (The exact numbers are difficult to compare as the loss depends on the properties of solutions obtained in Phase 2, and since in the current paper the analyses of Phase 2 and Phase 3 are done in a more synchronous way.) We note that the particular matching ρ mentioned here is not needed; the algorithm finds the most profitable matching with respect to the \mathcal{R} . This provides a solution at least as good as the one in the analysis.

3 Finding fair and efficient allocations

In this section, we restate and prove Theorem 1.2; the corresponding algorithm is described in Algorithm 3.

Theorem 1.2. *There is a deterministic strongly polynomial-time algorithm that given a symmetric NSW instance with subadditive valuations and given a (complete or partial) allocation \mathcal{S} of the items, it returns a complete allocation \mathcal{T} that is $1/2$ -EFX and $\text{NSW}(\mathcal{T}) \geq \text{NSW}(\mathcal{S})/2$.*

Our first key tool is a subroutine that finds a partial allocation that is $1/2$ -EFX and preserves a large fraction of the NSW value.

Algorithm 3: Guaranteeing $1/2$ -EFX for the symmetric NSW problem

Input: Allocation \mathcal{S} that is α -approximation to the NSW problem $(A, G, (v_i)_{i \in A})$.

Output: Allocation \mathcal{T} that is $1/2$ -EFX and 2α -approximation to the symmetric NSW problem.

```
1  $\mathcal{T} \leftarrow \mathcal{S}$ 
2 repeat
3    $\mathcal{T} \leftarrow \text{MakeFairOrEfficient}(\mathcal{T})$  // Algorithm 4
4 until  $\mathcal{T}$  is not  $1/2$ -EFX
5  $U \leftarrow G \setminus \cup_i T_i$  // set of unallocated items
6 repeat
7   Let  $j \in U$  be such that  $v_i(T_i) < v_i(j)$  for some agent  $i$ 
8    $T_i \leftarrow \{j\}$ 
9    $U \leftarrow (U \cup T_i) - j$ 
10 until  $v_i(T_i) \geq v_i(j), \forall i \in A, \forall j \in U$ 
11  $\mathcal{T} \leftarrow \text{EnvyFreeCycle}(\mathcal{T}, U)$ 
12 return  $\mathcal{T}$ 
```

Lemma 3.1. *There exists a deterministic strongly polynomial algorithm $\text{MakeFairOrEfficient}(\mathcal{T})$, that, for any partial allocation \mathcal{T} , returns another partial allocation \mathcal{R} that satisfies one of the following properties*

- (i) $\text{NSW}(\mathcal{R}) \geq \text{NSW}(\mathcal{T})$ and $\cup_{i \in A} R_i \subsetneq \cup_{i \in A} T_i$, or
- (ii) $\text{NSW}(\mathcal{R}) \geq \frac{1}{2} \text{NSW}(\mathcal{T})$ and \mathcal{R} is $1/2$ -EFX.

This is shown by modifying the approach of Caragiannis, Gravin, and Huang [15]. For additive valuations, their algorithm takes an input allocation \mathcal{T} and returns a partial allocation \mathcal{R} that is EFX and $\text{NSW}(\mathcal{R}) \geq \frac{1}{2} \text{NSW}(\mathcal{T})$. We simplify and extend this approach from additive to subadditive valuations, but prove only the weaker $1/2$ -EFX property.

The key subroutine for them provides a similar alternative as in Lemma 3.1. In outcome (ii), they have the stronger EFX guarantee, while in outcome (i), they show that the NSW value increases by a certain factor. In outcome (i), it is not clear how an increase in the NSW value could be shown for subadditive valuations. However, arguing about the support decrease leads to a simpler argument.

In [15], only a partial EFX allocation is found. Theorem 1.2 shows the existence of a complete allocation, albeit with the weaker $1/2$ -EFX property. To derive Theorem 1.2, we start by repeatedly calling $\text{MakeFairOrEfficient}$ until outcome (ii) is reached. Note that the outcome (i) can only happen at most m times because the number of items in \mathcal{R} reduces by at least one after each call.

The allocation at this point may be partial. We show that the remaining items can be allocated using the classical *envy-free cycle procedure* by Lipton, Markakis, Mossel, and Saberi [45]. Even though this procedure is known for the weaker EF1 property [14], we show that—after a suitable preprocessing step—it can produce an $1/2$ -EFX allocation while not decreasing the NSW value of the allocation.

In the remainder of this section, we prove Theorem 1.2. In Section 3.1, we derive Theorem 1.2 from Lemma 3.1. The algorithmic proof is given in Algorithm 3. This relies on the subroutines $\text{MakeFairOrEfficient}$ and EnvyFreeCycle . The first one is the subroutine asserted in Lemma 3.1; this is described and analyzed in Section 3.2. EnvyFreeCycle is the envy-free cycle procedure adapted from [45] and is described below.

3.1 Completing the partial allocation

The input of Algorithm 3 is an allocation \mathcal{S} that is α -approximation to the symmetric NSW problem. It then repeatedly calls `MakeFairOrEfficient`(\mathcal{T}) (Algorithm 4) until the final allocation is $1/2$ -EFX and 2α -approximation to the symmetric NSW problem. Recall that the output of this subroutine is either a partial allocation \mathcal{T}' that satisfies either $\text{NSW}(\mathcal{T}') \geq \text{NSW}(\mathcal{T})$ and $\cup_i T'_i \subsetneq \cup_i T_i$, or $\text{NSW}(\mathcal{T}') \geq 1/2 \text{NSW}(\mathcal{T})$ and \mathcal{T}' is $1/2$ -EFX. Since $\cup_i T'_i \subsetneq \cup_i T_i$ in each call in the first case, the number of calls to Algorithm 4 is at most m .

At this point, we have an $1/2$ -EFX partial allocation \mathcal{T} with $\text{NSW}(\mathcal{T}) \geq \frac{1}{2} \text{NSW}(\mathcal{S})$. The rest of Algorithm 3 allocates the remaining items $U = G \setminus \cup_{i \in A} T_i$ so that $\text{NSW}(\mathcal{T})$ does not decrease, and the $1/2$ -EFX property is maintained.

