Approximating One-Sided and Two-Sided Nash Social Welfare With Capacities

Salil Gokhale
IIT Delhi
New Delhi, India
mt1210237@iitd.ac.in

Harshul Sagar
IIT Delhi
New Delhi, India
ee1210211@ee.iitd.ac.in

Rohit Vaish
IIT Delhi
New Delhi, India
rvaish@iitd.ac.in

Jatin Yadav
IIT Delhi
New Delhi, India
csz237549@cse.iitd.ac.in

ABSTRACT

We study the problem of maximizing *Nash social welfare*, which is the geometric mean of agents' utilities, in two well-known models. The first model involves *one-sided* preferences, where a set of indivisible items is allocated among a group of agents (commonly studied in fair division). The second model deals with *two-sided* preferences, where a set of workers and firms, each having numerical valuations for the other side, are matched with each other (commonly studied in matching-under-preferences literature). We study these models under *capacity constraints*, which restrict the number of items (respectively, workers) that an agent (respectively, a firm) can receive. We contribute constant-factor approximation algorithms for both problems under a broad class of valuations. Specifically, our main results are the following: (a) For any $\varepsilon > 0$, a $(6 + \varepsilon)$-approximation algorithm for the one-sided problem when agents have *submodular* valuations, and (b) a 1.33-approximation algorithm for the two-sided problem when the firms have *subadditive* valuations. The former result provides the first constant-factor approximation algorithm for Nash welfare in the one-sided problem with submodular valuations and capacities, while the latter result significantly improves upon an existing $\sqrt{\mathsf{OPT}}$-approximation algorithm for additive valuations. Our result for the two-sided setting also establishes a computational separation between the Nash and utilitarian welfare objectives. We also complement our algorithms with hardness-of-approximation results.

KEYWORDS

Nash Social Welfare;Approximation Algorithms;Capacity Constraints;Fair Division;Matching Under Preferences

ACM Reference Format:
Salil Gokhale, Harshul Sagar, Rohit Vaish, and Jatin Yadav. 2025. Approximating One-Sided and Two-Sided Nash Social Welfare With Capacities. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 9 pages.

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

1 INTRODUCTION

Fairness and efficiency are quintessential requirements in many resource allocation problems, such as distributing a set of items among agents and assigning job applicants to positions. These objectives are often represented as opposite ends of the "collective utility" scale [33]: On one end, there is Bentham's *utilitarian* welfare, which maximizes the sum of individual utilities and represents a purely efficient outcome. On the other end, there is Rawls' *egalitarian* welfare (and its leximin refinement), which captures perfect fairness by maximizing the worst-off individual's utility.

The *Nash social welfare* [27, 34], defined as the geometric mean of agents' utilities, strikes a remarkable balance between fairness and efficiency. It provides a "sweet spot" between these seemingly incompatible objectives and satisfies several desirable properties such as scale invariance, Pigou-Dalton principle, (approximate) envy-freeness, and Pareto optimality [10, 33]. Due to its strong axiomatic appeal, the computational aspects of Nash welfare have received significant attention [1, 6, 10, 14, 16, 21, 22, 26, 30, 35].

In this work, we focus on the problem of maximizing Nash social welfare in two well-studied models of resource allocation:

- *One-sided* preferences: In this model, a set of indivisible items is allocated among a set of agents. Each agent has combinatorial preferences over the items, while each item can be assigned to exactly one agent. This setting is commonly explored in the literature on fair division with indivisible items [2, 9, 32].
- *Two-sided* preferences: In this model, a set of workers is matched with a set of firms. Each firm can be matched with multiple workers and has combinatorial preferences over them. Each worker has cardinal valuations over the firms and can only be assigned to one firm. This model is frequently studied in the matching-under-preferences literature [9, 25, 29, 31, 38].

An important feature distinguishing our work from much of the prior work is the consideration of *capacity* constraints. In the context of the one-sided problem, this means restricting the number of items that an agent can receive, while in the two-sided setting, each firm is restricted to be matched with at most a certain number of workers. We allow different agents to have different capacities.

Capacity constraints naturally capture some of the practical limitations in resource allocation problems. For example, when distributing pieces of artwork among museums, the space limitation of each museum restricts the number of items it can accommodate [39].

Similarly, the hiring capacity of a firm is often limited by its financial budget. In such scenarios, it is natural to aim for outcomes that maximize welfare while adhering to capacity constraints.

It is known that maximizing Nash welfare is hard to approximate (specifically, it is APX-hard) in the one-sided problem even when there are no capacity constraints [30, 35]. In the two-sided model, the problem of maximizing Nash welfare is known to be NP-hard even when each firm has a constant capacity [26]. Since the uncapacitated setting is a special case of the capacitated model, the latter problem is computationally more challenging. Therefore, it is important to develop approximation algorithms for these problems.

Our Contributions

We present constant-factor approximation algorithms for maximizing Nash welfare in the one-sided and two-sided models under capacity constraints for a broad class of valuations. Our results are summarized below (also see Table 1). Throughout, we will use $\varepsilon > 0$ to denote an arbitrary constant.

- *One-sided model*: In Section 3, we provide a $(6+\varepsilon)$-approximation algorithm for maximizing Nash welfare under capacity constraints when the agents have monotone *submodular* valuations over the items. In the same setting without capacity constraints, a local search-based $(4 + \varepsilon)$-approximation algorithm was recently proposed by Garg et al. [21]. Although their algorithm does not automatically handle capacity constraints, we show that with some necessary modifications (such as allowing two-way exchange of items instead of only one-way transfers), the algorithm can be adapted to the more general problem with capacities, albeit with a slight loss in approximation quality. On the hardness front, Garg et al. [22] have shown that, unless P = NP, no algorithm that makes a polynomial number of value queries can provide a better approximation of Nash welfare than $\frac{e}{e-1} \approx 1.58$. This intractability holds even for a constant number of agents with submodular valuations and even without capacity constraints and, therefore, extends to the capacitated setting.
- *Two-sided model*: In Section 4, we present a 1.33-approximation algorithm for maximizing Nash welfare when the firms have capacity constraints and monotone *subadditive* valuations over the workers, and the workers have cardinal valuations over individual firms. Prior to our work, a $\sqrt{\texttt{OPT}}$-approximation for positive additive valuations was known, where $\texttt{OPT}$ denotes the optimal Nash welfare [26]. Thus, our result presents a significant improvement over the current algorithm in both approximation quality and the generality of preferences. Notably, our result also establishes that in the two-sided setting, Nash welfare is computationally *easier* than utilitarian welfare for which a hardness result of $\frac{e}{e-1} - \varepsilon$ is known [28]. Our algorithm and its analysis are simple and use a single minimum cost flow computation. We also show that the same algorithm can be used to provide a PTAS when the number of firms is constant.
- *Hardness results*: In Section 5, we show that maximizing Nash welfare in the two-sided problem is APX-hard; specifically, it is NP-hard to approximate Nash welfare within a factor of 1.0000759 even without capacity constraints. This observation strengthens an existing intractability result which only shows NP-hardness [26]. In light of the aforementioned PTAS, it follows that our APX-hardness result cannot be extended to the case of a constant number of firms unless P = NP.

All missing proofs and other technical details can be found in the full version [24].

Related Work

We will now review some of the relevant literature on one-sided and two-sided Nash welfare. Due to space constraints, the detailed coverage of related work is deferred to the full version [24].

One-sided preferences. In the one-sided model without capacity constraints, it is known that maximizing Nash social welfare is APX-hard even under additive valuations [18, 20, 30, 35]. A substantial body of work has studied the design of exact and approximation algorithms for Nash welfare over various classes of valuations, including *additive* [1, 6, 7, 13, 14], *budget-additive* [19], *separable piecewise-linear concave* [3, 11], *matroid rank* [4, 8, 40], *submodular* [21, 22], and *subadditive* [5, 12, 15]. These valuation classes have been extensively studied in algorithmic game theory [36] and computational social choice [9]. In particular, additive valuations are attractive from an elicitation perspective, while submodular valuations capture the idea of diminishing marginal returns.

