Capacity Modification in the Stable Matching Problem

Salil Gokhale
Indian Institute of Technology Delhi
New Delhi, India
Salil.Gokhale.mt121@maths.iitd.ac.in

Samarth Singla
Indian Institute of Technology Delhi
New Delhi, India
mt6210942@iitd.ac.in

Shivika Narang
Simons Laufer Mathematical Science Institute
Berkeley, CA, United States
shivika@iisc.ac.in

Rohit Vaish
Indian Institute of Technology Delhi
New Delhi, India
rvaish@iitd.ac.in

ABSTRACT

We study the problem of capacity modification in the many-to-one stable matching of workers and firms. Our goal is to systematically study how the set of stable matchings changes when some seats are added to or removed from the firms. We make three main contributions: First, we examine whether firms and workers can improve or worsen upon changing the capacities under worker-proposing and firm-proposing deferred acceptance algorithms. Second, we study the computational problem of adding or removing seats to either match a fixed worker-firm pair in some stable matching or make a fixed matching stable with respect to the modified problem. We develop polynomial-time algorithms for these problems when only the overall change in the firms' capacities is restricted, and show NP-hardness when there are additional constraints for individual firms. Lastly, we compare capacity modification with the classical model of preference manipulation by firms and identify scenarios under which one mode of manipulation outperforms the other. We find that a threshold on a given firm's capacity, which we call its *peak*, crucially determines the effectiveness of different manipulation actions.

KEYWORDS

Stable matching; Capacity modification; Preference manipulation

ACM Reference Format:
Salil Gokhale, Samarth Singla, Shivika Narang, and Rohit Vaish. 2024. Capacity Modification in the Stable Matching Problem. In *Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), Auckland, New Zealand, May 6 – 10, 2024*, IFAAMAS, 9 pages.

1 INTRODUCTION

The stable matching problem is a classical problem at the intersection of economics, operations research, and computer science [21, 23, 30, 37]. The problem involves two sets of agents, such as workers and firms, each with a preference ordering over the agents on the other side. The goal is to find a matching that is *stable*, i.e., one where no worker-firm pair prefer each other over their current matches.

Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), N. Alechina, V. Dignum, M. Dastani, J.S. Sichman (eds.), May 6 – 10, 2024, Auckland, New Zealand.

Many real-world matching markets have been influenced by the stable matching problem, such as school choice [1–3], entry-level labor markets [33, 36], and refugee resettlement [4, 5]. In these applications, each agent on one side of the market (e.g., the firms) has a *capacity constraint* that limits the maximum number of agents on the other side (namely, workers) it can be feasibly matched with. Remarkably, for any given capacities, a stable matching of workers and firms always exists and can be computed using the celebrated deferred-acceptance algorithm [15, 33].

While the stable matching problem assumes fixed capacities, it is common to have *flexible* capacities in practice. This is particularly useful in settings with variable demand or popularity such as in vaccine distribution or course allocation. Flexible capacities also allow for accommodating other goals, such as Pareto optimality or social welfare [29]. For example, in 2016, nineteen colleges in Delhi University in India increased their total capacity by 2000 seats across various courses [12]. Another example is the *ScheduleScout* platform,[1] formerly known as Course Match [10], used in course allocation at the Wharton School. This platform allows the addition or removal of seats in courses that are either undersubscribed or oversubscribed, respectively.[2] In more complex matching environments such as stable matching with couples where a stable solution is not guaranteed to exist, a small change in the capacities can provably restore the existence of a stable outcome [31]. We will use the term *capacity modification* to refer to change in the capacities of the firms by a central planner.

The theoretical study of capacity modification was initiated by Sönmez [39], who showed that under any stable matching algorithm, there exists a scenario where some firm is better off when its capacity is reduced. The computational aspects of capacity modification have also recently gained attention [7, 8, 11]. However, some natural questions about how the set of stable matchings responds to changes in capacities have not been answered. Specifically, by modifying the capacities, can a given worker-firm pair be matched under some stable matching? Or, can a given matching be realized as a stable outcome of the modified instance? Furthermore, if we consider the perspective of a strategic firm, there has been a lack of a distinct comparison between "manipulation through capacity modification" and the traditional approach of "manipulation through misreporting preferences". Our interest in this work is to address these gaps.