First, we modify the allocation in the second repeat loop to ensure that each agent's value for their bundle is at least their value for each remaining item in U . This is done by swapping an agent's bundle T_i with a singleton item $j \in U$ whenever i values j more than the entire bundle T_i .

Finally, we run the envy-cycle procedure `EnvyFreeCycle`(\mathcal{T}, U) from [45] to allocate the remaining items in U , starting with the allocation \mathcal{T} . The envy-cycle procedure maintains the directed (envy) graph $D = (A, E)$, where $(i, j) \in E$ if i envies j 's bundle, i.e., $v_i(Y_i) < v_i(Y_j)$. If there is a cycle in G , then we can circulate bundles along the cycle to improve each agent's utility. Otherwise, there must be a source agent in G , whom no agent envies. We then assign an arbitrary item from U to a source agent. We update the envy graph, and iterate until U is fully assigned.

We now verify the correctness and efficiency of this algorithm.

Lemma 3.2. *The second repeat loop of Algorithm 3 is repeated at most nm times. It maintains the $1/2$ -EFX and $\text{NSW}(\mathcal{T})$ is non-decreasing.*

Proof. The bound on the number of swaps follows because every agent $i \in A$ may swap their bundle at most m times. After the first swap, they maintain a singleton bundle, and they can swap their bundle for the same item j only once, since their valuation $v_i(T_i)$ strictly increases in each swap.

It is immediate that $\text{NSW}(\mathcal{T})$ is non-decreasing. It is left to show that the $1/2$ -EFX property is maintained. Let $i \in A$ be the agent who swapped their bundle T_i for $T'_i = \{j\}$ in the current iteration. Then, the value of i 's own bundle increased while the allocation of everyone else remained the same. Hence, agent i cannot violate the $1/2$ -EFX property. For the other agents $k \neq i$, $v_k(T_k) \geq 1/2 \cdot v_k(T'_i - g)$ for all $g \in T'_i$ trivially holds, since T'_i is a singleton. \square

Property (5) below is satisfied after the second repeat loop. Hence, the next lemma completes the analysis of Algorithm 3.

Lemma 3.3. *The subroutine `EnvyFreeCycle`(\mathcal{T}, U) terminates in $O(n^3m)$ time, and $\text{NSW}(\mathcal{T})$ is non-decreasing. Assume that $\mathcal{T} = (T_i)_{i \in A}$ is $1/2$ -EFX, and*

$$v_i(T_i) \geq v_i(j) \quad \forall i \in A, \forall j \in U. \quad (5)$$

Then, `EnvyFreeCycle`(\mathcal{T}, U) also maintains the $1/2$ -EFX property.

Proof. The running time analysis is the same as in [45]. Finding and removing a cycle in the envy-graph can be done in $O(n^2)$ time. Further, whenever swapping around a cycle, at least one edge is removed from the envy graph. New edges can only be added when we allocate new items from U , with at most n edges every time. Since $|U| \leq m$, the total number of new edges added throughout is nm . This yields the overall $O(n^3m)$ bound.

Again, it is immediate that $\text{NSW}(\mathcal{T})$ is non-decreasing in every step. We need to show that the $1/2$ -EFX property is maintained both when swapping around cycles and when adding new items

from U . When swapping around a cycle, this follows since the set of bundles remains the same, and no agent's value decreases.

Consider the case when a source agent say i , gets a new item j : their new bundle becomes $T'_i = T_i + j$. Note that i is the only agent whose value increases; all other bundles remain the same. We need to show that for any $k \neq i$,

$$v_k(T_k) \geq \frac{1}{2}v_k(T'_i - g) \quad \forall g \in T'_i.$$

We show that

$$v_k(T'_i - g) = v_k(T_i + j - g) \leq v_k(T_i) + v_k(j) \leq 2v_k(T_k).$$

Here, the first inequality follows by subadditivity and monotonicity. The second inequality uses (5), and that $v_k(T_k) \geq v_k(T_i)$, since i was a source node in the envy graph. \square

3.2 Finding a fair or an efficient allocation

In this Section, we prove Lemma 3.1. The subroutine `MakeFairOrEfficient`(\mathcal{T}) is shown in Algorithm 4, and generalizes an algorithm by Caragiannis, Gravin, and Huang [15] from additive to subadditive valuations. We begin with defining the notions of *1/2-EFX feasible bundles and graph*.

Definition 3.4 (*1/2-EFX feasible bundles and graph*). Given a partial allocation $\mathcal{T} = (T_i)_{i \in A}$, we say that T_k is a *1/2-EFX feasible bundle for agent i* , if

$$v_i(T_k) \geq \frac{1}{2} \max_{\ell \in A, j \in T_\ell} v_i(T_\ell - j).$$

The *1/2-EFX feasibility graph* of \mathcal{T} is a bipartite graph $\mathcal{K} = (A \cup \mathcal{T}, E)$ where the edge set E is defined as:

$$E = \{(i, T_i) \mid T_i \text{ is } 1/2\text{-EFX feasible for } i\} \cup \left\{ (i, T_k) \mid v_i(T_k) > 2v_i(T_i) \text{ and } v_i(T_k) \geq \max_{\ell \in A, j \in T_\ell} v_i(T_\ell - j) \right\}. \quad (6)$$

The following claim can be easily verified using the definition.

Claim 3.5. *The degree of every node $i \in A$ is at least 1 in the graph $\mathcal{K} = (A \cup \mathcal{T}, E)$.*

In this section, a *matching* will refer to a matching between agents and bundles (and not between agents and items as in previous sections). Thus, a matching is a mapping $\rho : A \rightarrow \mathcal{T} \cup \{\perp\}$ such that $\rho(i) = \rho(k)$ implies $\rho(i) = \rho(k) = \perp$. A *perfect matching* has $\rho(i) \neq \perp$ for every $i \in A$. Matchings may use pairs (i, T_k) that are not in E ; we say that ρ is a matching in the bipartite graph $\mathcal{K} = (A \cup \mathcal{T}, E)$ if $(i, \rho(i)) \in E$ whenever $\rho(i) \neq \{\perp\}$. For two matchings ρ and τ , an *alternating path between ρ and τ* is a path $P = (i_1, S_{i_1}, i_2, \dots, S_{i_{k-1}}, i_\ell, S_{i_\ell})$ such that $\rho(i_t) = S_{i_t}$, $t = 1, \dots, \ell$, $\tau(i_{t+1}) = S_{i_t}$, $t = 1, \dots, \ell - 1$. The following lemma is immediate from the definition of the 1/2-EFX feasibility graph.