For submodular valuations, Garg et al. [21] have shown a $(4+\varepsilon)$-approximation algorithm in the *value query* model (see Section 2 for the definition). In the same setting, it is known that any algorithm that makes a polynomial number of value queries fails to provide better than $\frac{e}{e-1} \approx 1.58$ approximation [22]. For the case of a constant number of agents, Garg et al. [22] provide an $\frac{e}{e-1}$-approximation algorithm, matching the hardness threshold.

Two-sided preferences. Jain and Vaish [26] recently initiated the study of Nash welfare in the two-sided setting. They examined the computational complexity of maximizing Nash welfare when firms have additive valuations over the workers and are limited by capacity constraints, while workers have cardinal valuations for the individual firms. They proved that the problem is NP-hard even when each firm's capacity is at most 2. However, when each firm's capacity is 1, a simple matching computation is sufficient. The authors also presented a $\sqrt{\texttt{OPT}}$-approximation algorithm for the two-sided Nash welfare under positive additive valuations, where $\texttt{OPT}$ represents the optimal objective value for the given input.

2 PRELIMINARIES

For any $r \in \mathbb{N}$, let $[r] \coloneqq \{1, 2, \dots, r\}$. For any set R and a singleton $\{j\}$, we use $R + j$ to denote $R \cup \{j\}$ and $R - j$ to denote $R \setminus \{j\}$.

One-Sided Model

Problem instance. An instance of the one-sided problem is defined by a tuple $\langle N, M, \mathcal{V}, C\rangle$, where N is a set of n *agents*, M is a set of m *items* (or goods), $\mathcal{V} = \{v_1, \dots, v_n\}$ is a set of *valuation functions* and $C = (c_1, \dots, c_n)$ is a vector of *capacities*. The valuation function $v_i : 2^M \to \mathbb{Q}_{\geq 0}$ specifies agent i's value $v_i(S)$ for any subset $S \subseteq M$ of items. Note that we assume the valuations to be nonnegative rational numbers. The capacity $c_i \in \mathbb{Z}_{>0}$ is a positive integer that represents the maximum number of items that agent i can receive. We will write $v_i(j)$ instead of $v_i\{j\})$ for a singleton $\{j\}$.

One-sided	Nash		Utilitarian	
	Hardness	Algorithm	Hardness	Algorithm
Without Capacities	$\frac{e}{e-1} - \varepsilon$ [22, 28]	$4 + \varepsilon$ [21]	$\frac{e}{e-1} - \varepsilon$ [28]	$\frac{e}{e-1}$ [41]
With Capacities	$\frac{e}{e-1} - \varepsilon$ [22, 28]	$6 + \varepsilon$ (Theorem 3.1)	$\frac{e}{e-1} - \varepsilon$ [28]	$\frac{e}{e-1}$ [41]

Two-sided	Nash		Utilitarian	
	Hardness	Algorithm	Hardness	Algorithm
Without Capacities	1.0000759 (Theorem 5.1)	1.33 (subadd.) (Theorem 4.1)	$\frac{e}{e-1} - \varepsilon$ [28]	$\frac{e}{e-1}$ [41]
With Capacities	1.0000759 (Theorem 5.1)	1.33 (subadd.) (Theorem 4.1)	$\frac{e}{e-1} - \varepsilon$ [28]	$\frac{e}{e-1}$ [41]

Table 1: Summary of results on maximizing Nash social welfare for the one-sided (left) and two-sided (right) problems under *submodular* valuations. The rows specify whether or not capacities are considered. The columns specify the best-known approximation algorithms and hardness results (assuming P ≠ NP) for Nash and utilitarian welfare. Our contributions are highlighted in shaded boxes. Note that our 1.33-approximation algorithm for the two-sided problem applies to the more general domain of *subadditive* valuations.

Allocation. A partial allocation $A = (A_1, A_2, \ldots, A_n)$ refers to an n-subpartition of the set of items M such that, for all $i \neq j$, $A_i \cap A_j = \emptyset$ and $\cup_{i=1}^{n} A_i \subseteq M$. Here, A_i represents the subset of items or the *bundle* assigned to agent i. A partial allocation is called *complete* if $\cup_{i=1}^{n} A_i = M$. We will use the term "allocation" to denote a complete allocation, and explicitly write "partial allocation" otherwise. We will call a partial allocation *feasible* if, for each agent $i \in N$, we have $|A_i| \leq c_i$. Given a partial allocation A, we will call $v_i(A_i)$ the *utility* (or value) derived by agent i under A.

Nash social welfare. The Nash Social Welfare of a partial allocation A is defined as the geometric mean of the agents' utilities, i.e., $\text{NSW}(A) \coloneqq \left(\prod_{i \in N} v_i(A_i)\right)^{1/n}$. The Nash social welfare, or NSW for short, is known to be *scale-free*, which means that an allocation that maximizes Nash social welfare continues to do so even if agent i's valuation function $v_i(\cdot)$ is scaled by a factor $\alpha_i \geq 0$.

The computational problem associated with maximizing Nash social welfare, which we call Capacitated One-Sided NSW, is defined below. We will call a partial allocation *Nash optimal* if it maximizes the Nash social welfare among all feasible partial allocations for a given instance.

Capacitated One-Sided NSW	
Input:	An instance $\mathcal{I} = \langle N, M, \mathcal{V}, C \rangle$ where N is the set of agents, M is the set of items, $\mathcal{V}$ is the set of valuation functions, and C is a vector representing capacity of agents.
Goal:	Compute a feasible Nash optimal partial allocation A.

A special case of Capacitated One-Sided NSW is when there are no capacity constraints; equivalently, each agent's capacity is equal to the number of items. We call this problem Uncapacitated One-Sided NSW.

Valuation classes. A valuation function $v_i : 2^M \to \mathbb{Q}_{\geq 0}$ is said to be *monotone* if, for any pair of subsets $S, T \subseteq M$ such that $S \subseteq T$, we have $v_i(S) \leq v_i(T)$, and *normalized* if $v_i(\emptyset) = 0$. We will assume throughout that the valuations are monotone and normalized.

Various subclasses of monotone valuations will be of interest to us. Formally, a valuation function v_i is said to be:

- *additive* if, for any subset $S \subseteq M$, we have $v_i(S) = \sum_{j \in S} v_i(j)$,
- *submodular* if, for any subsets $S, T \subseteq M$ such that $S \subseteq T$ and any $j \notin T$, we have that $v(S \cup \{j\}) - v(S) \geq v(T \cup \{j\}) - v(T)$, and
- *subadditive* if, for any pair of subsets $S, T \subseteq M$, we have $v_i(S \cup T) \leq v_i(S) + v_i(T)$.

Observe that the containment relations among these classes of valuations are `additive` ⊆ `submodular` ⊆ `subadditive`.

Query model. Note that we allow for combinatorial valuations, which can have an exponential-sized representation in terms of the number of items. Therefore, when analyzing algorithms, it is natural to assume an oracle access to the valuations. We will focus on *value* queries in our work. Given as input a bundle S and an index i, a value query returns agent i's value for the bundle $v_i(S)$.

We will write $\text{poly}(n, m)$ to denote a polynomial function in n and m. We will be interested in designing algorithms that make $\text{poly}(n, m)$ number of value queries and have $\text{poly}(n, m)$ running time.

α-approximation algorithm. Given an instance $\mathcal{I}$, let $\texttt{ALG}(\mathcal{I})$ denote the allocation returned by a given algorithm ALG, and let $\texttt{OPT}(\mathcal{I})$ denote a Nash optimal allocation for $\mathcal{I}$. We say that ALG is α-approximate if, for all problem instances $\mathcal{I}$, we have that $\frac{\text{NSW}(\texttt{OPT}(\mathcal{I}))}{\text{NSW}(\texttt{ALG}(\mathcal{I}))} \leq \alpha$. Notice that $\alpha \geq 1$.

