

Maximizing Nash Social Welfare under Two-Sided Preferences

Pallavi Jain¹, Rohit Vaish²

¹Indian Institute of Technology Jodhpur, India

²Indian Institute of Technology Delhi, India
 pallavi@iitj.ac.in, rvaish@iitd.ac.in

Abstract

The maximum Nash social welfare (NSW)—which maximizes the geometric mean of agents’ utilities—is a fundamental solution concept with remarkable fairness and efficiency guarantees. The computational aspects of NSW have been extensively studied for *one-sided* preferences where a set of agents have preferences over a set of resources. Our work deviates from this trend and studies NSW maximization for *two-sided* preferences, wherein a set of workers and firms, each having a cardinal valuation function, are matched with each other. We provide a systematic study of the computational complexity of maximizing NSW for many-to-one matchings under two-sided preferences. Our main negative result is that maximizing NSW is NP-hard even in a highly restricted setting where each firm has capacity 2, all valuations are in the range $\{0, 1, 2\}$, and each agent positively values at most three other agents. In search of positive results, we develop approximation algorithms as well as parameterized algorithms in terms of natural parameters such as the number of workers, the number of firms, and the firms’ capacities. We also provide algorithms for restricted domains such as symmetric binary valuations and bounded degree instances.

Introduction

The problem of matching two sets of agents, with each side having preferences over the other, has been extensively studied in economics, computer science, and artificial intelligence (Roth and Sotomayor 1992; Manlove 2013; Brandt et al. 2016). Such *two-sided* matching problems have found several notable real-world applications such as in labor markets (Roth and Peranson 1999), school choice (Abdulkadiroğlu and Sönmez 2003), and online dating platforms (Hitsch, Hortaçsu, and Ariely 2010).

Despite their extensive practical applicability, commonly-used algorithms for these problems often give rise to concerns about *unfairness*. A well-known example is the deferred-acceptance algorithm, which is known to strongly favor one side at the expense of the other (Gale and Shapley 1962). Similarly, the matching algorithms used by ridesharing platforms have been found to contribute to the wage gap between men and women (Cook et al. 2021).

Copyright © 2024, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

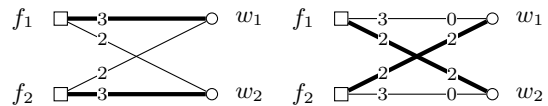


Figure 1: One-sided (left) and two-sided (right) instances. For each edge, the number close to a vertex denotes how much that agent values the other agent. The Nash welfare maximizing matchings are highlighted via thick edges.

The area of *fair division* provides a formal mathematical framework for rigorously analyzing fairness concepts (Brams and Taylor 1996; Moulin 2004). The literature on fair division has focused on *one-sided* preferences, wherein a set of agents have preferences over the resources. A prominent measure of fairness in this context is *Nash social welfare*, defined as the geometric mean of agents’ utilities (Nash Jr 1950; Kaneko and Nakamura 1979).

Nash welfare provides a “sweet spot” between the purely welfarist utilitarian objective and the purely egalitarian Rawlsian objective. It strikes a balance between the often-conflicting goals of fairness and economic efficiency, and enjoys a strong axiomatic support (Moulin 2004; Caragiannis et al. 2019). In recent years, the computational aspects of Nash welfare have gained considerable attention (Cole and Gkatzelis 2018; Barman, Krishnamurthy, and Vaish 2018a; Amanatidis et al. 2021; Akrami et al. 2022).

Somewhat surprisingly, Nash welfare has not been studied for *two-sided* preferences. Our work aims to address this gap by proposing a systematic examination of the computational complexity of maximizing Nash welfare in the two-sided matching problem.

Summary of Contributions

Our model consists of two disjoint sets of agents: *workers* and *firms*. Each worker has a nonnegative valuation for every firm and can be matched with at most one firm. Each firm can be matched with multiple workers (up to its capacity) and has *additive* valuations over the workers. The goal is to find a *Nash optimal* many-to-one matching, i.e., maximize the geometric mean of workers’ and firms’ utilities.

Before discussing our results, let us illustrate a key difference between one-sided and two-sided preferences through

# firms	Capacities	Valuations	Exact/Approx.	Complexity	Technique	Reference
—	2	Ternary $\{0, 1, 2\}$	Exact	NP-hard	Rainbow Matching	Theorem 1
2	Equal	Identical, symmetric	Exact	NP-hard	Partition	Proposition 3
—	—	Positive	$1/\sqrt{\text{opt}}$	Poly time	Submodularity	Theorem 2
—	Constant	Positive, constant	Constant approx.	Poly time	Submodularity	Corollary 2
Constant	—	$\mathcal{O}(\text{poly}(m, n))$	$1/(1+\varepsilon)$ -approx.	Quasipoly time	Bucketing	Theorem 3
Constant	—	Constant	Exact	Poly time	Bucketing	Corollary 3
—	—	—	Exact	FPT, $\mathcal{O}^*(3^m)$	Dynamic prog.	Theorem 4
—	Constant	—	Exact	FPT, $\mathcal{O}^*(2^m)$	Dynamic prog.	Theorem 5
—	—	—	$\frac{-(n+1)}{(1+\varepsilon)^{(m+n)}}\text{-approx.}$	FPT, $\mathcal{O}^*(1/\varepsilon^4 \cdot 2^m)$	Poly. multiplication	Theorem 6
—	—	Symmetric binary	Exact	Poly time	Greedy algorithm	Theorem 7
—	—	—	Exact	Poly time for $\text{deg} \leq 2$	Paths and cycles	Theorem 8
—	Exactly 2	—	Exact	Poly time for $\text{deg} \leq 3$	Reduction rules	Theorem 8

Table 1: Summary of our results on maximizing Nash welfare with n firms and m workers. Each row lists a result under the assumptions specified by the individual columns. A “—” means that the parameter for that column can be arbitrary. The red colored rows contain hardness results, blue rows contain approximation algorithms, and green rows contain exact algorithms.

the example in Figure 1. There are two workers w_1, w_2 and two firms f_1, f_2 , each with capacity 1. The assignment $\{(f_1, w_1), (f_2, w_2)\}$ maximizes Nash welfare in the one-sided instance (ref. left subfigure). But when the workers’ preferences are taken into account (ref. right subfigure), this assignment turns out to be *arbitrarily* suboptimal, giving zero values to all workers and having zero Nash welfare. Thus, the structure of Nash optimal solutions for two-sided instances can drastically differ from their one-sided counterparts, which makes the two-sided problem challenging.