Lemma 3.6. *If the 1/2-EFX feasibility graph $\mathcal{K} = (A \cup \mathcal{T}, E)$ of an allocation \mathcal{T} contains a perfect matching ρ , then $(i, \rho(i))_{i \in A}$ is a 1/2-EFX allocation.*

We now give an overview of Algorithm 4. For an input partial allocation $\mathcal{T} = (T_i)_{i \in A}$, it returns a partial allocation \mathcal{R} that satisfies one of the alternatives in Lemma 3.1: either (i) $\text{NSW}(\mathcal{R}) \geq \text{NSW}(\mathcal{T})$ and $\cup_i R_i \subsetneq \cup_i T_i$, or (ii) $\text{NSW}(\mathcal{R}) \geq \frac{1}{2} \text{NSW}(\mathcal{T})$ and \mathcal{R} is 1/2-EFX.

The algorithm gradually ‘*trims down*’ the bundles \mathcal{T} . That is, we maintain a partial allocation $\mathcal{S} = (S_i)_{i \in A}$ with $S_i \subseteq T_i$ throughout. Every main loop of the algorithm either terminates by

Algorithm 4: MakeFairOrEfficient(\mathcal{T})

Input: Partial allocation \mathcal{T} .**Output:** Partial allocation \mathcal{R} such that either $\text{NSW}(\mathcal{R}) \geq \text{NSW}(\mathcal{T})$ and $\cup_i R_i \subsetneq \cup_i T_i$, or $\text{NSW}(\mathcal{R}) \geq \frac{1}{2} \text{NSW}(\mathcal{T})$ and \mathcal{R} is $1/2$ -EFX.

```
1  $\mathcal{S} \leftarrow \mathcal{T}$ 
2 repeat
3    $\mathcal{K} = (A \cup \mathcal{S}, E) \leftarrow 1/2\text{-EFX}$  feasibility graph of  $\mathcal{S}$  // Definition 3.4
4    $\mathcal{L} \leftarrow \{S_i, i \in A \mid S_i \subsetneq T_i\}$  // set of trimmed down bundles
5   Define matching  $\tau$  with  $\tau(i) = S_i$  for all  $i \in A$  // candidate matching
6    $\rho \leftarrow$  matching in  $\mathcal{K}$  where // Lemma 3.7
      (a) all bundles in  $\mathcal{L}$  are matched,
      (b)  $|\{i : \rho(i) = S_i\}|$  is maximized subject to (a), and
      (c)  $\rho$  is maximum subject to (a) and (b)
      if  $\exists i_1 \in A$  not matched in  $\rho$  then
         $(S_h, g_h) \leftarrow \arg \max_{k \in A, g \in S_k} v_{i_1}(S_k - g)$ 
        if  $v_h(S_h - g_h) \geq \frac{1}{2} \cdot v_h(T_h)$  then
           $S_h \leftarrow S_h - g_h$ 
        else
           $P = (i_1, S_{i_1}, i_2, \dots, S_{i_{\ell-1}}, i_{\ell}, S_{i_{\ell}}) \leftarrow$  alternating path between  $\tau$  and  $\rho$  starting at
           $i_1$  and ending at either  $S_{i_{\ell}} = S_h$  or an unmatched  $S_{i_{\ell}} \neq S_h$  // Lemma 3.8
        7 Construct  $\mathcal{R}$ :
        8    $R_{i_1} \leftarrow S_h - g_h$ 
        9   for  $f \leftarrow 2$  to  $\ell$  do  $R_{i_f} \leftarrow S_{i_{f-1}}$ 
        10   for  $i \in A \setminus (\{i_1, \dots, i_{\ell}\} \cup \{h\})$  do  $R_i \leftarrow T_i$ 
        11   if  $P$  ends at an unmatched bundle  $S_{i_{\ell}} \neq S_h$  then
        12      $R_h \leftarrow T_h \setminus (S_h - g_h)$ 
        13   return  $\mathcal{R}$ 
14 until  $\rho$  is a perfect matching in  $\mathcal{K}$ 
15 return  $\mathcal{R} = (\rho(i))_{i \in A}$ 
```

constructing an allocation \mathcal{R} satisfying (ii), or removes an item from one of the S_h sets. The other possible termination option is when the $1/2$ -EFX feasibility graph of \mathcal{S} contains a perfect matching ρ . In this case, we return $\mathcal{R} = (\rho(i))_{i \in A}$. This is a $1/2$ -EFX allocation by Lemma 3.6; Lemma 3.9 shows it also satisfies $\text{NSW}(\mathcal{R}) \geq \frac{1}{2} \text{NSW}(\mathcal{T})$ and is thus a suitable output of type (ii).

At the beginning of each main loop, we define two matchings. The first is the perfect matching τ that simply defines $\tau(i) = S_i$ for all $i \in A$. The second is a matching ρ in \mathcal{K} . This is required to satisfy three properties: First, it matches all trimmed down bundles, i.e., all bundles S_i with $S_i \subsetneq T_i$. Second, $|\{i : \rho(i) = S_i\}|$ is maximized subject to the first requirement. Third, subject to these requirements, ρ is chosen as a maximal matching. (The existence of such a matching is guaranteed by Lemma 3.7 below).