[1] https://www.getschedulescout.com/
[2] https://www.youtube.com/watch?v=OSOanbdV3jI&t=1m38s

Our Contributions

We undertake a systematic analysis of the structural and computational aspects of the capacity modification problem and make three main contributions:

Capacity modification trends. In Section 3, we study the effect of capacity modification on workers and firms. We observe that increasing a firm's capacity by 1 can, in some cases, improve, and in other cases, worsen its outcome under both worker-proposing and firm-proposing deferred acceptance algorithms. The workers, on the other hand, can improve but never worsen (see Table 1).

Computational results. In Section 4, we study a natural computational problem faced by a central planner: Given a many-to-one instance, how can a fixed number of seats be added to (similarly, removed from) the firms in order to either match a fixed worker-firm pair in some stable matching or make a given matching stable in the new instance? We show that these problems admit polynomial-time algorithms. We also study a generalization where individual firms have constraints on the seats added to or removed from them, in addition to an aggregate budget. Here, the problem of matching a fixed worker-firm pair turns out to NP-hard while ensuring that a given matching is stable can still be efficiently solved (see Table 2).

Capacity modification v/s preference manipulation. In Section 5, we examine which mode of manipulation is more powerful for a strategic firm: underreporting/overreporting capacity or misreporting preferences. Interestingly, it turns out that the effectiveness of each manipulation action (i.e., adding/deleting capacity or misreporting preferences) depends on a threshold on the firm's capacity which we call *peak* (see Figure 1). For a firm to successfully manipulate its preferences, its capacity must be strictly below its peak (under the worker-proposing algorithm) or at most its peak (when firms propose). Thus, the concept of peak appears to have relevance beyond capacity modification.

All missing proofs and other technical details can be found in the full version [17].

Related Work

The stable matching problem has inspired a large body of work in economics, operations research, computer science, and artificial intelligence [15, 21, 23, 30, 37].

Prior work has demonstrated strategic vulnerabilities of stable matching algorithms. It is known that any stable matching algorithm is susceptible to manipulation via misreporting of preferences [13, 32], underreporting of capacities [39], and formation of pre-arranged matches [40].[3] Subsequently, Roth and Peranson [36] showed via experiments on the data from the National Resident Matching Program that less than 1% of the programs can benefit by misreporting preferences or underreporting capacities. Kojima and Pathak [27] provided theoretical justification for these findings by showing that incentives for such manipulations vanish in large markets. Note that, unlike the above results that only apply to specific datasets [36] or in the asymptotic setting [27], our algorithmic results provide worst-case guarantees for *any* given instance.

[3]In pre-arranged matches, a worker and firm can choose to match outside the algorithm. The worker does not participate in the algorithm, and in return, is offered a seat at the firm. The firm then has one less seat available through the algorithm.

Another line of work has explored restricted preference domains for circumventing the above impossibility results [22, 26, 28]. In particular, Konishi and Ünver [28] have shown that under strongly monotone preferences (formally defined in Section 2), a firm cannot manipulate by underreporting its capacity under the *worker-proposing algorithm* (although other algorithms, like the firm-proposing algorithm, can still be manipulated).

The computational problem of modifying capacities to serve a given objective has seen significant attention in recent years. Bobbio et al. [8] showed that the problem of adding (similarly, removing) seats from the firms in order to minimize the average rank of matched partners of the workers is NP-hard to approximate within $O(\sqrt{m})$, where m is the number of workers. Bobbio et al. [7] developed a mixed integer linear program for this problem.

Chen and Csáji [11] studied the problem of increasing the firms' capacities to obtain a stable and perfect matching, and similarly, a matching that is stable and Pareto efficient for the workers. They considered two objectives for this problem: minimizing the overall increase in the firms' capacities and minimizing the maximum increase in any firm's capacity. Dur and Van der Linden [14] studied the problem of adding seats to firms to achieve a matching that is stable (with respect to the modified capacities) and not Pareto dominated (as per workers' preferences only) by any other stable matching. Some of our computational results draw upon the work of Boehmer et al. [9], who studied the control problem for stable matchings in the one-to-one setting. We discuss the connection with this work in Section 4.