Table 1 summarizes our results. Some of the technical highlights of our work are discussed below.

Hardness Results. We show that finding a Nash optimal matching is NP-hard even when each firm has capacity 2 and the valuations are *ternary* in the range $\{0, 1, 2\}$ (Theorem 1). This hardness result is “tight” in the sense that further restricting either of these assumptions—firms having unit capacities or valuations being *symmetric* and *binary* $\{0, 1\}$ —allows efficient algorithms (Proposition 2 and Theorem 7).

Approximation Algorithms. We provide two approximation algorithms. The first one is a $\frac{1}{\sqrt{\text{opt}}}$ -approximation algorithm, where opt is the optimal Nash welfare (Theorem 2). This result assumes that the valuations are positive integers, and uses a reduction to the fair division problem under *submodular* valuations and *matroid* constraints. The second algorithm uses a discretization technique to give a quasipolynomial-time approximation scheme (QPTAS) for a constant number of firms and polynomially-bounded valuations (Theorem 3). Note that the problem is NP-hard even for *two* firms (Proposition 3). Whether this hardness implication can be achieved for polynomially-bounded valuations remains unresolved.

Parameterized Algorithms. We provide two parameterized algorithms. The first algorithm, based on dynamic programming, is fixed-parameter tractable (FPT) in the number of workers m and has a running time of $\mathcal{O}^*(3^m)$ (Theorem 4); here, the notation $\mathcal{O}^*(\cdot)$ suppresses multiplicative polynomial terms. The second algorithm is an FPT approxi-

mation scheme with a faster running time of $\mathcal{O}^*(2^m)$ (Theorem 6). This algorithm uses polynomial multiplication; to our knowledge, this is the first use of this technique in the context of Nash welfare.

Restricted Domains. Finally, we develop exact, polynomial-time algorithms for restricted settings such as symmetric binary valuations (Theorem 7) and bounded degree instances (Theorem 8). Our algorithm for symmetric binary valuations uses the greedy algorithm of Barman, Krishnamurthy, and Vaish (2018b) from fair division for maximizing Nash welfare under binary valuations.

Related Work

We will now briefly survey the relevant literature on Nash welfare in fair division. A detailed discussion of the related work can be found in the full version (Jain and Vaish 2023).

For fair division of *divisible* resources, the well-known Eisenberg-Gale convex program is known to efficiently compute a fractional allocation that maximizes Nash welfare (Eisenberg and Gale 1959). By contrast, for *indivisible* resources, the problem becomes APX-hard under additive valuations (Nguyen et al. 2014; Lee 2017). Note that the fair division problem for indivisible resources is a special case of our model when every worker (i.e., the ‘items’) values all firms (i.e., the ‘agents’) at 1, and the firms have *unrestricted* capacities. For the case of restricted capacities, which is the focus of our study, the hardness constructions from fair division do not automatically carry over to our setting (we discuss some exceptions to this remark in the section on hardness results).

On the algorithmic front, for additive valuations, a fully polynomial-time approximation scheme (FPTAS) is known for any fixed number of agents (Nguyen et al. 2014; Garg et al. 2022), and constant-factor approximation algorithms are known for a general number of agents (Cole and Gkatzelis 2018; Cole et al. 2017; Barman, Krishnamurthy, and Vaish 2018a). Several works have studied approximation algorithms for more general classes of valuations such as *budget-additive* (Garg, Hoefer, and Mehlhorn 2018), *sep-*

arable piecewise-linear concave (Anari et al. 2018), sub-modular (Garg, Kulkarni, and Kulkarni 2020; Garg et al. 2023), XOS (Barman et al. 2021), and subadditive (Barman et al. 2020; Chaudhury, Garg, and Mehta 2021). For more details on fair division with indivisible resources, we refer the reader to the survey by Amanatidis et al. (2023).

Preliminaries

For any positive integer r , let $[r] := \{1, 2, \dots, r\}$.

Problem instance. An instance of the *two-sided matching problem* is given by a tuple $\langle W, F, C, \mathcal{V} \rangle$, where $W = \{w_1, \dots, w_m\}$ is the set of m workers, $F = \{f_1, \dots, f_n\}$ is the set of n firms, $C = \{c_1, \dots, c_n\}$ is the set of capacities of the firms, and $\mathcal{V} = (v_{w_1}, \dots, v_{w_m}, v_{f_1}, \dots, v_{f_n})$ is the valuation profile consisting of the valuation function of each worker and firm. For each worker $w \in W$, its valuation function $v_w : F \cup \{\emptyset\} \rightarrow \mathbb{N} \cup \{0\}$ specifies its numerical value for each firm, where $v_w(\emptyset) := 0$. For each firm $f \in F$, its valuation function $v_f : 2^W \rightarrow \mathbb{N} \cup \{0\}$ specifies its numerical value for each subset of workers, where $v_f(\emptyset) := 0$.

Many-to-one matching. A *many-to-one matching* $\mu : W \times F \rightarrow \{0, 1\}$ is a function that assigns each worker-firm pair a weight of either 0 or 1 such that:

- for every worker $w \in W$, $\sum_{f \in F} \mu(w, f) \leq 1$, i.e., each worker is matched with at most one firm, and
- for every firm $f \in F$, $\sum_{w \in W} \mu(w, f) \leq c_f$, i.e., each firm f is matched with at most c_f workers.

We will write $\mu(w)$ to denote the firm that worker w is matched with under μ , i.e., $\mu(w) := \{f \in F : \mu(w, f) = 1\}$. Similarly, we will write $\mu(f)$ to denote the set of workers that are matched with firm f under μ , i.e., $\mu(f) := \{w \in W : \mu(w, f) = 1\}$. Thus, $|\mu(w)| \leq 1$ and $|\mu(f)| \leq c_f$.

For the special case when $c_f = 1$ for all firms $f \in F$, we obtain the *one-to-one matching problem*.

For simplicity, we will use the term *matching* to denote a ‘many-to-one matching’, and will explicitly use the qualifiers ‘one-to-one’ and ‘many-to-one’ when the distinction between the two is necessary. Further, the term *agent* will refer to a worker or a firm, i.e., an entity in the set $W \cup F$.