If ρ is not perfect, then we consider an unmatched agent i_1 , and find the bundle that maximizes i_1 's utility after removal of one item. Let $(S_h, g_h) \in \arg \max_{k \in A, g \in S_k} v_{i_1}(S_k - g)$. If agent h 's value of $S_h - g_h$ is at least $\frac{1}{2}$ times their value for the original bundle T_h , then we remove g_h from S_h and the main loop finishes. Otherwise, we construct an alternating path between ρ and τ , denoted

as $P = (i_1, S_{i_1}, i_2, S_{i_2}, \dots, S_{i_{\ell-1}}, i_\ell, S_{i_\ell})$, starting with i_1 and ending with either $S_{i_\ell} = S_h$ or an unmatched bundle $S_{i_\ell} \neq S_h$. Lemma 3.8 shows that such a P exists. Using P , we construct an allocation \mathcal{R} in line 7. Lemma 3.9 shows that this is a suitable output of type (i).

3.2.1 Analysis

The number of iterations of the repeat loop is at most m , because in each iteration except the final one, an item is removed from one of the bundles. Noting that we can find the maximum matching in line 6 and alternating path in line 6 in strongly polynomial-time, Algorithm 4 runs in strongly polynomial-time.

The next lemma guarantees that the matching ρ is well-defined. The proof follows similarly as in [15].

Lemma 3.7. *In each iteration of the repeat loop in Algorithm 4, a matching exists in \mathcal{K} where all bundles of \mathcal{L} are matched.*

Proof. Let $E^{(t)}$ denote the edge set of the 1/2-EFX feasibility graph and $\mathcal{L}^{(t)}$ the set of trimmed down bundles, and $\rho^{(t)}$ the maximum matching in the t -th iteration.

We show by induction that there exists a matching $\rho^{(t)}$ such that all bundles in $\mathcal{L}^{(t)}$ are matched. At the beginning of the first iteration, $\mathcal{L}^{(1)}$ is empty, so the claim is clearly true. Suppose the claim is true until the beginning of $(t+1)$ -st iteration. Let \mathcal{S} denote the trimmed down bundles in the t -th iteration, and let i_1 be the unmatched agent, and (S_h, g_h) the bundle and item selected in line 6.

By the requirement that $|\{i : \rho(i) = S_i\}|$ is maximized subject to all trimmed down bundles being matched, we have $(i_1, S_{i_1}) \notin E^{(t)}$. This means that $2v_{i_1}(S_{i_1}) < \max_{\ell \in A, j \in S_\ell} v_{i_1}(S_\ell - j) = v_{i_1}(S_h - g_h)$ by the choice of h . Thus, $(i_1, S_h) \in E^{(t)}$ follows.

Note that $\mathcal{L}^{(t+1)} = \mathcal{L}^{(t)} \cup \{h\}$. Consider the 1/2-EFX feasibility graph in the $(t+1)$ -st iteration. Since all bundles different from $S'_h := S_h - g_h$ remained unchanged, for every edge $(i, S_k) \in E^{(t)}$ with $k \neq h$ it follows that $(i, S_k) \in E^{(t+1)}$. According to Definition 3.4, $(i_1, S'_h) \in E^{(t+1)}$. Let us define ρ' as

$$\rho'(i) := \begin{cases} S'_h & \text{if } i = i_1, \\ \rho^{(t)}(i) & \text{if } i \neq i_1 \text{ and } \rho^{(t)}(i) \neq S_h, \\ \perp & \text{otherwise.} \end{cases}$$

By the above, this gives a matching in $E^{(t+1)}$, and it matches all bundles in $\mathcal{L}^{(t+1)} = \mathcal{L}^{(t)} \cup \{h\}$. \square

Lemma 3.8. *The alternating path P , as described in line 6 of Algorithm 4, exists.*

Proof. Since i_1 is an unmatched agent and the requirement that $|\{i : \rho(i) = S_i\}|$ is maximized subject to all trimmed down bundles being matched in the maximum matching ρ in line 6, we must have $(i_1, S_{i_1}) \notin E$. If $\rho(i_2) = S_{i_1}$ for an agent $i_2 \in A$, then we continue with S_{i_2} , otherwise we stop. Continuing this way, we eventually reach either $S_{i_\ell} = S_h$ or an unmatched bundle $S_{i_\ell} \neq S_h$. \square

Lemma 3.9. *If Algorithm 4 returns an allocation \mathcal{R} in line 15, then $\text{NSW}(\mathcal{R}) \geq \frac{1}{2} \text{NSW}(\mathcal{T})$ and \mathcal{R} is 1/2-EFX. If it returns \mathcal{R} in line 13, then $\text{NSW}(\mathcal{R}) \geq \text{NSW}(\mathcal{T})$ and $\cup_i R_i \subsetneq \cup_i T_i$.*

Proof. Let us start with the case when a perfect matching $\mathcal{R} = (\rho(i))_{i \in A}$ is returned in line 15. The 1/2-EFX property follows by Lemma 3.6. Let us show $\text{NSW}(\mathcal{R}) \geq \frac{1}{2} \text{NSW}(\mathcal{T})$.

Throughout the algorithm, $v_i(S_i) \geq \frac{1}{2} v_i(T_i)$ is maintained according to the condition on bundle trimming. By Claim 3.5, either $R_i = S_i$, or $v_i(R_i) > 2v_i(S_i)$. Therefore, we have

$$\forall i : v_i(R_i) \geq v_i(S_i) \geq \frac{1}{2} v_i(T_i). \quad (7)$$

Consequently,

$$\text{NSW}(\mathcal{R}) = \prod_{i \in A} v_i(R_i)^{1/n} \geq \frac{1}{2} \cdot \prod_{i \in A} v_i(T_i)^{1/n} \geq \frac{1}{2} \text{NSW}(\mathcal{T}).$$

Consider now the case when the algorithm terminated with \mathcal{R} in line 13. We need to show $\text{NSW}(\mathcal{R}) \geq \text{NSW}(\mathcal{T})$ and $\cup_i R_i \subsetneq \cup_i T_i$. Two cases here depend on whether $S_{i_\ell} = S_h$ or $S_{i_\ell} (\neq S_h)$ is an unmatched bundle. For the first case, we have

$$\begin{aligned} v_{i_f}(R_{i_f}) &= v_{i_f}(S_{i_{f-1}}) > 2v_{i_f}(S_{i_f}) \geq v_{i_f}(T_{i_f}), \quad \forall f \in \{2, \dots, \ell\}, \\ v_{i_1}(R_{i_1}) &= v_{i_1}(S_h - g_h) > 2v_{i_1}(S_{i_1}) \geq v_{i_1}(T_{i_1}) \end{aligned}$$

This implies

$$\text{NSW}(\mathcal{R}) = \prod_{i \in A} v_i(R_i)^{1/n} > \prod_{i \in A} v_i(T_i)^{1/n} = \text{NSW}(\mathcal{T}).$$

Since we do not assign g_h to any agent in \mathcal{R} , we must have $\cup_i R_i \subsetneq \cup_i T_i$.