2 PRELIMINARIES

For any positive integer r, let $[r] \coloneqq \{1, 2, \ldots, r\}$.

Problem instance. An instance of the *many-to-one* matching problem is given by a tuple $\langle F, W, C, \succ \rangle$, where $F = \{f_1, \ldots, f_n\}$ is the set of $n \in \mathbb{N}$ *firms*, $W = \{w_1, \ldots, w_m\}$ is the set of $m \in \mathbb{N}$ *workers*, $C = \{c_1, \ldots, c_n\}$ is the set of *capacities* of the firms (where, for every $i \in [n]$, $c_i \in \mathbb{N} \cup \{0\}$), and $\succ = (\succ_{f_1}, \ldots, \succ_{f_n}, \succ_{w_1}, \ldots, \succ_{w_m})$ is the *preference profile* consisting of the ordinal preferences of all firms and workers. Each worker $w \in W$ is associated with a linear order (i.e., a strict and complete ranking) $\succ_w$ over the set $F \cup \{\emptyset\}$. Each firm $f \in F$ is associated with a linear order $\succ_f$ over the set $W \cup \{\emptyset\}$. Throughout, we will use the term *agent* to refer to a worker or a firm, i.e., an element in the set $W \cup F$.

For two capacity vectors $C, \overline{C} \in (\mathbb{N} \cup \{0\})^n$, we will write $\overline{C} \geq C$ to denote coordinate-wise greater than or equal to, i.e., for every $i \in [n]$, $\overline{c}_i \geq c_i$, where c_i and $\overline{c}_i$ are the i^{th} coordinate of vectors C and $\overline{C}$, respectively. Additionally, we will write $|\overline{C} - C|_1$ to denote the L^1 norm of the difference vector, i.e., $|\overline{C} - C|_1 \coloneqq \sum_{i=1}^{n} |\overline{c}_i - c_i|$.

When all firms have unit capacities (i.e., for each firm $f \in F$, $c_f = 1$), we obtain the *one-to-one* matching problem. In this case, we will follow the terminology from the literature on the stable marriage problem [15] and denote a problem instance by $\langle P, Q, \succ \rangle$, where P and Q denote the set of n *men* and m *women*, respectively, and $\succ$ denotes the corresponding preference profile.

Complete preferences. A worker w is said to be *acceptable* to a firm f if $w \succ_f \emptyset$. A set of workers $S \subseteq W$ is said to be acceptable to a firm f, denoted by $S \succ_f \emptyset$, if all workers in it are acceptable to f. Likewise, a firm f is acceptable to a worker w if $f \succ_w \emptyset$. An agent's preferences are said to be *complete* if all agents on the other side are acceptable to it.

Responsive preferences. Throughout the paper, we will assume that firms' preferences over subsets of workers are *responsive* [34]. Informally, this means that for any subsets $S, S' \subseteq W$ of workers where S is derived from S' by replacing a worker $w' \in S'$ with a more preferred worker w, it must be that $S \succ_f S'$. More formally, the extension of firm f's preferences over subsets of workers is responsive if for any subset $S \subseteq W$ of workers,

- for all $w \in W \setminus S$, $S \cup \{w\} \succ_f S$ if and only if $w \succ_f \emptyset$, and
- for all $w, w' \in W \setminus S$, $S \cup \{w\} \succ_f S \cup \{w'\}$ if and only if $w \succ_f w'$.

We will write $S \succeq_f S'$ to denote that either $S \succ_f S'$ or $S = S'$. Further, we will always consider the transitive closure of any responsive extension of $\succ_f$, which, in turn, induces a partial order over the set of all subsets of workers.

We will now define two subdomains of responsive preferences that will be of interest to us: *strongly monotone* and *lexicographic.*

Strongly monotone preferences. A firm is said to have *strongly monotone* preferences [28] if its preferences are responsive and it prefers cardinality-wise larger subsets of workers. That is, for any pair of acceptable subsets of workers S, T such that $|S| > |T|$, it holds that $S \succ_f T$.