Utilities. Given a matching μ , the *utility* of a worker w under μ is its value for the firm that it is matched with, i.e., $u_w(\mu) := \sum_{f \in F} v_{w,f} \cdot \mu(w, f)$. Similarly, the utility of a firm f under μ is the sum of its values for the workers that it is matched with, i.e., $u_f(\mu) := \sum_{w \in W} v_{f,w} \cdot \mu(w, f)$. In other words, the firms’ utilities are assumed to be *additive*.

Welfare measures. We will now define two welfare measures that can be associated with a matching μ .

- *Utilitarian social welfare* is the sum of the utilities of the agents under μ , i.e., $\mathcal{W}^{\text{util}}(\mu) := \sum_{i \in W \cup F} u_i(\mu)$.
- *Nash social welfare* is the geometric mean of the utilities of the agents under μ , i.e.,

$$\mathcal{W}^{\text{Nash}}(\mu) := \left(\prod_{i \in W \cup F} u_i(\mu) \right)^{1/(n+m)}.$$

We will use the term *Nash product* to denote the product of agents’ utilities.

For any $\alpha \in [0, 1]$ and any welfare measure \mathcal{W} , a matching μ is said to be α - \mathcal{W} *optimal* if its welfare is at least α times the maximum welfare achieved by any matching for the given instance. When $\alpha = 1$, we use the term \mathcal{W} *optimal*, e.g., utilitarian optimal or Nash optimal.

The case of zero Nash welfare. Since we allow the agents to have zero valuations, it is possible that every matching for a given instance has zero Nash welfare. In the full version (Jain and Vaish 2023), we give a polynomial-time algorithm for identifying such instances. This algorithm uses a natural linear program for many-to-one matchings along with the rounding technique of Budish et al. (2013) and Aziz et al. (2023).

Proposition 1. *There is a polynomial-time algorithm that, given any two-sided matching instance, determines whether there exists a matching with nonzero Nash welfare.*

The ‘zerness’ of Nash welfare turns out to be important in analyzing its axiomatic properties in the fair division literature. Indeed, for an instance where all allocations have zero Nash welfare, a Nash optimal allocation is defined in terms of the largest set of agents that can have positive utility (Caragiannis et al. 2019). This refinement is necessary for demonstrating the approximate envy-freeness property of a Nash optimal allocation. Since our focus in this work is on computational—rather than axiomatic—properties, we will assume that instances where the optimal Nash welfare is zero are discarded. Thus, in the forthcoming sections, we will assume that any given instance admits some matching with nonzero Nash welfare. Equivalently, there is some matching in which all agents have positive utility. Note that this condition enforces that the number of workers is at most the total capacity of the firms, i.e., $m \leq \sum_{f \in F} c_f$.

Fair division as a special case. An important special case of our two-sided model is the problem of fair division with indivisible items. An instance of this problem consists of a set of agents with numerical valuations over a set of indivisible items. The goal is to partition the items among the agents subject to fairness constraints, such as finding an allocation that maximizes Nash welfare. When each worker values every firm at 1 and the firms have unrestricted capacities, we obtain fair division as a special case of our problem.

Relevant parameters. In addition to natural parameters such as the number of workers, number of firms, and firms’ capacities, we will use other structural parameters in our analysis. Consider the bipartite graph (similar to the one shown in Figure 1) associated with any matching instance. Suppose all ‘0—0’ edges, wherein a worker and firm value each other at 0, are removed. The *degree* of an agent is defined as the number of edges incident to the corresponding vertex in the residual graph. Thus, the degree of an agent gives an upper bound on the number of other agents that it positively values. We also consider the *number of distinct valuations* parameter, defined as the number of distinct elements in the set $\{v_{w,f}\}_{(w,f) \in W \times F} \cup \{v_{f,w}\}_{(w,f) \in W \times F}$.

All omitted proofs are in the full version of the paper (Jain and Vaish 2023).

Hardness Results

In this section, we will prove our main negative result (Theorem 1) concerning the hardness of maximizing Nash welfare under two-sided preferences.

To motivate our hardness result, let us first consider an easy case where the capacity of each firm is 1. In this case, a Nash optimal matching can be computed in polynomial time (Proposition 2). The algorithm is fairly natural: It computes a maximum weight matching in a bipartite graph whose vertices are the workers and the firms, and the weight of the edge between worker w and firm f is $\log(v_{w,f} \cdot v_{f,w})$ if both $v_{w,f}$ and $v_{f,w}$ are positive, and 0 otherwise.

Proposition 2. *When every firm has capacity 1, a Nash optimal matching can be computed in polynomial time.*

Interestingly, the problem becomes significantly harder when the firms can have capacity 2 (Theorem 1). On first thought, it may seem that the hardness arises because of *large* valuations. Or, if the valuations are small, the difficulty may be due to balancing a *large number* of small values for each firm. However, we show that the hardness persists even when all valuations are in the range $\{0, 1, 2\}$ (also known as *ternary* or *3-value* instances) and each agent has *degree* at most 3; thus, it positively values at most three other agents.

Theorem 1 (NP-hardness under constant capacities). *Computing a Nash optimal matching is NP-complete even when each firm has capacity 2, all valuations are in $\{0, 1, 2\}$, and each agent has degree at most 3.*

Before discussing the proof of Theorem 1, let us compare it with some known results on the hardness of maximizing Nash welfare in the fair division literature. Amanatidis et al. (2021) have shown that maximizing Nash welfare is NP-hard for 3-value instances when all valuations are in the range $\{0, 1, a\}$ for some $a > 1$. The parameter ‘ a ’ in their construction *depends* on the size of the instance; specifically, they use $a > 1 / \sqrt[2m]{2} - 1$, where m is the number of clauses in the 3-SAT instance which they reduce from. By contrast, our reduction *does not* require the values to grow with the size of the instance; indeed, all valuations in our construction are in the range $\{0, 1, 2\}$.

Nguyen et al. (2014) showed NP-hardness of approximating Nash welfare to within a factor $\frac{8}{9}$ via reduction from EXACT COVER BY THREE SETS (X3C). Their reduction requires the capacities to be 3 (instead of 2) and the number of positively valued agents to grow with the problem size (instead of being constant). Finally, Garg, Hoefer, and Mehlhorn (2018) showed NP-hardness of approximating Nash welfare to within a factor $\sqrt{7/8}$ via a reduction from a variant of MAXIMUM LINEAR EQUATION problem. The capacities in their reduction are required to be 4 (instead of 2) and the valuations are in $\{0, 1, k\}$ for a large constant k (even under a conservative calculation, $k \geq 8$).