Two-Sided Model

Problem instance. An instance of the two-sided problem is defined by a tuple $\langle F, W, \mathcal{V}, \mathcal{W}, C \rangle$, where F is a set of n *firms*, W is a set of m *workers*, $\mathcal{V} = \{v_1, \ldots, v_n\}$ is a set of firms' valuation functions, $\mathcal{W} = \{w_1, \ldots, w_m\}$ is a set of workers' valuation functions, and $C = (c_1, \ldots, c_n)$ is a vector of capacities. The valuation function $v_i : 2^W \to \mathbb{Q}_{\geq 0}$ specifies firm i's value $v_i(S)$ for any subset $S \subseteq W$ of workers. The capacity $c_i \in \mathbb{Z}_{>0}$ is a positive integer that represents the maximum number of workers that firm i can be matched with. Every worker $j \in W$ has a valuation function $w_j : F \to \mathbb{Q}_{\geq 0}$, and $w_j(i)$ represents the value that worker j associates with firm i.

Many-to-one matching. Given an instance $\mathcal{I} = \langle F, W, \mathcal{V}, \mathcal{W}, C \rangle$, a *many-to-one matching* for $\mathcal{I}$, denoted by $\mu : F \times W \to \{0, 1\}$, is a function that assigns each worker-firm pair a weight of either 0 or 1 such that for every worker $j \in W$, $\sum_{i \in F} \mu(i, j) \leq 1$, and for every firm $i \in F$, $\sum_{j \in W} \mu(i, j) \leq c_i$. Further, we define $\mu_i \coloneqq \{j \in$

$W : \mu(i,j) = 1\}$ and $\mu_j := \{i \in F : \mu(i,j) = 1\}$ as the set of workers and firms matched with firm i and worker j, respectively.

Nash social welfare. The Nash Social Welfare of a many-to-one matching μ is defined as the geometric mean of the utilities of the firms and workers under μ, i.e.,

$$\text{NSW}(\mu) = \left(\prod_{i\in F} v_i(\mu_i) \prod_{j\in W} w_j(\mu_j)\right)^{\frac{1}{m+n}}.$$

Note that similar to the one-sided model, the two-sided Nash social welfare is also *scale-free*.

The computational problem associated with maximizing the two-sided Nash Social Welfare, which we call Capacitated Two-Sided NSW, is defined below. We will call a many-to-one matching *Nash optimal* if it maximizes the Nash social welfare among all feasible many-to-one matchings for a given instance.

Capacitated Two-Sided NSW	
Input:	An instance $\mathcal{I} = \langle F, W, \mathcal{V}, \mathcal{W}, C\rangle$ where F is the set of firms, W is the set of workers, $\mathcal{V}$ is the set of firms' valuation functions, $\mathcal{W}$ is the set of workers' valuation functions and C is a vector representing capacity of firms.
Goal:	Compute a feasible Nash optimal many-to-one matching μ.

A special case of Capacitated Two-Sided NSW is when there are no capacity constraints on the firms; equivalently, each firm's capacity is equal to the number of workers. We call this problem Uncapacitated Two-Sided NSW. The concepts of an α-approximation algorithm, query model, and valuation classes for the two-sided model are defined analogously to the one-sided setting. In the two-sided case, value queries can be made for both workers and firms.

The case of zero Nash welfare and inadequate capacities. In the Capacitated Two-Sided NSW problem, we will assume that the capacities are adequate, that is, $\sum c_i \geq m$. Note that there can be situations, in both the Capacitated One-Sided NSW and Capacitated Two-Sided NSW problems, where the optimal solution has zero Nash welfare. In this case however, an arbitrary feasible solution satisfies the required approximation ratio. Therefore, we will assume in the rest of the paper that the optimal solution has a nonzero Nash welfare.

3 APPROXIMATION ALGORITHM FOR THE ONE-SIDED MODEL

In this section, we will show that given any $\varepsilon > 0$, there is a $(6+\varepsilon)$-approximation algorithm for Capacitated One-Sided NSW under submodular valuations. Our algorithm and its analysis are obtained by modifying the algorithm of Garg et al. [21] that gives a $(4+\varepsilon)$-approximation under submodular valuations for the Uncapacitated One-Sided NSW problem, i.e., the one-sided Nash welfare maximization problem *without* capacity constraints.

Before discussing our algorithm, it is relevant to discuss some natural approaches that do not work.

Limitations of natural approaches. The algorithm of Garg et al. [21] is not designed to handle capacities and may return infeasible allocations. Additionally, it is possible to design a family of instances in which the optimal Nash welfare under capacities is arbitrarily smaller than the optimal Nash welfare without capacities. In such instances, a $(4+\varepsilon)$-approximation algorithm for the unconstrained problem would decidedly return an infeasible allocation.

Another common approach for handling constraints is by *modifying* the valuation functions. For example, an instance with *additive* valuations and capacity constraints can be converted to an unconstrained instance with *submodular* valuations via the following transformation: For any set S of items, define agent i's value $v_i(S)$ as the sum of values of the $\min\{c_i, |S|\}$ most valuable items in S, where c_i is the capacity for agent i. The resulting valuation function $v_i(\cdot)$ can be observed to be submodular. Consequently, the algorithm of Garg et al. [21], which gives a $(4+\varepsilon)$ approximation in the unconstrained instance, gives a similar approximation for the original additive valuations instance with capacity constraints. Unfortunately, when the given instance has *submodular* valuations with capacity constraints, such a transformation may not result in an unconstrained submodular instance, as demonstrated by the following example.

Example 1. *Let v be a submodular valuation function of an agent with capacity c. Let us define a new valuation function v' such that $v'(S) \coloneqq \max_{S'\subseteq S:|S'|\leq c} v(S')$. Then, v' can fail to be submodular. Consider a set of four items $M = \{w_1, w_2, w_3, w_4\}$ and say $c = 2$. Suppose the agent's values under $v(\cdot)$ are:*

- $v(\emptyset) = 0$,
- $v(\{w_2, w_4\}) = 4$ *and* $v(\{w_1, w_3, w_4\}) = 3$, *and*
- *for all other subsets* $S \subseteq M$, $v(S) = \min\{4, |S|+1\}$.

It can be observed that $v(\cdot)$ is submodular and monotone.

Note that $v'(\{w_1, w_3\}) = 3$ and $v'(\{w_1, w_3, w_4\}) = 3$, implying a marginal increase of 0 upon adding w_4. Similarly, $v'(\{w_1, w_2, w_3\}) = 3$ and $v'(\{w_1, w_2, w_3, w_4\}) = 4$, which gives a marginal increase of 1 upon adding w_4. Since adding w_4 to a smaller set results in a smaller marginal, we get that $v'(\cdot)$ is not submodular. □

Having motivated the challenge of approximating Nash welfare in the capacitated setting, let us now formally state our main result.

Theorem 3.1. *For any $\varepsilon > 0$, there exists a $(6+\varepsilon)$-approximation algorithm for* Capacitated One-Sided NSW *under submodular valuations that runs in* $\text{poly}(n,m)$ *time and makes a* $\text{poly}(n,m)$ *number of value queries.*

Remark (Approximate fairness). In the unconstrained setting under additive valuations, it is known that any Nash optimal allocation, say A, is *envy-free up to one item* (EF1) [10]. This property entails that for any pair of agents i, k, there exists some item $g \in A_j$ such that $v_i(A_i) \geq v_i(A_k \setminus \{g\})$. However, this implication breaks down under capacity constraints. Recently, Wu et al. [42] studied *budget constraints* in the one-sided problem, of which capacity constraints are a special case. They showed that under subadditive valuations, any budget-feasible α-approximate allocation for Nash welfare satisfies $\frac{1}{4\alpha}$-EF1. In the presence of capacity constraints, this property requires that for any pair of agents i, k and any subset $S \subseteq A_k$ such that $|S| \leq c_i$, there exists some item $g \in S$ such that $v_i(A_i) \geq \frac{1}{4\alpha} v_i(S \setminus \{g\})$. By combining their result with the guarantee in Theorem 3.1, we obtain that the allocation returned by our algorithm additionally satisfies $\frac{1}{(24+4\varepsilon)}$-EF1. □

Towards proving Theorem 3.1, we will solve a related problem, Exact Capacitated One-Sided NSW, in which each agent is assigned *exactly* as many items as its capacity. Formally, an allocation $A = (A_i)_{i \in N}$ is feasible for a given instance of Exact Capacitated One-Sided NSW if, for every $i \in N$, we have $|A_i| = c_i$. It is easy to construct an approximation-preserving reduction between the two problems by adding dummy items that do not affect the value of any subset of the original items. Thus, Theorem 3.1 follows from Lemma 3.2 below. Due to space limitations, we will defer some of the proofs to the supplementary material.