Lexicographic preferences. A firm f is said to have *lexicographic* preferences if it prefers any subset of workers containing its favorite worker over any subset not containing it, subject to which, it prefers any subset containing its second-favorite worker over any subset not containing it, and so on. Formally, given a linear order $\succ_f$ over the set $W \cup \{\emptyset\}$ and any pair of distinct acceptable subsets of workers S and T, we have $S \succ_f T$ if and only if the favorite worker of firm f (as per $\succ_f$) in the set difference of S and T (i.e., $S \setminus T \cup T \setminus S$) lies in S. Observe that lexicographic preferences are responsive.

For many-to-one instances with two workers (i.e., $|W| = 2$) that are both acceptable to a firm, lexicographic and strongly monotone preferences coincide. However, for instances with three or more workers, strongly monotone preferences are not lexicographic and lexicographic preferences are not strongly monotone.[4]

Many-to-one matching. Given an instance $\mathcal{I} = \langle F, W, C, \succ \rangle$, a many-to-one matching for $\mathcal{I}$ is specified by a function $\mu : F \cup W \rightarrow 2^{F \cup W}$ such that:

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

A firm f with capacity c_f is said to be *saturated* under the matching μ if $|\mu(f)| = c_f$; otherwise, it is said to be *unsaturated.*

[4]This can be easily seen by considering a firm with preference over singletons as $w_1 \succ w_2 \succ w_3 \succ \cdots$. A firm with lexicographic preferences will prefer $\{w_1\}$ over $\{w_2, w_3\}$. On the other hand, under strongly monotone preferences, the firm will prefer $\{w_2, w_3\}$ over $\{w_1\}$. Hence, lexicographic and strongly monotone preferences do not coincide when there are three or more workers.

For simplicity, we will use the term *matching* in place of 'many-to-one matching' whenever it is clear from context. We will explicitly use the qualifiers 'one-to-one' and 'many-to-one' when the distinction between the two notions is relevant to the context.

Stability. A many-to-one matching μ is said to be

- *blocked by a firm f* if there is some worker $w \in \mu(f)$ such that $\emptyset \succ_f \{w\}$. That is, firm f prefers to keep a seat vacant rather than offer it to worker w.
- *blocked by a worker w* if $\emptyset \succ_w \mu(w)$. That is, worker w prefers being unmatched over being matched with firm $\mu(w)$.
- *blocked by a worker-firm pair (w, f)* if worker w prefers being matched with firm f over its current outcome under μ, and, simultaneously, firm f prefers being matched with worker w along with a subset of the workers in $\mu(f)$ over being matched with the set $\mu(f)$. That is, $f \succ_w \mu(w)$ and there exists a subset $S \subseteq \mu(f)$ such that $S \cup \{w\} \succ_f \mu(f)$ and $|S \cup \{w\}| \leq c_f$.[5]
- *stable* if it is not blocked by any worker, any firm, and any worker-firm pair.

The set of stable matchings for an instance $\mathcal{I}$ is denoted by $\mathcal{S}_{\mathcal{I}}$. Note that the above definition of stability assumes responsive preferences. A more general definition of stability in terms of choice sets can be found in [39].

Firm and worker optimal stable matchings. It is known that given any many-to-one matching instance, there always exists a *firm-optimal* (respectively, *worker-optimal*) stable matching that is weakly preferred by all firms (respectively, all workers) over any other stable matching. This result, due to Roth [33], is recalled in Proposition 2.1 below. We will write FOSM and WOSM to denote the firm-optimal and worker-optimal stable matching, respectively.

Proposition 2.1 (Firm-optimal and worker-optimal stable matchings [33]). *Given any instance $\mathcal{I}$, there exist (not necessarily distinct) stable matchings $\mu_F, \mu_W \in \mathcal{S}_{\mathcal{I}}$ such that for every stable matching $\mu \in \mathcal{S}_{\mathcal{I}}$, $\mu_F(f) \succeq_f \mu(f) \succeq_f \mu_W(f)$ for every firm $f \in F$ and $\mu_W(w) \succeq_w \mu(w) \succeq_w \mu_F(w)$ for every worker $w \in W$.*

Worker-proposing and firm-proposing algorithms. Two well-known algorithms for finding stable matchings are the worker-proposing and firm-proposing deferred acceptance algorithms, denoted by WPDA and FPDA, respectively. The WPDA algorithm proceeds in rounds, with each round consisting of a *proposal* phase followed by a *rejection* phase. In the proposal phase, every unmatched worker proposes to its favorite acceptable firm that hasn't rejected it yet. Subsequently, in the rejection phase, each firm f tentatively accepts its favorite c_f proposals and rejects the rest. The algorithm continues until no further proposals can be made.