Let us now briefly sketch the proof of Theorem 1.

Proof sketch. (of Theorem 1) Our reduction is from RAINBOW PERFECT MATCHING. The input to this prob-

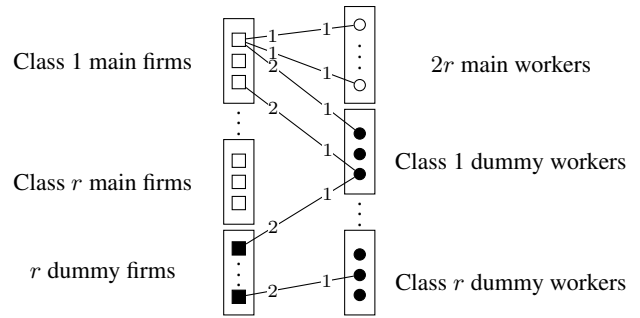


Figure 2: The reduced two-sided matching instance in the proof of Theorem 1. Firms are denoted by squares and workers by circles. The shaded and unshaded nodes denote the dummy and the main agents, respectively.

lem consists of a bipartite multigraph whose vertex sets are of size r each and whose edge set is partitioned into r color classes. The goal is to determine if there is a perfect matching consisting of one edge of each color. In the full version of the paper (Jain and Vaish 2023), we show that this problem is NP-hard even when each vertex has degree 3 and there are three edges of each color. These restrictions on RAINBOW PERFECT MATCHING are needed in our proof for obtaining the bound on degree in Theorem 1.

The reduced instance consists of a *main firm* for every edge and a *main worker* for every vertex of the given multigraph. Additionally, there are three *dummy workers* and one *dummy firm* for each color class. The main workers only value the main firm corresponding to their incident edges, while the main firms have a slightly higher value for dummy workers; see Figure 2. Each dummy firm only values the dummy workers of the same color class and has zero value for all other workers. The capacity of every firm is 2, and the threshold for Nash welfare is $\theta := 2^{\#\text{firms}/\#\text{agents}}$. That is, the goal is to determine if there is a feasible matching with Nash welfare at least θ . Note that there are $4r$ firms and $5r$ workers in the reduced instance; thus $9r$ agents in total.

Given a rainbow perfect matching, the desired Nash welfare solution can be obtained by assigning the main workers to the firms corresponding to the matched edges. This leaves two main firms in each color class, who each absorb one dummy worker, leaving the third dummy worker for a dummy firm. Thus, all firms get a utility of 2, as desired.

For the reverse direction, we use the AM-GM inequality. Indeed, each dummy worker contributes 2 to the sum of firms’ utilities, while each main worker contributes 1. Thus, the arithmetic mean of firms’ utilities is *exactly* 2. The decision threshold of θ forces the geometric mean to also be exactly 2, implying that every firm gets a utility of 2. This naturally induces a rainbow matching. \square

By reducing from the PARTITION problem, we can show NP-hardness of maximizing Nash welfare even for two firms with identical valuations and equal capacities and even under *symmetric* valuations (i.e., if for every worker w and firm f , $v_{w,f} = v_{f,w}$). This reduction is well-known in the

fair division literature; Proposition 3 simply adapts it to two-sided preferences setting.

Proposition 3 (NP-hardness for two identical firms). *Computing a Nash optimal matching is NP-complete even with two identical firms (that have the same valuation functions and equal capacities) and under symmetric valuations.*

A slight modification of the same reduction gives NP-hardness for the case when all agents have *strict* preferences.

Corollary 1 (NP-hardness under strict preferences). *Computing a Nash optimal matching is NP-complete even when all agents have strict preferences.*

Approximation Algorithms

The intractability of maximizing Nash welfare motivates the study of approximation algorithms in search of positive results. To this end, we provide two algorithms: The first algorithm works for an arbitrary number of firms, but gives a somewhat weak approximation (Theorem 2). The second algorithm provides a stronger approximation for a constant number of firms (Theorem 3 and Corollary 3).

Our first main result in this section is a $\frac{1}{\sqrt{\text{opt}}}$ -approximation algorithm, where opt is the optimal Nash welfare in the given instance. The algorithm requires all valuations to be positive, i.e., for every worker $w \in W$ and every firm $f \in F$, $v_{w,f} > 0$ and $v_{f,w} > 0$.

Theorem 2 ($\frac{1}{\sqrt{\text{opt}}}$ -approximation). *There is a polynomial-time algorithm that, given any matching instance with positive valuations, returns a $\frac{1}{\sqrt{\text{opt}}}$ -Nash optimal matching, where opt is the optimal Nash welfare.*

The approximation factor in Theorem 2 *decreases* as a function of the optimal Nash welfare and provides only a weak guarantee as the problem size grows. Nevertheless, when the valuations are bounded by a constant and each firm has a constant capacity, the maximum utility of any agent—and thus their geometric mean—is also constant. In this setting, Theorem 2 gives a *constant-factor* approximation.

Corollary 2 (Constant approximation for constant valuations and capacities). *Let c_0 be a constant. There is a polynomial-time algorithm that, given any matching instance with positive valuations where all valuations and capacities are bounded by c_0 , returns an α -Nash optimal matching for some constant α (that only depends on c_0).*

Recall that even when all valuations and capacities are bounded by a constant, the problem of maximizing Nash welfare is NP-hard (Theorem 1). Whether there exists a polynomial-time approximation scheme (PTAS) in this setting is an interesting open problem. Later in this section, we will present a *quasipolynomial-time* approximation scheme or QPTAS for a constant number of firms (Theorem 3).

Proof sketch. (of Theorem 2) We will show that the task of approximating Nash welfare in the matching problem reduces to approximating *utilitarian* welfare in the *fair division* problem under *submodular* valuations and *matroid* constraints. Specifically, we let the firms and workers play the

role of agents and items, respectively, in the fair division problem. For each firm f and any subset $X \subseteq W$ of workers, the *modified* valuation of ‘agent’ f for the set of ‘items’ X is defined as $\hat{v}_f(X) := \log(v_f(X) \cdot \prod_{w \in X} v_{w,f})$.