Lemma 3.2. *For any $\varepsilon > 0$, there exists a $(6 + \varepsilon)$ approximation algorithm for* Exact Capacitated One-Sided NSW *under submodular valuations that runs in* $\text{poly}(n, m)$ *time and makes* $\text{poly}(n, m)$ *number of value queries.*

The algorithm in Lemma 3.2 is a modification of that of Garg et al. [21] and works in three phases. The first phase involves computing a *one-to-one maximum weight matching* τ of the agents and items. We denote the set of items matched in this phase by H. Let $J \coloneqq M \setminus H$ be the set of remaining items. In the second phase, the items in J are allocated through *local search*. Crucially, we only allow for *two-way transfers* (or swaps) between pairs of agents to ensure that the capacity constraints are not violated during the course of the local search. In the final phase, the items in H are *rematched* to the agents. Note that the first and the third phases of our algorithm are identical to that of Garg et al. [21]. In the second phase, the algorithm of Garg et al. [21] performs local search via *one-way* transfer of items, whereas our algorithm uses *two-way* exchanges to maintain the capacity constraints.

In the following, we will define some terminology that will be used in describing our algorithm for Exact Capacitated One-Sided NSW.

Endowed valuation functions. In the local search phase of the algorithm, we do not use the actual valuation functions of the agents; instead, we use the *endowed* valuation functions. Let $\overline{N} \coloneqq \{i \in N : v_i(J) > 0\}$ be the set of agents that assign a positive value to the set J. For every agent $i \in \overline{N}$, we define the endowed valuation function $\overline{v}_i$ as $\overline{v}_i(S) \coloneqq v_i(S) + v_i(\ell(i))$ for any subset $S \subseteq M$, where $\ell(i)$ is the favourite item of agent i in J. Note that if v_i is submodular, then $\overline{v}_i$ is also submodular.

Accuracy parameter. Our local search subroutine will use the parameter $\overline{\varepsilon} = -1 + (1+\varepsilon)^{1/m}$ such that the minimum multiplicative increase in NSW required to perform a swap is $1 + \overline{\varepsilon}$.

Swaps in local search. Our local search phase uses swaps (or item exchanges) between agents whenever the swap provides a large enough increase in NSW. Note that two-way swaps maintain capacity constraints. Specifically, we will consider the following two kinds of swaps.

- Full Swap: Let $\mathcal{R} = (R_i)_{i \in \overline{N}}$ be a partial allocation of the items in J among the agents in $\overline{N}$. A Full Swap is said to exist if there is an item $j \in R_i$ allocated to an agent $i \in \overline{N}$ and an item $k \in R_{i'}$ allocated to an agent $i' \in \overline{N}$ such that $\left(\frac{\overline{v}_i(R_i - j + k)\overline{v}_{i'}(R_{i'} - k + j)}{\overline{v}_i(R_i)\overline{v}_{i'}(R_{i'})}\right)^{1/n} > 1 + \overline{\varepsilon}$. The algorithm performs this swap by allocating j to i' and k to i.
- Partial Swap: Let $\mathcal{R} = (R_i)_{i \in \overline{N}}$ be a partial allocation of the items in J among the agents in $\overline{N}$. A Partial Swap is said to exist if there is an unallocated item $k \in J$ and an item $j \in R_i$ allocated to an agent $i \in \overline{N}$ such that $\left(\frac{\overline{v}_i(R_i - j + k)}{\overline{v}_i(R_i)}\right)^{1/n} > 1 + \overline{\varepsilon}$. We perform this swap by allocating k to i and j is left unallocated.

$\overline{\varepsilon}$-local optimum. A partial allocation $\mathcal{R} = (R_i)_{i \in \overline{N}}$ of J to $\overline{N}$ is an $\overline{\varepsilon}$-local optimum with respect to the endowed valuations $\overline{v}_i$ if there is no Full Swap or Partial Swap possible.

With the terminology in place, we formally describe our algorithm in Algorithm 1 and the local search subroutine in Algorithm 2.

Algorithm 1: Approximating Exact Capacitated One-Sided NSW for submodular valuations

Input: An instance $\langle N, M, \mathcal{V}, C \rangle$ of Exact Capacitated One-Sided NSW.
Output: A partial allocation $A = (A_i)_{i \in N}$.
/* Phase 1: Find a maximum weight one-to-one matching */
Compute a one-to-one matching $\tau : N \to M$ that maximizes $\prod_{i \in N} v_i(\tau(i))$.
/* H and J are the allocated and unallocated items respectively. */
Set $H \coloneqq \{\tau(i) : i \in N\}$ and $J \coloneqq M \setminus H$.
/* Phase 2: Local search using the unallocated items */
Compute the allocation $\mathcal{R} \coloneqq \texttt{LocalSearch}(\langle N, J, \mathcal{V}, C \rangle)$.
/* Phase 3: Rematching the previously allocated items in H */
Find a matching $\delta : N \to H$ maximizing $\prod_{i \in N} v_i(R_i + \delta(i))$.
return $A = (R_i + \delta(i))_{i \in N}$

Algorithm 2: LocalSearch

Input: An instance $\langle N, J, \mathcal{V}, C \rangle$.
Output: A partial allocation $\mathcal{R} \coloneqq (R_i)_{i \in N}$.
/* The set of agents which have positive utility for J. */
$\overline{N} \leftarrow \{i \in N : v_i(J) > 0\}$
/* $\ell(i)$ is the favourite item of i in J */
$\ell(i) \leftarrow \arg\max\{v_i(\ell) : \ell \in J\}$ for $i \in \overline{N}$
Define the endowed valuations by $\overline{v}_i(S) \coloneqq v_i(\ell(i)) + v_i(S)$ for all $i \in \overline{N}, S \subseteq M$.
Pick an arbitrary partial allocation $\mathcal{R}$ of items in J to agents in $\overline{N}$ such that $|R_i| = c_i - 1$ for every agent $i \in \overline{N}$. Keep all other items in J unallocated.
while *there exists a* Full Swap *or* Partial Swap *with respect to* $\overline{v}(\cdot)$ **do**
  perform the corresponding swap
Extend the partial allocation $\mathcal{R}$ to all the agents in N by arbitrarily allocating the unallocated items to agents in $N \setminus \overline{N}$ such that $|R_i| = c_i - 1$ for every agent $i \in N \setminus \overline{N}$.
return $\mathcal{R} \coloneqq (R_i)_{i \in N}$

Our next lemma shows that the local search subroutine (corresponding to Phase 2 in Algorithm 1) converges to a feasible $\overline{\varepsilon}$-local optimum in polynomial time.