Under the FPDA algorithm, firms make proposals and workers do the rejections. Each firm makes (possibly) multiple proposals in each round according to its ranking over individual workers. Each worker tentatively accepts its favorite proposal and rejects the rest. Roth [33] showed that the WPDA and FPDA algorithms return the worker-optimal and firm-optimal stable matchings, respectively.

[5]One might ask about blocking coalitions, wherein a set of workers and firms together block a given matching. It is known that if a coalition of workers and firms blocks a matching, then so does some worker-firm pair [37, Theorem 3.3].

Rural hospitals theorem. The *rural hospitals theorem* is a well-known result which states that, for any fixed firm f, the *number* of workers matched with f is the same in *every* stable matching [33]. Furthermore, if f is unsaturated in any stable matching, then it is matched with the same *set* of workers in *every* stable matching [35].

Proposition 2.2 (Rural hospitals theorem [33, 35]). *Given any instance $\mathcal{I}$, any firm f, and any pair of stable matchings $\mu, \mu' \in \mathcal{S}_\mathcal{I}$, we have that $|\mu(f)| = |\mu'(f)|$. Furthermore, if $|\mu(f)| < c_f$ for some stable matching $\mu \in \mathcal{S}_\mathcal{I}$, then $\mu(f) = \mu'(f)$ for every other stable matching $\mu' \in \mathcal{S}_\mathcal{I}$.*

Canonical one-to-one instance. Given a many-to-one instance $\mathcal{I} = \langle F, W, C, \succ \rangle$ with responsive preferences, there exists an associated one-to-one instance $\mathcal{I}' = \langle P, Q, \succ' \rangle$ obtained by creating c_f men for each firm f and one woman for each worker. Each man's preferences for the women mirror the corresponding firm's preferences for the corresponding workers. Each woman prefers all men corresponding to a more preferred firm over all men corresponding to any less preferred firm (in accordance with the corresponding worker's preferences). For any fixed firm, all women prefer the man corresponding to its first copy over the man representing its second copy, and so on. Any stable matching in the one-to-one instance $\mathcal{I}'$ maps to a unique stable matching in the many-to-one instance $\mathcal{I}$, obtained by "compressing" the former matching in a natural way (see Example 5 in [17]).

Proposition 2.3 (Canonical instance [16]). *Given any many-to-one instance $\mathcal{I} = \langle F, W, C, \succ \rangle$, there exists a one-to-one instance $\mathcal{I}' = \langle P, Q, \succ' \rangle$ such that there is a bijection between the stable matchings of $\mathcal{I}$ and $\mathcal{I}'$. Furthermore, the instance $\mathcal{I}'$ can be constructed in polynomial time.*

3 HOW DOES CAPACITY MODIFICATION AFFECT WORKERS AND FIRMS?

In this section, we study how changing the capacity of a firm can affect the outcomes of the firms and the workers. Specifically, we consider the worker-proposing and firm-proposing algorithms (WPDA and FPDA) and ask if a firm can improve/worsen when a unit capacity is added to it. Similarly, we will ask whether all workers can improve or if some worker can worsen when a firm's capacity is increased. Table 1 summarizes these trends.

The trends for capacity *decrease* by a firm can be readily inferred from Table 1. In particular, if increasing capacity can improve the firm's outcome, then going back from the new to the old instance implies that decreasing its capacity makes it worse off.

One might intuitively expect that a firm should improve upon increasing its capacity, as it can now be matched with a strict superset of workers. Similarly, it is natural to think that increase in a firm's capacity can also make some workers better off because an extra seat at a more preferable firm can allow some worker to switch to that firm, opening up the space for some other interested worker and so on, thus initiating a chain of improvements. Example 3.1 confirms this intuition on an instance where the workers' preferences are identical, also known as the *master list* setting.