The modified valuation function $\hat{v}_f(\cdot)$ captures the combined contribution of firm f and the set of workers X to the log of Nash welfare objective. It is easy to show that an α -utilitarian optimal allocation (under modified valuations) induces a matching that is $\frac{1}{\text{opt}^{1-\alpha}}$ -Nash optimal.

The key observation is that the function $\hat{v}_f(\cdot)$ is *submodular*. Furthermore, capacity constraints in the matching problem can be captured by matroid constraints in the fair division problem. We can now use the natural greedy algorithm for submodular maximization to obtain a feasible and $\frac{1}{2}$ -utilitarian optimal allocation with respect to the modified valuations (Fisher, Nemhauser, and Wolsey 1978). The desired approximation for Nash welfare follows. \square

It would be very interesting to know if a constant-factor approximation algorithm can be designed for our problem. Such algorithms are known for the one-sided fair division problem. However, as the example in Figure 1 suggests, incorporating the preferences of the ‘items’ can come at the expense of making the ‘agents’ worse off. This makes the design of algorithms for the two-sided problem challenging.

Our second main result in this section uses the idea of *bucketing* (or discretization) to classify the workers and firms according to the *range* in which they value each other.

Specifically, given any $\varepsilon > 0$ and any firm f , define a set of $\tau + 1$ *buckets* $B_0^f, B_1^f, B_2^f, \dots, B_\tau^f$, where $\tau := \lceil \log_{1+\varepsilon} v_{\max} \rceil$ and v_{\max} is the maximum valuation of any agent for any other agent. For any $i \in [\tau]$, the bucket B_i^f denotes the set of workers who value firm f between $(1+\varepsilon)^{i-1}$ and $(1+\varepsilon)^i$. Bucket B_0^f contains workers who value firm f at 0. For each bucket B_i^f , we further define $\tau + 1$ *sub-buckets* $b_0^{f,i}, b_1^{f,i}, \dots, b_\tau^{f,i}$, where, for any $j \in [\tau]$, the sub-bucket $b_j^{f,i}$ denotes the set of workers in the bucket B_i^f whom the firm f values between $(1+\varepsilon)^{j-1}$ and $(1+\varepsilon)^j$. The workers in B_i^f valued at 0 are assigned to $b_0^{f,i}$. Thus, membership in a sub-bucket specifies the valuations of a worker and a firm for each other within a multiplicative factor of $(1+\varepsilon)$.

Our algorithm *guesses* the number of workers in each sub-bucket in an optimal solution. Each such guess, if feasible for the given capacities, induces a matching. The Nash welfare of this matching can be correctly computed to within a multiplicative factor of $(1+\varepsilon)$ by only knowing the *number* of workers in each sub-bucket. The total number of sub-buckets is $\mathcal{O}(n\tau^2)$. For m workers, the total number of guesses is at most $m^{\mathcal{O}(n\tau^2)}$; the calculation is analogous to distributing m identical balls into $n(\tau+1)^2$ bins. For each guess, the algorithm checks feasibility and computes Nash welfare in polynomial time. The guess with the maximum Nash welfare is returned as the solution. For *polynomially-bounded* valuations, i.e., when $v_{\max} \leq \text{poly}(m, n)$, we obtain a QPTAS for a constant number of firms.

Theorem 3 (QPTAS for constant no. of firms). *There is an algorithm that, given any $\varepsilon > 0$ and any matching in-*

stance with a constant number of firms and polynomially-bounded valuations, runs in $\mathcal{O}(m^{\mathcal{O}(\frac{1}{\varepsilon^2} \cdot \log^2 m)})$ time and returns a $\frac{1}{(1+\varepsilon)}$ -Nash optimal matching.

By bucketing on the exact valuations instead of powers of $(1 + \varepsilon)$, we obtain a polynomial-time exact algorithm for the case when n and v_{\max} are both constant.

Corollary 3 (Exact algorithm for constant no. of firms and constant valuations). *There is an algorithm that, given any matching instance with a constant number of firms and the maximum valuation v_{\max} bounded by a constant, runs in $\mathcal{O}(\text{poly}(m))$ time and returns a Nash optimal matching.*

Note that the problem is NP-hard for a constant number of firms but with possibly large valuations (Proposition 3).

Parameterized Algorithms

We will now study the problem in the realm of parameterized complexity and discuss two main results: An exact algorithm that is fixed-parameter tractable (FPT) in the number of workers m with running time $\mathcal{O}^*(3^m)$ (Theorem 4), and a faster FPT approximation algorithm with running time $\mathcal{O}^*(2^m)$ (Theorem 5).

We have already shown that maximizing Nash welfare is NP-hard even when the maximum capacity of any firm (denoted by cap) or the number of firms n is constant. Thus, a natural next step is to study the combined parameter $n + \text{cap}$. By the nonzeroness of Nash welfare, we know that $m \leq n \cdot \text{cap}$. Thus, designing an FPT algorithm in m automatically gives an FPT algorithm in $n \cdot \text{cap}$, and thus also FPT in $n + \text{cap}$.

Note that designing some FPT-in- m algorithm is straightforward: Since the number of firms is at most the number of workers (i.e., $n \leq m$), a brute force search over the space of all feasible matchings can be done in $\mathcal{O}^*(m^m)$ time. Our goal in this section is to obtain FPT-in- m algorithms with significantly better running times.

The following notation will be useful in proving both results in this section: For any firm f and any subset of workers $S \subseteq W$, define $\mathcal{W}_f(S) := v_f(S) \cdot \prod_{w \in S} v_w(f)$. Recall that the log of the function $\mathcal{W}_f(\cdot)$ was used in the proof of Theorem 2 in defining the modified valuation function.

Let us now state our first main result in this section.

Theorem 4 (FPT exact algorithm). *A Nash optimal matching can be computed in $\mathcal{O}^*(3^m)$ time.*

To prove Theorem 4, we will use dynamic programming. Let the firms be indexed as f_1, \dots, f_n . Define a table T with n rows (one for each firm) and 2^m columns (one for each subset of workers). The entry $T(i, S)$ will contain the value of the maximum Nash product achievable in a subinstance consisting of the first i firms and the subset S of the workers.