Lemma 3.3 (Local search converges efficiently). *For any given instance $\langle N, J, \mathcal{V}, C\rangle$, Algorithm 2 terminates in $O\left(\frac{m}{\varepsilon}\log m\right)$ iterations and returns a partial allocation $(R_i)_{i\in N}$ that satisfies $|R_i| = c_i - 1$ for all $i \in N$. Moreover, the partial allocation $(R_i)_{i\in\overline{N}}$ computed just before Line 7 is an $\overline{\varepsilon}$-local optimum.*

Next, we will analyze Phase 3 of Algorithm 1. Given a matching $\rho : N \to H \cup \{\emptyset\}$, we will define $(\mathcal{R}, \rho)$ as a partial allocation in which every agent receives the set $R_i \cup \rho(i)$. The NSW of this allocation will be:

$$\mathrm{NSW}(\mathcal{R}, \rho) \coloneqq \prod_{i\in N} v_i(R_i + \rho(i))^{\frac{1}{n}}.$$

Recall that, in Phase 3, our algorithm computes a matching δ such that $\mathrm{NSW}(\mathcal{R}, \delta)$ is maximized. If we prove that there exists a matching $\sigma : N \to H \cup \{\emptyset\}$ such that $\mathrm{NSW}(\mathcal{R}, \sigma) \geq \frac{\mathrm{NSW}(\mathtt{OPT})}{6(1+\varepsilon)}$, then we will be done. To prove the existence of such a matching σ, we first define a mapping $\mathcal{T} : N \to 2^M$, which may not correspond to a valid partial allocation. However, this mapping is defined only for the purpose of the analysis, and since each agent is mapped under $\mathcal{T}$ to a bundle, we can nevertheless define NSW of $\mathcal{T}$. We will show that $\mathrm{NSW}(\mathcal{T}) \geq \frac{\mathrm{NSW}(\mathtt{OPT})}{6(1+\varepsilon)}$, and then prove that a matching σ satisfying $\mathrm{NSW}(\mathcal{R}, \sigma) \geq \mathrm{NSW}(\mathcal{T}) \geq \frac{\mathrm{NSW}(\mathtt{OPT})}{6(1+\varepsilon)}$ exists.

Consider an optimal partial allocation $\mathtt{OPT}$ of the given Capacitated One-Sided NSW instance. Further, let $S_i \coloneqq \mathtt{OPT}_i \cap J$ and $H_i \coloneqq \mathtt{OPT}_i \cap H$. For every agent $i \in N$, define $g(i)$ to be the item in H_i which provides the maximum marginal gain when added to S_i. Formally, $g(i) \coloneqq \arg\max_{j\in H_i} v_i(S_i + j)$. If H_i is empty, we define $g(i) \coloneqq \emptyset$.

Consider the following partitioning of the set of agents N:

- $N_g \coloneqq \{i \in N : v_i(g(i)) \geq \max\{v_i(R_i), v_i(\ell(i))\}\}$
- $N_R \coloneqq \left\{i \in N \setminus N_g : v_i(R_i) \geq \max\{v_i(g(i)), v_i(\ell(i))\}\right\}$
- $N_\ell \coloneqq N \setminus (N_g \cup N_R)$.

The intermediate mapping $\mathcal{T} = (T_i)_{i\in N}$ is defined as follows:

$$T_i \coloneqq \begin{cases} \{g(i)\}, & \text{if } i \in N_g, \\ R_i, & \text{if } i \in N_R, \\ \{\ell(i)\}, & \text{if } i \in N_\ell. \end{cases}$$

Note that the mapping $\mathcal{T}$ may not induce a feasible partial allocation because the item $\ell(i)$ may not be unique for each agent and may even be contained in some $R_{i'}$.

Lemma 3.4. *The mapping $\mathcal{T}$ satisfies $\mathrm{NSW}(\mathcal{T}) \geq \frac{\mathrm{NSW}(\mathtt{OPT})}{6(1+\varepsilon)}$.*

Using Lemma 3.4, we can show the existence of the required matching σ.

Lemma 3.5 (Existence of σ). *There exists a matching $\sigma : N \to H \cup \{\emptyset\}$ such that $\mathrm{NSW}(\mathcal{R}, \sigma) \geq \frac{\mathrm{NSW}(\mathtt{OPT})}{6(1+\varepsilon)}$.*

We can now present the proof of Lemma 3.2.

Proof. (of Lemma 3.2) We first prove that Algorithm 1 terminates in polynomial time and makes a polynomial number of value queries. The first phase can be computed in polynomial time by finding a max-weight matching of the bipartite graph whose vertex sets are the set of agents N and the set of items M. The edge (i, j) between agent i and item j has weight $\log(v_i(j))$. This construction clearly requires only a polynomial number of value queries. The second phase terminates in polynomial time with a polynomial number of value queries as shown in Lemma 3.3. Finally, the third phase also requires polynomial time and a polynomial number of value queries because we can create a bipartite graph as in the first phase where the weight of edge (i, j) for an agent $i \in N$ and an item $j \in H$ is $\log(v_i(R_i + j))$.

The algorithm returns a feasible partial allocation because $|R_i|$ is guaranteed to be $c_i - 1$, and we allocate one more item to each agent in the final phase. Since our algorithm finds a matching that maximizes $\mathrm{NSW}(\mathcal{R}, \delta)$, we get $\mathrm{NSW}(\mathtt{ALG}) \geq \frac{\mathrm{NSW}(\mathtt{OPT})}{6(1+\varepsilon)}$ using Lemma 3.5. To get the factor of $(6 + \varepsilon)$, we can scale down ε by a factor of 6. □

4 APPROXIMATION ALGORITHMS FOR THE TWO-SIDED MODEL

We will now present our algorithmic results for two-sided Nash welfare. Our main result is a 1.33-approximation algorithm for Capacitated Two-Sided NSW under subadditive valuations (Theorem 4.1). This result significantly improves upon an existing $\sqrt{\mathtt{OPT}}$-approximation algorithm for positive additive valuations [26].

A corollary of our main result is that when the number of firms is constant, we obtain a polynomial-time (and polynomial-query) approximation scheme (PTAS) for the two-sided problem under subadditive valuations (Corollary 4.3). This result improves upon a quasipolynomial-time approximation scheme (QPTAS) for a constant number of firms under polynomially bounded additive valuations [26]. We will start by proving our main result in Theorem 4.1.

Theorem 4.1. *There exists a 1.33-approximation algorithm for* Capacitated Two-Sided NSW *under subadditive valuations. The algorithm runs in* $\mathrm{poly}(n, m)$ *time and makes* $\mathrm{poly}(n, m)$ *number of value queries.*

Towards proving Theorem 4.1, we will first state a lemma that provides an upper bound on the Nash welfare of any many-to-one matching. The bound is in terms of the firms' values for only their *favorite* matched worker.

Lemma 4.2. *Let $\mathcal{I} = \langle F, W, \mathcal{V}, \mathcal{W}, C\rangle$ be an instance of* Capacitated Two-Sided NSW *with subadditive valuations and let μ be any feasible many-to-one matching for $\mathcal{I}$. Let $j_i \in \arg\max_{j\in\mu_i} v_i(j)$ be firm i's favourite worker in μ_i. Then,*

$$\mathrm{NSW}(\mu) \leq 1.33\left(\prod_{i\in F} v_i(j_i)\prod_{j\in W} w_j(\mu_j)\right)^{\frac{1}{m+n}}.$$

Proof. (of Lemma 4.2) We know that among the workers in μ_i, firm i assigns the highest value to the worker j_i. By the subadditivity of v_i, we get that $v_i(\mu_i) \leq |\mu_i| v_i(j_i)$.