Example 3.1 (All workers can improve). Consider an instance $\mathcal{I}$ with two firms f_1, f_2 and two workers w_1, w_2. The firm f_1 initially has zero capacity, while the firm f_2 has capacity 1 (i.e., $c_1 = 0$ and $c_2 = 1$). Both workers have the preference $f_1 \succ f_2 \succ \emptyset$, and both firms have the preference $w_1 \succ w_2 \succ \emptyset$. The unique stable matching for this instance is $\mu_1 = \{(w_1, f_2)\}$.

Now consider a new instance $\mathcal{I}'$ obtained by adding unit capacity to firm f_1 (i.e., $c_1' = 1$). The instance $\mathcal{I}'$ has a unique stable matching $\mu_2 = \{(w_1, f_1), (w_2, f_2)\}$. Observe that both workers w_1, w_2 as well as the firm f_1 that increased its capacity are better off under the new matching μ_2. Furthermore, as there is only one stable matching, the said trend holds under both FPDA and WPDA algorithms. Also note that the two sets of stable matchings are disjoint. Thus, no matching is simultaneously stable for both old and new instances. □

	WPDA	FPDA
Can the firm improve?	Yes [Ex 3.1]	Yes [Ex 3.1]
Can the firm worsen?	Yes [Ex 3.2], [39]	Yes [Ex 3.2], [39]
Can all workers improve?	Yes [Ex 3.1]	Yes [Ex 3.1]
Can some worker worsen?	No [Cor. 3.6], [16, 37]	No [Cor. 3.6], [16, 37]

Table 1: The effect of one firm increasing its capacity by 1 on itself and the workers, under the worker-proposing (WPDA) and firm-proposing (FPDA) algorithms.

Somewhat surprisingly, it turns out that increasing capacity can also *worsen* a firm. This observation follows from the construction of Sönmez [39], who showed that any stable matching algorithm is vulnerable to manipulation via underreporting of capacity by some firm. We recall Sönmez's construction in Example 3.2 below.

Intuitively, when workers propose under the WPDA algorithm, a firm can worsen upon capacity increase (equivalently, improve upon capacity decrease) because of the following reason: By having *fewer* seats, and thus by being *more* selective, the firm can initiate rejection chains which may prompt more preferable workers to propose to it. On the other hand, by adding an extra seat, a firm may be forced to accept a suboptimal set of workers. This is precisely what drives the manipulation in Example 3.2.

A similar reasoning works when the firms propose under the FPDA algorithm: Due to extra seats, a firm may be *forced* to make additional proposals to less-preferred workers, thus kicking off rejection chains that prompt other firms to take away its more preferred workers. Again, this phenomenon is at play in Example 3.2.

Example 3.2 (Increasing capacity can worsen a firm [39]). Consider an instance $\mathcal{I}$ with two firms f_1, f_2 and three workers w_1, w_2, w_3. The workers' preferences are given by

$$w_1 : f_2 \succ f_1 \succ \emptyset \qquad w_2, w_3 : f_1 \succ f_2 \succ \emptyset$$

The firms have lexicographic preferences given by

$$f_1 : \{w_1, w_2, w_3\} \succ \{w_1, w_2\} \succ \{w_1, w_3\} \succ \{w_1\} \succ \{w_2, w_3\} \succ \{w_2\} \succ \{w_3\} \succ \emptyset$$

$$f_2 : \{w_1, w_2, w_3\} \succ \{w_2, w_3\} \succ \{w_1, w_3\} \succ \{w_3\} \succ \{w_1, w_2\} \succ \{w_2\} \succ \{w_1\} \succ \emptyset$$

Initially, each firm has unit capacity, i.e., $c_1 = c_2 = 1$. In this case, there is a unique stable matching, namely

$$\mu_1 = \{(w_1, f_1), (w_3, f_2)\}.$$

Now consider a new instance $\mathcal{I}'$ derived from the instance $\mathcal{I}$ by increasing the capacity of firm f_1 by 1 (i.e., $c_1' = 2$ and $c_2' = 1$). The stable matchings for the instance $\mathcal{I}'$ are

$$\mu_2 = \{(\{w_1, w_2\}, f_1), (w_3, f_2)\} \text{ and}$$
$$\mu_3 = \{(\{w_2, w_3\}, f_1), (w_1, f_2)\}.$$

Here, the firm-optimal stable matching (FOSM) is μ_2 and the worker-optimal stable matching (WOSM) is μ_3.