For the second case, since S_{i_ℓ} is an unmatched bundle in ρ by the choice of the path P , we have $S_{i_\ell} \notin \mathcal{L}$ by the requirements on ρ . That is, $S_{i_\ell} = T_{i_\ell}$. By Claim 3.5, we have

$$\begin{aligned} v_{i_\ell}(R_{i_\ell}) &= v_{i_\ell}(S_{i_{\ell-1}}) > 2v_{i_\ell}(S_{i_\ell}) = 2v_{i_\ell}(T_{i_\ell}), \\ v_{i_f}(R_{i_f}) &= v_{i_f}(S_{i_{f-1}}) > 2v_{i_f}(S_{i_f}) \geq v_{i_f}(T_{i_f}), \quad \forall f \in \{2, \dots, \ell-1\}, \\ v_{i_1}(R_{i_1}) &= v_{i_1}(S_h - g_h) > 2v_{i_1}(S_{i_1}) \geq v_{i_1}(T_{i_1}) \\ v_h(R_h) &= v_h(T_h \setminus (S_h - g_h)) > \frac{1}{2}v_h(T_h). \end{aligned} \tag{8}$$

The last inequality follows from subadditivity using $v_h(T_h) \leq v_h(S_h - g_h) + v_h(T_h \setminus (S_h - g_h)) \leq \frac{1}{2}v_h(T_h) + v_h(T_h \setminus (S_h - g_h))$. Using (8), we get

$$\text{NSW}(\mathcal{R}) = \prod_{i \in A} v_i(R_i)^{1/n} > \prod_{i \in A} v_i(T_i)^{1/n} = \text{NSW}(\mathcal{T}).$$

Finally, since R we do not assign items in T_{i_ℓ} to any agent in \mathcal{R} , we must have $\cup_i R_i \subsetneq \cup_i T_i$. Note that if $T_{i_\ell} = \emptyset$, then $\text{NSW}(\mathcal{T}) = 0$ and $R_i = \emptyset, \forall i$ is a suitable output of type (ii). \square

4 A constant factor FPT algorithm for the asymmetric case

In this section, we prove the fixed parameter tractability result; the corresponding algorithm is Algorithm 5.

Theorem 1.4. *For any $\varepsilon > 0$, there is a deterministic $(6\varepsilon + \varepsilon)$ -approximation algorithm for the asymmetric Nash social welfare problem with submodular valuations, where the number of arithmetic operations and value oracle calls is $2^{O(n \log n)} \text{poly}(m, 1/\varepsilon)$.*

We start by identifying a set L_i of $3n$ largest-value singletons for each agent i , and consider the set $L = \bigcup_{i \in A} L_i$. Note that $|L| = O(n^2)$ and hence any enumeration over items in L takes time depending on n only. For each guess $(T_i)_{i \in A}$ with $|T_i| = \lceil w_i n \rceil + 1, i \in A$, we perform the local search procedure (Algorithm 2) to allocate the rest of the items with respect to the modified valuation functions $\bar{v}(S) = v(T_i \cup S)$. The following lemma is the key to Theorem 1.4.

Lemma 4.1. *There exists a partial allocation \mathcal{T} of $\lceil w_i n \rceil + 1$ items $T_i \subset L$ to agent i such that*

(i) \mathcal{T} can be extended to a $1/6$ -approximate allocation.

Algorithm 5: FPT algorithm for the asymmetric submodular NSW problem

Input: Valuations $(v_i)_{i \in A}$, weights $w \in \mathbb{R}_{>0}^n$ such that $\sum_{i \in A} w_i = 1$, and $\varepsilon > 0$.

Output: Allocation $\mathcal{S} = (S_i)_{i \in A}$.

- 1 Let L_i be the $3n$ most valuable singletons for agent i and $L = \cup_{i \in A} L_i$
 - 2 Enumerate over all possible partial allocations $\mathcal{T} = (T_i)_{i \in A}$, where $T_i \subset L$ and $|T_i| = \lceil w_i n \rceil + 1$
 - 3 Define $\bar{v}_i(S) := v_i(T_i \cup S)$
 - 4 $\mathcal{R} = (R_i)_{i \in A} := \text{LocalSearch}(G \setminus \cup T_i, (\bar{v}_i)_{i \in A})$
 - 5 Return $\mathcal{S} = (R_i \cup T_i)_{i \in A}$ that has the highest NSW value among all choices of \mathcal{T} .
-

(ii) For each agent i , every singleton outside $\cup_{i \in A} T_i$ has value at most $v_i(T_i)$.

Proof. Let us call a partial allocation *good* if it can be extended to an allocation that is a $1/3$ -approximation in Nash social welfare. We first show that there exists a good allocation assigning exactly $\lceil w_i n \rceil$ items $H'_i \subset L$ to each agent i . Let (S_1^*, \dots, S_n^*) be an optimal allocation for maximizing NSW.