Substituting this inequality in the expression of $\mathrm{NSW}(\mu)$ gives:

$$\mathrm{NSW}(\mu) \leq \left(\prod_{i\in F}|\mu_i|\prod_{i\in F} v_i(j_i)\prod_{j\in W} w_j(\mu_j)\right)^{\frac{1}{m+n}}. \quad (1)$$

Since $\sum_{i\in F}|\mu_i| \leq m$, using AM-GM inequality, we have $\prod_{i\in F}|\mu_i| \leq \left(\frac{m}{n}\right)^n$. Hence, $\left(\prod_{i\in F}|\mu_i|\right)^{1/m+n} \leq \left(\frac{m}{n}\right)^{n/m+n} = x^{1/1+x}$, where $x = m/n$. The function $x^{1/1+x}$ is upper bounded by 1.3211 for all $x > 0$. Substituting this bound in Equation (1) completes the proof. □

With this bound on NSW in place, we will now create a Min-Cost-Flow instance. Recall that in the Min-Cost-Flow problem,

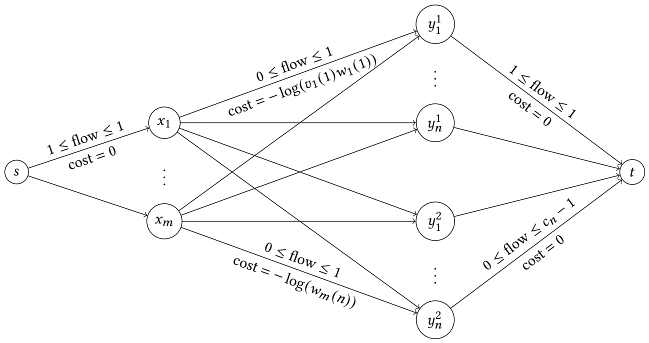

Figure 1: MIN-COST-FLOW network used in the proof of Theorem 4.1. The edge labels show the cost per unit flow and the lower and upper bounds on the flow.

we are given a flow network where each edge has a lower as well as an upper bound on the flow that can pass through it, and a cost per unit flow associated with it. The objective is to find a feasible flow with the minimum total cost. The problem is known to be solvable in polynomial time [37].

PROOF. (of Theorem 4.1) Let $\mathcal{I} = \langle F, W, \mathcal{V}, \mathcal{W}, C \rangle$ be the given instance of CAPACITATED TWO-SIDED NSW. For an edge e in a flow network, we will use $L(e)$ and $U(e)$ to denote the lower and the upper bounds on the flow through e, respectively, and will write $\text{cost}(e)$ to denote the cost per unit flow through e. Also, we will use $f(e)$ to denote the flow through the edge e. The flow network is constructed as follows (see Figure 1):

- Create m vertices $x_1, \dots, x_m$ corresponding to the m workers.
- Create $2n$ vertices $y_1^1, \dots, y_n^1$ and $y_1^2, \dots, y_n^2$ corresponding to the n firms. We call y_i^1 the *main copy* and y_i^2 the *secondary copy* of firm i.
- Create a source vertex s. For each worker j, add an edge from s to x_j with $L(s \to x_j) = 1, U(s \to x_j) = 1$, and $\text{cost}(s \to x_j) = 0$.
- Create a sink vertex t. For each firm i, add an edge from y_i^1 to t with $L(y_i^1 \to t) = 1$, $U(y_i^1 \to t) = 1$ and $\text{cost}(y_i^1 \to t) = 0$. Also, add an edge from y_i^2 to t with $L(y_i^2 \to t) = 0, U(y_i^2 \to t) = c_i - 1$ and $\text{cost}(y_i^2 \to t) = 0$.
- For all $j \in W$ and $i \in F$, such that $v_i(j) \neq 0$ and $w_j(i) \neq 0$, add an edge from x_j to y_i^1 with $L(x_j \to y_i^1) = 0, U(x_j \to y_i^1) = 1$ and $\text{cost}(x_j \to y_i^1) = -\log(v_i(j) w_j(i))$.
- For all $j \in W$ and $i \in F$ such that $w_j(i) \neq 0$, add an edge from x_j to y_i^2 with $L(x_j \to y_i^2) = 0, U(x_j \to y_i^2) = 1$ and $\text{cost}(x_j \to y_i^2) = -\log(w_j(i))$.

Let $\mu^\star$ be a Nash optimal many-to-one matching of $\mathcal{I}$. We will assume that, under $\mu^\star$, every firm is assigned at least one worker for whom it has a nonzero value. Likewise, each worker is matched to a firm that it values positively. Indeed, if this were not the case, then the optimal Nash welfare would be zero, making every possible feasible assignment compliant with the approximation value. For a firm i, let $j_i^\star$ denote its favorite assigned worker under $\mu^\star$. Notice that the following flow vector f is feasible in the above network:

- $f(s \to x_j) = 1$ for all the workers j.
- $f(y_i^1 \to t) = 1$ for all firms i.
- $f(x_{j_i^\star} \to y_i^1) = 1$ for all firms i.
- $f(x_j \to y_i^2) = 1$ for all $j \in \mu_i^\star \setminus \{j_i^\star\}$
- $f(y_i^2 \to t) = |\mu_i^\star| - 1$ for all firms i.

For all other edges e in the network, we set $f(e) = 0$. Clearly, the cost of f is:

$$\text{cost}(f) = -\sum_{i \in F} \log(v_i(j_i^\star)) - \sum_{j \in W} \log(w_j(\mu_j^\star)). \tag{2}$$

Our algorithm finds the minimum-cost flow, say $f^\star$, in the network. Note that any feasible flow from s has the same value as the maximum flow. Furthermore, since the upper and lower bounds on the flow are integral, we have that the resulting flow is integral. Thus, the flow $f^\star$ induces a valid many-to-one matching, say $\bar{\mu}$, of workers to firms. Under $\bar{\mu}$, each worker j is assigned to a unique firm i (i.e., $\bar{\mu}_j \coloneqq i$) such that $f^\star(x_j \to y_i^1) = 1$ or $f^\star(x_j \to y_i^2) = 1$. For each firm i, let $\bar{\mu}_i$ be the the set of workers assigned to the firm i, and let $\bar{j}_i$ be the unique worker j for which $f^\star(x_j \to y_i^1) = 1$. Therefore, the cost of $f^\star$ is:

$$\text{cost}(f^\star) = -\sum_{i \in F} \log(v_i(\bar{j}_i)) - \sum_{j \in W} \log(w_j(\bar{\mu}_j)). \tag{3}$$

Using the cost optimality of $f^\star$ along with Equations (2) and (3),

$$\textstyle\prod_{i \in F} v_i(\bar{j}_i) \prod_{j \in W} w_j(\bar{\mu}_j) \geq \prod_{i \in F} v_i(j_i^\star) \prod_{j \in W} w_j(\mu_j^\star). \tag{4}$$

Therefore, the Nash welfare of our assignment is at least:

$$\begin{aligned} \text{NSW}(\bar{\mu}) &= \left(\textstyle\prod_{i \in F} v_i(\bar{\mu}_i) \prod_{j \in W} w_j(\bar{\mu}_j)\right)^{\frac{1}{m+n}} \\ &\geq \left(\textstyle\prod_{i \in F} v_i(\bar{j}_i) \prod_{j \in W} w_j(\bar{\mu}_j)\right)^{\frac{1}{m+n}} \\ &\geq \left(\textstyle\prod_{i \in F} v_i(j_i^\star) \prod_{j \in W} w_j(\mu_j^\star)\right)^{\frac{1}{m+n}} \\ &\geq \frac{\text{NSW}(\mu^\star)}{1.33}, \end{aligned}$$

where the second-last inequality follow from Equation (4) and the last inequality follows from Lemma 4.2. This finishes the proof of Theorem 4.1. □

We will now show that when the number of firms is constant, the algorithm in the proof of Theorem 4.1 can be used to obtain a polynomial-time approximation scheme (PTAS) for the two-sided problem under subadditive valuations.

Corollary 4.3 (PTAS for constant number of firms). *For any $\varepsilon > 0$ and a constant number of firms, there exists a $(1 + \varepsilon)$-approximation algorithm for* CAPACITATED TWO-SIDED NSW *under subadditive valuations. The algorithm makes* $\text{poly}(m)$ *number of value queries and runs in* $\text{poly}(m)$ *time.*

PROOF. (of Corollary 4.3) If $\varepsilon \geq 0.33$, the algorithm in Theorem 4.1 is already a $(1 + \varepsilon)-$ approximate algorithm. Hence, we assume $\varepsilon < 0.33$. Our algorithm runs the MIN-COST-FLOW algorithm of Theorem 4.1 if $m \geq \frac{n}{\varepsilon^2}$, and otherwise solves the problem exactly by iterating over all the $O(n^m)$ many-to-one matchings. The running time of our algorithm is $O\left(n^{n/\varepsilon^2} \cdot \text{poly}(n, m)\right)$.