More concretely, for every subset $S \subseteq W$, the first row of the table can be computed as follows:

$$T[1, S] = \begin{cases} \mathcal{W}_{f_1}(S) & \text{if } |S| \leq c_1, \\ 0 & \text{otherwise.} \end{cases}$$

Further, for any $i > 1$ and any subset $S \subseteq W$, we have

$$T[i, S] = \max_{S' \subseteq S, |S'| \leq c_i} \mathcal{W}_{f_i}(S') \times T[i-1, S \setminus S'].$$

Once the entry $T[n, W]$ is computed correctly, we can use backtracking to obtain the Nash optimal matching. To compute the entry $T[i, S]$, we may need to check all subsets of S in the worst case. Thus, the running time of the algorithm is $\mathcal{O}(n \sum_{i=0}^m \binom{m}{i} 2^i)$, or $\mathcal{O}(n \cdot 3^m)$.

The running time of our DP algorithm can be improved if the capacity of each firm is bounded by a constant. In this case, we only need to consider subsets of constant size, implying that each entry of the table can be computed in polynomial time. Theorem 5 formalizes this observation.

Theorem 5 ($\mathcal{O}^*(2^m)$ algorithm for constant capacity). *When every firm has a constant capacity, a Nash optimal matching can be computed in $\mathcal{O}^*(2^m)$ time.*

Next, we will present a parameterized approximation algorithm that, given any $\varepsilon \in (0, 1]$, finds a $(1 + \varepsilon)^{-(n+1)/m+n}$ -approximate solution in $\mathcal{O}^*(1/\varepsilon^4 \cdot 2^m)$ time for arbitrary capacities. The algorithm uses the techniques of polynomial multiplication and binning.

Given any instance $\langle W, F, C, \mathcal{V} \rangle$, let μ and η denote a Nash optimal matching and the maximum Nash product, respectively. For every $i \in [n]$, let \mathcal{S}_i denote the set of all subsets of workers that can be feasibly assigned to the firm f_i , i.e., \mathcal{S}_i contains all subsets of workers of size at most c_i .

The basic idea is as follows: For each firm $f \in F$, guess $\mathcal{W}_f(\mu(f))$, remove the set S from \mathcal{S}_i if $\mathcal{W}_f(S)$ is not same as the guessed value, and then find n disjoint sets, one from each \mathcal{S}_i . The disjoint sets can be found using polynomial multiplication in $\mathcal{O}^*(2^m)$ time. Unfortunately, this will not lead to the desired running time as guessing depends on the Nash product. Thus, we will create $(1 + \varepsilon)$ -sized bins for the Nash product and obtain an approximation algorithm.

Before we discuss our algorithm, let us introduce some relevant definitions. The characteristic vector of a subset $S \subseteq [m]$, denoted by $\chi(S)$, is an m -length binary string whose i^{th} bit is 1 if and only if $i \in S$. Two binary strings of length m are said to be disjoint if for each $i \in [m]$, the i^{th} bits in the two strings are different. The Hamming weight of a binary string S , denoted by $\mathcal{H}(S)$, is the number of 1s in the string S . A monomial y^i is said to have Hamming weight w if the degree term i , when represented as a binary string, has Hamming weight w . The Hamming projection of a polynomial $p(y)$ to h , denoted by $\mathcal{H}_h(p(y))$, is the sum of all the monomials of $p(y)$ which have Hamming weight h . The representative polynomial of $p(y)$, denoted by $\mathcal{R}(p(y))$, is a polynomial derived from $p(y)$ by changing the coefficients of all monomials contained in it to 1.

We begin with the following known result.

Proposition 4 (Gupta et al. 2021; Cygan and Pilipczuk 2010). *Subsets $S_1, S_2 \subseteq W$ are disjoint if and only if Hamming weight of the monomial $x^{\chi(S_1) + \chi(S_2)}$ is $|S_1| + |S_2|$.*

Let us now state our main result.

Theorem 6 (FPT approximation scheme). *There is an algorithm that, given any matching instance and any $\varepsilon \in (0, 1]$, returns a $(1 + \varepsilon)^{-(n+1)/m+n}$ -Nash optimal matching in $\mathcal{O}^*(1/\varepsilon^4 \cdot 2^m)$ time.*

Proof. Let $\mathcal{I} = \langle W, F, C, \mathcal{V} \rangle$ be the given instance, where $F = \{f_1, \dots, f_n\}$ and $W = \{1, \dots, m\}$ are the sets of firms and workers, respectively. Let η be the maximum possible Nash product for \mathcal{I} ; thus, $\eta \leq (mv_{\max})^{m+n}$. Let $Z := \{1, (1+\varepsilon), (1+\varepsilon)^2, \dots, (1+\varepsilon)^q, (1+\varepsilon)^{q+1}\}$, where q is the largest positive integer such that $(1+\varepsilon)^q \leq \eta$. For every $j \in [n]$, $s \in [m]$, and $\ell \in Z$, we will construct a polynomial $p_{s,\ell}^j$ in which every nonzero monomial corresponds to an assignment of s workers to the first j firms f_1, \dots, f_j such that the Nash product is at least ℓ .

We will construct these polynomials $p_{s,\ell}^j$ iteratively. First, we will construct a polynomial $h_{s,\ell}^j$ in which every nonzero monomial corresponds to an assignment of some set $X \subseteq W$ of s workers to the j^{th} firm f_j such that $\mathcal{W}_{f_j}(X) \geq \ell$. That is, for every $j \in [n]$, $s \in [c_j]$, and $\ell \in Z$,

$$h_{s,\ell}^j := \sum_{\substack{X \subseteq W, |X|=s, \\ \mathcal{W}_{f_j}(X) \geq \ell}} y^{X(X)}. \quad (1)$$

Define $p_{s,\ell}^1 := h_{s,\ell}^1$. For every $j \in \{2, \dots, n\}$, $s \in [m]$ and $\ell \in Z$, define

$$p_{s,\ell}^j := \mathcal{R} \left(\mathcal{H}_s \left(\sum_{\substack{s=s'+s'', s' \leq c_j, \\ \ell = \ell' \times \ell'', \ell', \ell'' \in Z}} h_{s',\ell'}^j \times p_{s'',\ell''}^{j-1} \right) \right), \quad (2)$$

where $\mathcal{R}(\cdot)$ is the representative polynomial and $\mathcal{H}_s(\cdot)$ is the Hamming projection of weight s . The $\mathcal{H}(\cdot)$ operator ensures disjointness due to Proposition 4. The $\mathcal{R}(\cdot)$ operator is only required for the running time.