Let us fix an arbitrary subset $L' \subset L$ of cardinality $|L'| = 2n$, and define $Q_i := L' \cap S_i^*$. Let $H_i \subseteq Q_i$ be a subset of $\min\{|Q_i|, \lceil w_i n \rceil\}$ items such that $v_i(H_i \cup (S_i^* \setminus L'))$ is maximized. Note that by submodularity, $v_i(H_i \cup (S_i^* \setminus L')) \geq \min\left\{\frac{\lceil w_i n \rceil}{|Q_i|}, 1\right\} \cdot v_i(S_i^*)$. Also, we can extend the sets H_i arbitrarily to disjoint sets $H'_i \subset L'$ such that $|H'_i| = \lceil w_i n \rceil$ (which is possible since $\sum_{i \in A} \lceil w_i n \rceil \leq 2n$). The ratio of NSW between the optimum and the allocation $(\bar{S}_1, \dots, \bar{S}_n)$ where $\bar{S}_i := H'_i \cup (S_i^* \setminus L')$ is

$$\prod_{i \in A} \left(\frac{v_i(S_i^*)}{v_i(\bar{S}_i)} \right)^{w_i} \leq \prod_{i \in A} \left(\max \left\{ \frac{|Q_i|}{\lceil w_i n \rceil}, 1 \right\} \right)^{w_i} \leq \prod_{i \in A} \left(\frac{|Q_i|}{w_i n} + 1 \right)^{w_i} \leq \sum_{i \in A} w_i \left(\frac{|Q_i|}{w_i n} + 1 \right) \leq 3$$

using the AM-GM inequality as well as $\sum_{i \in A} |Q_i| \leq 2n$ and $\sum_{i \in A} w_i = 1$. This proves that a good partial allocation of $\lceil w_i n \rceil$ items from L to each agent i exists. Among all such good partial allocations, pick the one which has maximum NSW value on its own, and call it $\mathcal{T}' = (T'_1, \dots, T'_n)$.

Next, extend T'_1, \dots, T'_n one-by-one to sets $T_i = T'_i + \ell_i$, by including greedily the most valuable available singleton in $\ell_i \in L_i \setminus \cup_{i \in A} T'_i$, so that these items are distinct for each agent. Consequently, the sets T_i are disjoint. This is possible since $|L_i \setminus \cup T'_i| \geq 3n - \sum \lceil w_i n \rceil \geq n$, and hence there is always an additional item in L_i available for every agent. The extended solution T_i contains T'_i and also each remaining item in $L \setminus \cup_{i \in A} T_i$ is at most as valuable as ℓ_i for agent i . We prove that \mathcal{T} satisfies the requirements of the lemma.

Let $(\bar{S}_1, \dots, \bar{S}_n)$ be a good extension of the allocation \mathcal{T}' as above (achieving a $1/3$ -approximation in NSW), and let $S'_i := (\bar{S}_i \cap L) \setminus T'_i$ be the additional items that agent i receives in L . Recall that we added a large singleton $\ell_i \in L_i$ to each T'_i to obtain T_i . Let us remove these items from the respective sets S'_i , to obtain disjoint sets $T_i \cup S_i$. Suppose that $|S'_i \setminus S_i| = s_i$, i.e. agent i lost s_i items in this step. Since we selected one large singleton for each agent, we have $\sum_{i \in A} s_i \leq n$.

\mathcal{T}' was chosen as an NSW-maximizing good allocation; hence we claim that T'_i is the most valuable choice of $\lceil w_i n \rceil$ items inside \bar{S}_i . If not, replacing T'_i by a more valuable such subset T''_i would yield an allocation which is still good (T''_i being a subset of \bar{S}_i) and achieving a higher value of NSW. Therefore, T'_i is the most valuable such subset of cardinality $\lceil w_i n \rceil$, in particular inside

$T'_i \cup (S'_i \setminus S_i)$, and by fractional subadditivity of submodular functions²

$$\frac{v_i(T'_i \cup (S'_i \setminus S_i))}{v_i(T'_i)} \leq 1 + \frac{|S'_i \setminus S_i|}{|T'_i|} \leq 1 + \frac{s_i}{w_i n}.$$

Thus we obtain

$$\frac{v_i(\bar{S}_i)}{v_i(T_i \cup S_i \cup (\bar{S}_i \setminus L))} \leq \frac{v_i(T'_i \cup S_i \cup (S'_i \setminus S_i) \cup (\bar{S}_i \setminus L))}{v_i(T'_i \cup S_i \cup (\bar{S}_i \setminus L))} \leq \frac{v_i(T'_i \cup (S'_i \setminus S_i))}{v_i(T'_i)} \leq 1 + \frac{s_i}{w_i n}.$$

where in the second inequality we repeatedly applied Proposition 2.3 to remove elements of set $S_i \cup (\bar{S}_i \setminus L)$ from the numerator and denominator at the same.

From here, we can conclude

$$\prod_{i \in A} \left(\frac{v_i(\bar{S}_i)}{v_i(T_i \cup S_i \cup (\bar{S}_i \setminus L))} \right)^{w_i} \leq \prod_{i \in A} \left(1 + \frac{s_i}{w_i n} \right)^{w_i} \leq \sum_{i \in A} w_i \left(1 + \frac{s_i}{w_i n} \right) \leq 2$$

by AM-GM. Combining this with the bound $\prod_{i \in A} \left(\frac{v_i(S_i^*)}{v_i(S_i)} \right)^{w_i} \leq 3$, we conclude that \mathcal{T} can be extended to a $1/6$ -approximate allocation $(T_1 \cup S_1 \cup (\bar{S}_1 \setminus L), \dots, T_n \cup S_n \cup (\bar{S}_n \setminus L))$.

Also, T_i contains a large singleton which dominates all items outside of $\bigcup_{i \in A} T_i$, and hence any item outside of $\bigcup_{i \in A} T_i$ has value at most $v_i(T_i)$ for agent i . This proves the lemma. \square

To derive Theorem 1.4, let us consider the local search algorithm, where each agent i receives the set T_i as an initial endowment, and these items will never be removed from them. We know that there is an allocation of the remaining items which gives a $1/6$ -approximation in NSW. By a proof similar to that of Proposition 2.7, one can show that the local search with endowed valuation $\bar{v}_i(S) = v_i(S \cup T_i)$ returns a $(e + \varepsilon)$ -approximation with respect to the best solution extending (T_1, \dots, T_n) . Combining the two statements, we obtain a $(6e + \varepsilon)$ -approximation.