The approximation ratio of our algorithm, as derived from the proof of Lemma 4.2, equals $r = x^{1/1+x}$, where $x = \frac{m}{n} > \frac{1}{\varepsilon^2} > \frac{1}{0.33^2} > 9$. It can be verified that $\frac{\log(x)}{1+x} < \frac{1}{x^{0.75}}$ for all $x > 9$, and that $\varepsilon^{1.5} < \log(1+\varepsilon)$ for all $\varepsilon < 0.33$. Therefore, $\log(r) = \frac{\log(x)}{1+x} < \frac{1}{x^{0.75}} < \varepsilon^{1.5} < \log(1+\varepsilon)$, as desired for the $(1+\varepsilon)$-approximation. □

5 HARDNESS RESULTS

In this section, we will show that the problem of maximizing two-sided Nash social welfare is APX-hard; specifically, the problem is NP-hard to approximate within a factor of 1.0000759. This result holds even in the absence of capacity constraints and even under additive valuations. Prior to our result, only NP-hardness was known for this problem [26].

Theorem 5.1 (Hardness for two-sided Nash welfare). *Unless* P = NP*, no polynomial-time algorithm can approximate* Uncapacitated Two-Sided NSW *to within a factor of* 1.0000759 *even under additive valuations.*

To prove Theorem 5.1, we will use Lemma 5.2, as stated below. This lemma is based on a result of Garg and Murhekar [23], who showed APX-hardness of Uncapacitated One-Sided NSW under additive valuations. The key property from their reduction, needed in Lemma 5.2, is that the number of items in the reduced instance is at most a constant (specifically, 1.5) times the number of agents.

Lemma 5.2 (modified from [23]). *Unless* P = NP*, no polynomial-time algorithm can approximate* Uncapacitated One-Sided NSW *with additive valuations to a factor smaller than* 1.00019*, even when the number of items is at most* 1.5 *times the number of agents (i.e.,* $m \leq 1.5n$*).*

We will now prove the APX-hardness of two-sided Nash welfare even in the absence of capacity constraints.

Proof. (of Theorem 5.1) Consider an instance of Uncapacitated One-Sided NSW instance denoted by $\mathcal{I}_1 = \langle N, M, \mathcal{V}_1, C\rangle$, where the number of items $m = |M|$ is at most 1.5 times the number of agents $n = |N|$, and the valuations are additive.

We will create an instance of Uncapacitated Two-Sided NSW, denoted by $\mathcal{I}_2 = \langle F, W, \mathcal{V}_2, \mathcal{W}, C\rangle$, as follows: The set of firms (respectively, workers) corresponds to the set of agents (respectively, items) in the one-sided instance $\mathcal{I}_1$, i.e., $F = N$ and $W = M$. A firm's valuations for the workers as well as its capacity are identical to the corresponding agent's valuations for the items and capacity, respectively; thus, $\mathcal{V}_2 = \mathcal{V}_1$. Finally, the workers have uniform valuations over the firms, i.e., for each worker $j \in W$ and each firm $i \in F$, $w_j(i) = 1$.

Note that, any allocation $A^{\mathcal{I}_1}$ of the items to the agents in the one-sided instance $\mathcal{I}_1$ corresponds naturally to a many-to-one matching $\mu^{\mathcal{I}_2}$ in the instance $\mathcal{I}_2$ and vice-versa. Here, the set of workers $\mu_i^{\mathcal{I}_2}$ assigned to firm i under $\mu^{\mathcal{I}_2}$ equals $A_i^{\mathcal{I}_1}$, the set of items allocated to agent i in the allocation $A^{\mathcal{I}_1}$. Now, since all the worker valuations are uniformly equal to 1, we have:

$$\text{NSW}(A^{\mathcal{I}_1}) = \left(\text{NSW}(\mu^{\mathcal{I}_2})\right)^{\frac{m+n}{n}} = \left(\text{NSW}(\mu^{\mathcal{I}_2})\right)^{1+\frac{m}{n}}.$$

Hence, any γ-approximate solution for $\mathcal{I}_2$ yields a $\left(\gamma^{1+m/n}\right)$- approximate solution for $\mathcal{I}_1$. Therefore, using Lemma 5.2, we have that unless P = NP, no polynomial-time algorithm can approximates Uncapacitated Two-Sided NSW to a factor smaller than $(1.00019)^{1/1+1.5} > 1.0000759$. □

Remark. It follows from the proof of Theorem 5.1 that any reduction showing that Uncapacitated One-Sided NSW is hard to approximate within a factor of α under additive valuations, for instances where $m \leq cn$ for some constant c, yields a $\alpha^{1/1+c}$ hardness-of-approximation ratio for Uncapacitated Two-Sided NSW.

6 CONCLUDING REMARKS

We studied algorithmic aspects of maximizing Nash social welfare for one-sided and two-sided preferences under capacity constraints. We developed constant-factor approximation algorithms for both settings, complementing the APX-hardness results. Our algorithm for the one-sided problem provides the first constant-factor approximation algorithm for Nash welfare under submodular valuations and capacity constraints, while our result for the two-sided problem applies to subadditive valuations and significantly improves upon the existing $\sqrt{\texttt{OPT}}$-approximation for additive valuations.

Our work opens up several directions for future work. Firstly, it would be interesting to obtain tight bounds for the approximations mentioned in Table 1. In particular, our hardness reduction uniformly assigns a valuation of 1 to all workers, thereby essentially leveraging the hardness of the one-sided problem. An important question to address is whether stronger hardness-of-approximation results are attainable when workers' valuations are not uniform. Secondly, exploring extensions to more general constraints than capacities, such as matroid constraints, would be a fruitful direction for further investigation. Lastly, it would also be of interest to study *weighted* Nash welfare in the two-sided setting, wherein each agent i (worker or firm) is associated with a nonnegative weight α_i, and the objective function is defined as:

$$\text{NSW}(\mu) = \prod_{i\in F} v_i(\mu_i)^{\alpha_i} \cdot \prod_{j\in W} w_j(\mu_j)^{\alpha_j}.$$

The weights are normalized so that $\sum_{i\in F}\alpha_i + \sum_{j\in W}\alpha_j = 1$. This family of welfare measures includes several natural proposals for measuring welfare in the two-sided setting. When all weights are equal to $1/m+n$, we recover the objective studied in our work. Additionally, when $\alpha_i = 1/2n$ for all $i \in F$ and $\alpha_j = 1/2m$ for all $j \in W$, we obtain a "separable" objective that multiplies the geometric mean of the workers with the geometric mean of the firms. There exist approximation algorithms for weighted Nash welfare in the one-sided problem [17, 21], and it would be useful to develop similar results for the two-sided problem.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their valuable feedback. We are grateful to Vignesh Viswanathan for helpful discussions and useful feedback on an earlier version of this paper. RV acknowledges support from DST INSPIRE grant no. DST/INSPIRE/04/2020/000107, SERB grant no. CRG/2022/002621, and iHub Anubhuti IIITD Foundation. JY acknowledges support from Google PhD fellowship.

REFERENCES

[1] Hannaneh Akrami, Bhaskar Ray Chaudhury, Martin Hoefer, Kurt Mehlhorn, Marco Schmalhofer, Golnoosh Shahkarami, Giovanna Varricchio, Quentin Vermande, and Ernest van Wijland. 2022. Maximizing Nash Social Welfare in 2-Value Instances. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence*, Vol. 36. 4760–4767.