We return the largest value of ℓ for which $p_{m,\ell}^n$ is nonzero. The corresponding matching can be found by backtracking.

To argue correctness, we will prove that if the algorithm returns ℓ^* , then $\eta \leq (1+\varepsilon)^{n+1}\ell^*$. Towards this, we show that if $(1+\varepsilon)^q \leq \eta \leq (1+\varepsilon)^{q+1}$, then $p_{m,\ell}^n$ is nonzero for some $\ell \in \{(1+\varepsilon)^{q-n}, \dots, (1+\varepsilon)^{q+1}\}$. Thus, we return a matching $\tilde{\mu}$ with Nash product at least $(1+\varepsilon)^{q-n}$. Since, $\eta \leq (1+\varepsilon)^{q+1}$, it follows that $\eta \leq (1+\varepsilon)^{n+1}\ell^*$. Thus, $\mathcal{W}^{\text{Nash}}(\mu) \leq (1+\varepsilon)^{\frac{n+1}{m+n}} \mathcal{W}^{\text{Nash}}(\tilde{\mu})$, where μ is a Nash optimal matching. The detailed proof of correctness can be found in the full version (Jain and Vaish 2023).

Since $|Z| = \mathcal{O}(\log_{1+\varepsilon} \eta)$, we compute $\mathcal{O}(nm \log_{1+\varepsilon} \eta)$ polynomials. Note that the degree of the polynomial is at most 2^m (as the the m -length binary vector in the polynomial can have all 1s). Each polynomial can be computed in $\mathcal{O}(m^2 \cdot 2^m \cdot \log \eta)$ time due to the following result and the fact that $s \leq m$ and $\ell \in Z$.

Proposition 5 (Moenck 1976). *Two polynomials of degree d can be multiplied in $\mathcal{O}(d \log d)$ time.*

Since $\eta \leq (mv_{\max})^{m+n}$, $\log_{1+\varepsilon} \eta \leq (m+n) \log_{1+\varepsilon}(mv_{\max})$. By changing the base of logarithm from $(1+\varepsilon)$ to 2, and the fact that $\log(1+\varepsilon) > \varepsilon^2/2$, for every $\varepsilon \in (0, 1]$, we get the running time of $\mathcal{O}^*(1/\varepsilon^4 \cdot 2^m)$. \square

Restricted Domains

In this section, we will design polynomial-time algorithms for restricted domains. Our first result is an algorithm for *symmetric binary valuations*, which is when, for every worker-firm pair $(w, f) \in W \times F$, either $v_{w,f} = v_{f,w} = 1$ or $v_{w,f} = v_{f,w} = 0$.

Theorem 7. *For symmetric binary valuations, a Nash optimal matching can be computed in polynomial time.*

We use the algorithm of Barman, Krishnamurthy, and Vaish (2018b) which maximizes Nash welfare in the one-sided fair division problem under binary valuations. Starting with a suboptimal allocation, their algorithm greedily picks a sequence of item transfers between agents to improve the Nash objective. We follow the same strategy, with the difference that the initial matching in our case is chosen using the algorithm in Proposition 1 to ensure nonzero Nash welfare.

Note that for general symmetric valuations, the problem remains NP-hard (Proposition 3). Whether an efficient algorithm is possible for (possibly asymmetric) binary valuations is an interesting problem for future work.

We will now discuss bounded degree instances.

Theorem 8. *A Nash optimal matching can be computed in polynomial time when:*

1. *all agents have degree at most 2,*
2. *all firms have degree at most 3 and exactly two workers need to be assigned to every firm,*
3. *every worker values only one firm positively, and*
4. *the number of firms and the number of distinct valuations are constant.*

Furthermore, when the number of firms is constant and the number of distinct valuations is logarithmically bounded in the input size (i.e., at most $\log(\text{poly}(m, n, v_{\max}))$), a Nash optimal matching can be computed in quasipolynomial time.

The first three results in Theorem 8 provide tractable subcases vis-à-vis the NP-hardness result in Theorem 1 for degree at most 3 and capacity 2. The fourth result is in contrast with NP-hardness for two firms when the number of distinct valuations can be large (Proposition 3).

Concluding Remarks

We have initiated a systematic study of the computation of Nash optimal many-to-one matchings under two-sided preferences. Our work contributes a variety of hardness and algorithmic results, spanning a broad range of techniques including polynomial multiplication, submodular fair division, rainbow perfect matching, etc. (see Table 1).

A number of intriguing open problems remain. Designing a constant-factor approximation algorithm for our problem and developing lower bounds in terms of natural parameters are two outstanding questions. Additionally, studying Nash optimal solutions in conjunction with established solution concepts such as *stability* and *popularity* will also be of interest (our hardness result in Theorem 1 extends to these concepts). Finally, analogous to the literature in fair division, it would be interesting to find axiomatic justification for the use of Nash welfare in the two-sided setting.

Acknowledgments

We are grateful to the anonymous reviewers for helpful comments. PJ acknowledges support from SERB-SUPRA grant no. S/SERB/PJ/20220047 and IITJ Seed Grant grant no. I/SEED/PJ/20210119. RV acknowledges support from DST INSPIRE grant no. DST/INSPIRE/04/2020/000107 and SERB grant no. CRG/2022/002621.