5 Conclusion

We have shown a $(4 + \varepsilon)$ -approximation algorithm for the symmetric NSW problem with submodular valuations, which is the largest natural class of valuations that allows a constant-factor approximation (using value queries) even for utilitarian social welfare. Moreover, our algorithm gives an $e(2 + nw_{\max} + \varepsilon)$ -approximation algorithm for the asymmetric NSW problem under submodular valuations. However, there are still several directions and open problems to explore. An obvious one is to improve the approximation ratio for the symmetric case. The current hardness of approximation stands at $\frac{e}{e-1} \simeq 1.58$ for submodular valuations, which is the same as the optimal factor for maximizing utilitarian social welfare. It would be interesting to prove a separation between the two optimization objectives for submodular valuations.

As noted in the introduction, Feng and Li [29] recently gave an $e^{1/e} + \varepsilon$ -approximation for asymmetric NSW with additive valuations. However, the problem remains open for submodular valuations. The simplest case not covered by our algorithm is when one agent has weight $1/2$ and all other agents have weight $1/2n$.

There are several open questions on the existence of EFX and its relaxations for submodular valuations. A significant one is: Does a (complete) α -EFX allocation exist for $\alpha > 1/2$? Here, we do not impose any efficiency requirements.

²Every monotone submodular function f is also fractionally subadditive, which means in particular that $\frac{1}{\binom{a}{b}} \sum_{A \subset B, |A|=a} f(A) \geq \frac{a}{b} f(B)$, whenever $1 \leq a < b = |B|$. This implies that $\max_{A \subset B, |A|=a} f(A) \geq \frac{a}{b} f(B)$. See e.g. [27] for more details on the relationship between submodularity and fractional subadditivity.

Acknowledgements

We are grateful to Tomasz Ponitka for providing an improved version of Lemma 2.12 (previous version had factor 6 instead of 4) and for pointing out a couple of minor issues in Section 3 of the previous version that have been corrected in this paper. We also thank Bento Natura for the improved bound included in Remark 2.15.

References

- [1] G. Amanatidis, G. Birmpas, A. Filos-Ratsikas, A. Hollender, and A. A. Voudouris. Maximum Nash welfare and other stories about EFX. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 24–30, 2020.
- [2] N. Anari, S. O. Gharan, A. Saberi, and M. Singh. Nash social welfare, matrix permanent, and stable polynomials. In *Proceedings of the 8th Innovations in Theoretical Computer Science Conference (ITCS)*, 2017.
- [3] N. Anari, T. Mai, S. O. Gharan, and V. 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. SIAM, 2018.
- [4] N. Bansal and M. Sviridenko. The Santa Claus problem. In *Proceedings of the 38th ACM Symposium on Theory of Computing (STOC)*, pages 31–40, 2006.
- [5] J. B. Barbanel. *The geometry of efficient fair division*. Cambridge University Press, 2005.
- [6] S. Barman, U. Bhaskar, A. Krishna, and R. G. Sundaram. Tight approximation algorithms for p-mean welfare under subadditive valuations. In *28th Annual European Symposium on Algorithms (ESA)*, volume 173, pages 11:1–11:17, 2020.
- [7] S. Barman, A. Krishna, P. Kulkarni, and S. Narang. Sublinear approximation algorithm for Nash social welfare with XOS valuations. In *Proceedings of the 15th Innovations in Theoret. Computer Science (ITCS)*, 2024.
- [8] S. Barman, S. K. Krishnamurthy, and R. Vaish. Finding fair and efficient allocations. In *Proceedings of the 2018 ACM Conference on Economics and Computation (EC)*, pages 557–574, 2018.
- [9] S. Barman and M. Suzuki. Compatibility of fairness and Nash welfare under subadditive valuations. *arXiv preprint: arXiv:2407.12461*, 2024.
- [10] B. Berger, A. Cohen, M. Feldman, and A. Fiat. Almost full EFX exists for four agents. In *Proceedings of the 36th Conf. Artif. Intell. (AAAI)*, 2022.
- [11] S. J. Brams and A. D. Taylor. *Fair Division: From cake-cutting to dispute resolution*. Cambridge University Press, 1996.
- [12] F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia, editors. *Handbook of Computational Social Choice*. Cambridge University Press, 2016.
- [13] A. Brown, A. Laddha, M. R. Pittu, and M. Singh. Approximation algorithms for the weighted Nash social welfare via convex and non-convex programs. In *Proceedings of the 35th Symposium on Discrete Algorithms (SODA)*, pages 1307–1327, 2024.