[2] Georgios Amanatidis, Haris Aziz, Georgios Birmpas, Aris Filos-Ratsikas, Bo Li, Hervé Moulin, Alexandros A Voudouris, and Xiaowei Wu. 2023. Fair Division of

Indivisible Goods: Recent Progress and Open Questions. *Artificial Intelligence* (2023), 103965.
[3] Nima Anari, Tung Mai, Shayan Oveis Gharan, and Vijay V Vazirani. 2018. Nash Social Welfare for Indivisible Items under Separable, Piecewise-Linear Concave Utilities. In *Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms*. 2274–2290.
[4] Moshe Babaioff, Tomer Ezra, and Uriel Feige. 2021. Fair and Truthful Mechanisms for Dichotomous Valuations. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, Vol. 35. 5119–5126.
[5] Siddharth Barman, Umang Bhaskar, Anand Krishna, and Ranjani G Sundaram. 2020. Tight Approximation Algorithms for p-Mean Welfare Under Subadditive Valuations. In *Proceedings of the 28th Annual European Symposium on Algorithms*.
[6] Siddharth Barman, Sanath Kumar Krishnamurthy, and Rohit Vaish. 2018. Finding Fair and Efficient Allocations. In *Proceedings of the 2018 ACM Conference on Economics and Computation*. 557–574.
[7] Siddharth Barman, Sanath Kumar Krishnamurthy, and Rohit Vaish. 2018. Greedy Algorithms for Maximizing Nash Social Welfare. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. 7–13.
[8] Nawal Benabbou, Mithun Chakraborty, Ayumi Igarashi, and Yair Zick. 2021. Finding Fair and Efficient Allocations for Matroid Rank Valuations. *ACM Transactions on Economics and Computation* 9, 4 (2021), 1–41.
[9] Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D Procaccia. 2016. *Handbook of Computational Social Choice*. Cambridge University Press.
[10] Ioannis Caragiannis, David Kurokawa, Hervé Moulin, Ariel D Procaccia, Nisarg Shah, and Junxing Wang. 2019. The Unreasonable Fairness of Maximum Nash Welfare. *ACM Transactions on Economics and Computation* 7, 3 (2019), 1–32.
[11] Bhaskar Ray Chaudhury, Yun Kuen Cheung, Jugal Garg, Naveen Garg, Martin Hoefer, and Kurt Mehlhorn. 2022. Fair Division of Indivisible Goods For a Class of Concave Valuations. *Journal of Artificial Intelligence Research* 74 (2022), 111–142.
[12] Bhaskar Ray Chaudhury, Jugal Garg, and Ruta Mehta. 2021. Fair and Efficient Allocations under Subadditive Valuations. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, Vol. 35. 5269–5276.
[13] Richard Cole, Nikhil Devanur, Vasilis Gkatzelis, Kamal Jain, Tung Mai, Vijay V Vazirani, and Sadra Yazdanbod. 2017. Convex Program Duality, Fisher Markets, and Nash Social Welfare. In *Proceedings of the 2017 ACM Conference on Economics and Computation*. 459–460.
[14] Richard Cole and Vasilis Gkatzelis. 2018. Approximating the Nash Social Welfare with Indivisible Items. *SIAM J. Comput.* 47, 3 (2018), 1211–1236.
[15] Shahar Dobzinski, Wenzheng Li, Aviad Rubinstein, and Jan Vondrák. 2024. A Constant-Factor Approximation for Nash Social Welfare with Subadditive Valuations. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*. 467–478.
[16] Edmund Eisenberg and David Gale. 1959. Consensus of Subjective Probabilities: The Pari-Mutuel Method. *The Annals of Mathematical Statistics* 30, 1 (1959), 165–168.
[17] Yuda Feng and Shi Li. 2024. A Note on Approximating Weighted Nash Social Welfare with Additive Valuations. In *Proceedings of the 51st International Colloquium on Automata, Languages, and Programming*, Vol. 297. 63:1–63:9.
[18] Zack Fitzsimmons, Vignesh Viswanathan, and Yair Zick. 2024. On the Hardness of Fair Allocation under Ternary Valuations. *arXiv preprint arXiv:2403.00943* (2024).
[19] Jugal Garg, Martin Hoefer, and Kurt Mehlhorn. 2018. Approximating the Nash Social Welfare with Budget-Additive Valuations. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*. 2326–2340.
[20] Jugal Garg, Martin Hoefer, and Kurt Mehlhorn. 2024. Satiation in Fisher Markets and Approximation of Nash Social Welfare. *Mathematics of Operations Research* 49, 2 (2024), 1109–1139.
[21] Jugal Garg, Edin Husić, Wenzheng Li, László A Végh, and Jan Vondrák. 2023. Approximating Nash Social Welfare by Matching and Local Search. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*. 1298–1310.
[22] Jugal Garg, Pooja Kulkarni, and Rucha Kulkarni. 2023. Approximating Nash Social Welfare under Submodular Valuations through (un) Matchings. *ACM Transactions on Algorithms* 19, 4 (2023), 1–25.
[23] Jugal Garg and Aniket Murhekar. 2021. Computing Fair and Efficient Allocations with Few Utility Values. In *Proceedings of the 14th International Symposium on Algorithmic Game Theory*. 345–359.
[24] Salil Gokhale, Harshul Sagar, Rohit Vaish, and Jatin Yadav. 2024. Approximating One-Sided and Two-Sided Nash Social Welfare With Capacities. *arXiv preprint arXiv:2411.14007* (2024).
[25] Dan Gusfield and Robert W Irving. 1989. *The Stable Marriage Problem: Structure and Algorithms*. MIT press.
[26] Pallavi Jain and Rohit Vaish. 2024. Maximizing Nash Social Welfare under Two-Sided Preferences. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence*, Vol. 38. 9798–9806.
[27] Mamoru Kaneko and Kenjiro Nakamura. 1979. The Nash Social Welfare Function. *Econometrica: Journal of the Econometric Society* (1979), 423–435.
[28] Subhash Khot, Richard J Lipton, Evangelos Markakis, and Aranyak Mehta. 2008. Inapproximability Results for Combinatorial Auctions with Submodular Utility Functions. *Algorithmica* 52 (2008), 3–18.
[29] Donald Ervin Knuth. 1997. *Stable Marriage and its Relation to Other Combinatorial Problems: An Introduction to the Mathematical Analysis of Algorithms*. Vol. 10. American Mathematical Soc.
[30] Euiwoong Lee. 2017. APX-Hardness of Maximizing Nash Social Welfare with Indivisible Items. *Inform. Process. Lett.* 122 (2017), 17–20.
[31] David Manlove. 2013. *Algorithmics of Matching under Preferences*. Vol. 2. World Scientific.
[32] Evangelos Markakis. 2017. Approximation Algorithms and Hardness Results for Fair Division with Indivisible Goods. *Trends in Computational Social Choice* (2017), 231–247.
[33] Hervé Moulin. 2004. *Fair Division and Collective Welfare*. MIT press.
[34] John F Nash Jr. 1950. The Bargaining Problem. *Econometrica: Journal of the Econometric Society* (1950), 155–162.
[35] Nhan-Tam Nguyen, Trung Thanh Nguyen, Magnus Roos, and Jörg Rothe. 2014. Computational Complexity and Approximability of Social Welfare Optimization in Multiagent Resource Allocation. *Autonomous Agents and Multi-Agent Systems* 28, 2 (2014), 256–289.
[36] Noam Nisan, Tim Roughgarden, Éva Tardos, and Vijay V. Vazirani. 2007. *Algorithmic Game Theory*. Cambridge University Press, New York, NY, USA.
[37] James B Orlin. 1997. A Polynomial Time Primal Network Simplex Algorithm for Minimum Cost Flows. *Mathematical Programming* 78 (1997), 109–129.
[38] Alvin E Roth and Marilda Sotomayor. 1992. Two-Sided Matching. *Handbook of Game Theory with Economic Applications* 1 (1992), 485–541.
[39] Warut Suksompong. 2021. Constraints in Fair Division. *ACM SIGecom Exchanges* 19, 2 (2021), 46–61.
[40] Vignesh Viswanathan and Yair Zick. 2023. A General Framework for Fair Allocation under Matroid Rank Valuations. In *Proceedings of the 24th ACM Conference on Economics and Computation*. 1129–1152.
[41] Jan Vondrák. 2008. Optimal Approximation for the Submodular Welfare Problem in the Value Oracle Model. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*. 67–74.
[42] Xiaowei Wu, Bo Li, and Jiarui Gan. 2021. Budget-Feasible Maximum Nash Social Welfare is Almost Envy-free. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence*. 465–471.