References

- Abdulkadiroğlu, A.; and Sönmez, T. 2003. School Choice: A Mechanism Design Approach. *American Economic Review*, 93(3): 729–747.
- Akrami, H.; Chaudhury, B. R.; Hoefer, M.; Mehlhorn, K.; Schmalhofer, M.; Shahkarami, G.; Varricchio, G.; Vermande, Q.; and van Wijland, E. 2022. Maximizing Nash Social Welfare in 2-Value Instances. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence*, volume 36, 4760–4767.
- Amanatidis, G.; Aziz, H.; Birmpas, G.; Filos-Ratsikas, A.; Li, B.; Moulin, H.; Voudouris, A. A.; and Wu, X. 2023. Fair Division of Indivisible Goods: Recent Progress and Open Questions. *Artificial Intelligence*, 103965.
- Amanatidis, G.; Birmpas, G.; Filos-Ratsikas, A.; Hollender, A.; and Voudouris, A. A. 2021. Maximum Nash Welfare and Other Stories about EFX. *Theoretical Computer Science*, 863: 69–85.
- Anari, N.; Mai, T.; Gharan, S. O.; and Vazirani, V. V. 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.
- Aziz, H.; Freeman, R.; Shah, N.; and Vaish, R. 2023. Best of Both Worlds: Ex-Ante and Ex-Post Fairness in Resource Allocation. *Operations Research*.
- Barman, S.; Bhaskar, U.; Krishna, A.; and Sundaram, R. G. 2020. Tight Approximation Algorithms for p -Mean Welfare Under Subadditive Valuations. In *Proceedings of the 28th Annual European Symposium on Algorithms*.
- Barman, S.; Krishna, A.; Kulkarni, P.; and Narang, S. 2021. Sublinear Approximation Algorithm for Nash Social Welfare with XOS Valuations. *arXiv preprint arXiv:2110.00767*.
- Barman, S.; Krishnamurthy, S. K.; and Vaish, R. 2018a. Finding Fair and Efficient Allocations. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, 557–574.
- Barman, S.; Krishnamurthy, S. K.; and Vaish, R. 2018b. Greedy Algorithms for Maximizing Nash Social Welfare. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, 7–13.
- Brams, S. J.; and Taylor, A. D. 1996. *Fair Division: From Cake-Cutting to Dispute Resolution*. Cambridge University Press.
- Brandt, F.; Conitzer, V.; Endriss, U.; Lang, J.; and Procaccia, A. D. 2016. *Handbook of Computational Social Choice*. Cambridge University Press.
- Budish, E.; Che, Y.-K.; Kojima, F.; and Milgrom, P. 2013. Designing Random Allocation Mechanisms: Theory and Applications. *American Economic Review*, 103(2): 585–623.
- Caragiannis, I.; Kurokawa, D.; Moulin, H.; Procaccia, A. D.; Shah, N.; and Wang, J. 2019. The Unreasonable Fairness of Maximum Nash Welfare. *ACM Transactions on Economics and Computation*, 7(3): 1–32.
- Chaudhury, B. R.; Garg, J.; and Mehta, R. 2021. Fair and Efficient Allocations under Subadditive Valuations. In *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence*, volume 35, 5269–5276.
- Cole, R.; Devanur, N.; Gkatzelis, V.; Jain, K.; Mai, T.; Vazirani, V. V.; and Yazdanbod, S. 2017. Convex Program Duality, Fisher Markets, and Nash Social Welfare. In *Proceedings of the 2017 ACM Conference on Economics and Computation*, 459–460.
- Cole, R.; and Gkatzelis, V. 2018. Approximating the Nash Social Welfare with Indivisible Items. *SIAM Journal on Computing*, 47(3): 1211–1236.
- Cook, C.; Diamond, R.; Hall, J. V.; List, J. A.; and Oyer, P. 2021. The Gender Earnings Gap in the Gig Economy: Evidence from Over a Million Rideshare Drivers. *The Review of Economic Studies*, 88(5): 2210–2238.
- Cygan, M.; and Pilipczuk, M. 2010. Exact and Approximate Bandwidth. *Theoretical Computer Science*, 411(40–42): 3701–3713.
- Eisenberg, E.; and Gale, D. 1959. Consensus of Subjective Probabilities: The Pari-Mutuel Method. *The Annals of Mathematical Statistics*, 30(1): 165–168.
- Fisher, M. L.; Nemhauser, G. L.; and Wolsey, L. A. 1978. An Analysis of Approximations for Maximizing Submodular Set Functions—II. *Mathematical Programming Studies*, 8: 73–87.
- Gale, D.; and Shapley, L. S. 1962. College Admissions and the Stability of Marriage. *The American Mathematical Monthly*, 69(1): 9–15.
- Garg, J.; Hoefer, M.; and Mehlhorn, K. 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.
- Garg, J.; HusiĆ, E.; Li, W.; Végh, L. A.; and Vondrák, J. 2023. Approximating Nash Social Welfare by Matching and Local Search. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, 1298–1310.
- Garg, J.; Husic, E.; Murhekar, A.; and Végh, L. 2022. Tractable Fragments of the Maximum Nash Welfare Problem. In *Proceedings of the 18th International Conference on Web and Internet Economics*, volume 13778, 362.
- Garg, J.; Kulkarni, P.; and Kulkarni, R. 2020. Approximating Nash Social Welfare under Submodular Valuations through (Un) Matchings. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, 2673–2687.

- Gupta, S.; Jain, P.; Saurabh, S.; and Talmon, N. 2021. Even More Effort Towards Improved Bounds and Fixed-Parameter Tractability for Multiwinner Rules. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, 217–223.
- Hitsch, G. J.; Hortaçsu, A.; and Ariely, D. 2010. Matching and Sorting in Online Dating. *American Economic Review*, 100(1): 130–163.
- Jain, P.; and Vaish, R. 2023. Maximizing Nash Social Welfare under Two-Sided Preferences. *arXiv preprint arXiv:2312.09167*.
- Kaneko, M.; and Nakamura, K. 1979. The Nash Social Welfare Function. *Econometrica: Journal of the Econometric Society*, 423–435.
- Lee, E. 2017. APX-Hardness of Maximizing Nash Social Welfare with Indivisible Items. *Information Processing Letters*, 122: 17–20.
- Manlove, D. 2013. *Algorithmics of Matching under Preferences*, volume 2. World Scientific.
- Moenck, R. T. 1976. Practical Fast Polynomial Multiplication. In *Proceedings of the Third ACM Symposium on Symbolic and Algebraic Computation*, 136–148.
- Moulin, H. 2004. *Fair Division and Collective Welfare*. MIT press.
- Nash Jr, J. F. 1950. The Bargaining Problem. *Econometrica: Journal of the Econometric Society*, 155–162.
- Nguyen, N.-T.; Nguyen, T. T.; Roos, M.; and Rothe, J. 2014. Computational Complexity and Approximability of Social Welfare Optimization in Multiagent Resource Allocation. *Autonomous Agents and Multi-Agent Systems*, 28(2): 256–289.
- Roth, A. E.; and Peranson, E. 1999. The Redesign of the Matching Market for American Physicians: Some Engineering Aspects of Economic Design. *American Economic Review*, 89(4): 748–780.
- Roth, A. E.; and Sotomayor, M. 1992. Two-Sided Matching. *Handbook of Game Theory with Economic Applications*, 1: 485–541.