- [14] E. Budish. The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes. *J. Political Economy*, 119(6):1061–1103, 2011.
- [15] I. Caragiannis, N. Gravin, and X. Huang. Envy-freeness up to any item with high Nash welfare: The virtue of donating items. In *Proceedings of the 2019 ACM Conference on Economics and Computation (EC)*, pages 527–545. ACM, 2019.
- [16] I. Caragiannis, D. Kurokawa, H. Moulin, A. D. Procaccia, N. Shah, and J. Wang. The unreasonable fairness of maximum Nash welfare. *ACM Transactions on Economics and Computation (TEAC)*, 7(3):1–32, 2019.
- [17] S. Chae and H. Moulin. Bargaining among groups: an axiomatic viewpoint. *International Journal of Game Theory*, 39(1-2):71–88, 2010.
- [18] B. R. Chaudhury, Y. K. Cheung, J. Garg, N. Garg, M. Hoefer, and K. Mehlhorn. Fair division of indivisible goods for a class of concave valuations. *J. Artif. Intell. Res.*, 74:111–142, 2022.
- [19] B. R. Chaudhury, J. Garg, and K. Mehlhorn. EFX exists for three agents. In *Proceedings of the 21st Conf. Econom. Comput. (EC)*, pages 1–19. ACM, 2020.
- [20] B. R. Chaudhury, J. Garg, and R. Mehta. Fair and efficient allocations under subadditive valuations. In *Proceedings of the AAI Conference on Artificial Intelligence*, pages 5269–5276, 2021.
- [21] B. R. Chaudhury, T. Kavitha, K. Mehlhorn, and A. Sgouritsa. A little charity guarantees almost envy-freeness. *SIAM J. Comput.*, 50(4):1336–1358, 2021.
- [22] R. Cole, N. Devanur, V. Gkatzelis, K. Jain, T. Mai, V. V. Vazirani, and S. Yazdanbod. Convex program duality, Fisher markets, and Nash social welfare. In *Proceedings of the 2017 ACM Conference on Economics and Computation (EC)*, pages 459–460, 2017.
- [23] R. Cole and V. Gkatzelis. Approximating the Nash social welfare with indivisible items. In *Proceedings of the 47th ACM Symposium on Theory of Computing (STOC)*, pages 371–380. ACM, 2015.
- [24] D. M. Degefu, W. He, L. Yuan, and J. H. Zhao. Water allocation in transboundary river basins under water scarcity: a cooperative bargaining approach. *Water resources management*, 30(12):4451–4466, 2016.
- [25] S. Dobzinski, W. Li, A. Rubinstein, and J. Vondrák. A constant factor approximation for Nash social welfare with subadditive valuations. In *Proceedings of the 56th Symp. Theory of Computing (STOC)*, 2024.
- [26] S. Dobzinski, N. Nisan, and M. Schapira. Approximation algorithms for combinatorial auctions with complement-free bidders. *Mathematics of Operations Research*, 35(1):1–13, 2010.
- [27] U. Feige. On maximizing welfare when utility functions are subadditive. *SIAM Journal on Computing (SICOMP)*, 39(1):122–142, 2009.
- [28] M. Feldman, S. Mauras, and T. Ponitka. On optimal tradeoffs between EFX and Nash welfare. In *Proceedings of the 38th Conf. Artif. Intell. (AAAI)*, pages 9688–9695, 2024.

- [29] Y. Feng and S. Li. A Note on Approximating Weighted Nash Social Welfare with Additive Valuations. In *Proceedings of the 51st Intl. Coll. Automata, Languages and Programming (ICALP)*, pages 63:1–63:9, 2024.
- [30] H. Fu, R. Kleinberg, and R. Lavi. Conditional equilibrium outcomes via ascending price processes with applications to combinatorial auctions with item bidding. In *Proceedings of the 13th ACM Conference on Electronic Commerce (EC)*, page 586, 2012.
- [31] J. Garg, M. Hoefer, and K. 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. SIAM, 2018.
- [32] J. Garg, M. Hoefer, and K. Mehlhorn. Satiation in Fisher markets and approximation of Nash social welfare. *Math. Oper. Res.*, 49(2):1109–1139, 2024.
- [33] J. Garg, E. Husic, W. Li, L. A. Vegh, and J. Vondrak. Approximating Nash social welfare by matching and local search. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing (STOC)*, pages 1298–1310, 2023.
- [34] J. Garg, E. Husic, and L. A. Vegh. Approximating Nash social welfare under Rado valuations. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 1412–1425, 2021.
- [35] J. Garg, P. Kulkarni, and R. Kulkarni. Approximating Nash social welfare under submodular valuations through (un)matchings. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2673–2687, 2020.
- [36] H. H. V. der Laan G, and Z. Y. Asymmetric Nash solutions in the river sharing problem. *Strategic Behavior and the Environment*, 4:321–360, 2014.
- [37] J. C. Harsanyi and R. Selten. A generalized Nash solution for two-person bargaining games with incomplete information. *Management science*, 18(5-part-2):80–106, 1972.
- [38] E. Kalai. Nonsymmetric Nash solutions and replications of 2-person bargaining. *International Journal of Game Theory*, 6(3):129–133, 1977.
- [39] M. Kaneko and K. Nakamura. The Nash social welfare function. *Econometrica: Journal of the Econometric Society*, pages 423–435, 1979.
- [40] F. Kelly. Charging and rate control for elastic traffic. *European transactions on Telecommunications*, 8(1):33–37, 1997.
- [41] A. Laruelle and F. Valenciano. Bargaining in committees as an extension of Nash’s bargaining theory. *Journal of Economic Theory*, 132(1):291–305, 2007.
- [42] E. Lee. APX-hardness of maximizing Nash social welfare with indivisible items. *Information Processing Letters*, 122:17–20, 2017.
- [43] W. Li and J. Vondrak. Estimating the Nash social welfare for coverage and other submodular valuations. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1119–1130. SIAM, 2021.

- [44] W. Li and J. Vondrák. A constant-factor approximation algorithm for Nash social welfare with submodular valuations. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 25–36, 2022.
- [45] R. J. Lipton, E. Markakis, E. Mossel, and A. Saberi. On approximately fair allocations of indivisible goods. In *Proceedings of the 5th Conf. Econom. Comput. (EC)*, pages 125–131. ACM, 2004.
- [46] R. Mahara. Extension of additive valuations to general valuations on the existence of EFX. In *ESA*, volume 204, pages 66:1–66:15, 2021.
- [47] H. Moulin. *Fair division and collective welfare*. MIT press, 2004.
- [48] J. F. Nash. The bargaining problem. *Econometrica: Journal of the econometric society*, pages 155–162, 1950.
- [49] B. Natura. Personal communication.
- [50] B. Plaut and T. Roughgarden. Almost envy-freeness with general valuations. In *Proceedings of the 29th Symposium on Discrete Algorithms (SODA)*, pages 2584–2603. SIAM, 2018.
- [51] A. D. Procaccia. An answer to fair division’s most enigmatic question: technical perspective. *Commun. ACM*, 63(4):118, 2020.
- [52] J. Robertson and W. Webb. *Cake-cutting algorithms: Be fair if you can*. CRC Press, 1998.
- [53] J. Rothe, editor. *Economics and Computation, An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division*. Springer, 2016.
- [54] H. R. Varian. Equity, envy, and efficiency. *Journal of Economic Theory*, 9(1):63–91, 1974.
- [55] H. P. Young. *Equity: in theory and practice*. Princeton University Press, 1995.
- [56] S. Yu, E. 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.