Finally, consider another instance $\mathcal{I}''$ derived from $\mathcal{I}'$ by increasing the capacity of firm f_2 by 1 (i.e., $c_1'' = 2$ and $c_2'' = 2$). The unique stable matching for the instance $\mathcal{I}''$ is μ_3.

By virtue of being the unique stable matching, the matching μ_1 is FOSM and WOSM for the instance $\mathcal{I}$, and the matching μ_3 is FOSM and WOSM for the instance $\mathcal{I}''$. Observe that firm f_1 prefers μ_1 over μ_3. Thus, under WPDA algorithm, the transition from $\mathcal{I}$ to $\mathcal{I}'$ exemplifies that a firm (namely, f_1) can worsen upon increasing its capacity. Similarly, the firm f_2 prefers μ_2 over μ_3. Thus, under FPDA algorithm, the transition from $\mathcal{I}'$ to $\mathcal{I}''$ exemplifies that a firm (namely, f_2) can worsen upon increasing its capacity. □

Note that Example 3.2 crucially uses the lexicographic preference structure; indeed, firm f_1 prefers being matched with the solitary worker $\{w_1\}$ over being assigned the pair $\{w_2, w_3\}$. One might ask whether the implication of Example 3.2 holds in the absence of the lexicographic assumption. Proposition 3.3, due to Konishi and Ünver [28], shows that under *strongly monotone* preferences and WPDA algorithm, a firm *cannot* worsen upon capacity increase.

Proposition 3.3 ([28]). *Let μ and μ' denote the worker-optimal stable matching before and after a firm f with strongly monotone preferences increases its capacity by 1. Then, $\mu'(f) \succeq_f \mu(f)$.*

The main idea in the proof of Proposition 3.3 is as follows: Under WPDA, it can be shown that if the number of workers matched with a firm f does not change upon capacity increase, then the *set* of workers matched with f also remains the same. (Notably, this observation does not require the preferences to be strongly monotone.) It can also be shown that the number of workers matched with firm f cannot decrease upon capacity increase. (Again, this observation does not require strong monotonicity.). Thus, in order for the firm's outcome to change, it must be matched with strictly more workers in the new matching. Strong monotonicity then implies that the firm must strictly prefer the new outcome.

In contrast to WPDA, a firm *can* worsen upon capacity increase under the FPDA algorithm even under strongly monotone preferences (Example 3.4).

Example 3.4 (Increasing capacity can worsen a firm under strongly monotone preferences [39]). Consider the following instance, with two workers w_1, w_2 and two firms f_1, f_2 with strongly monotone preferences:

$$w_1 : f_2 \succ f_1 \succ \emptyset \qquad f_1 : \{w_1, w_2\} \succ \{w_1\} \succ \{w_2\} \succ \emptyset$$
$$w_2 : f_1 \succ f_2 \succ \emptyset \qquad f_2 : \{w_1, w_2\} \succ \{w_2\} \succ \{w_1\} \succ \emptyset$$

Initially, each firm has unit capacity, i.e., $c_1 = c_2 = 1$. In this case, the firm-optimal stable matching is

$$\mu_1 = \{(w_1, f_1), (w_2, f_2)\}.$$

Upon increasing the capacity of firm f_2 to $c_2 = 2$ while keeping $c_1 = 1$, the firm-optimal stable matching of the new instance is

$$\mu_2 = \{(w_1, f_2), (w_2, f_1)\},$$

which is worse for firm f_2 compared to the old matching μ_1. □

Finally, we note that under both FPDA and WPDA algorithms, no worker can worsen when a firm increases its capacity. The reason is that increasing the capacity of a firm corresponds to "adding a man" in the corresponding canonical one-to-one instance. Due to the increased "competition" among the men, the outcomes of all women weakly improve (Proposition 3.5).

Proposition 3.5 ([16, 37]). *Given any one-to-one instance $\mathcal{I} = \langle P, Q, \succ \rangle$, let $\mathcal{I}' = \langle P \cup \{p\}, Q, \succ' \rangle$ be another one-to-one instance derived from $\mathcal{I}$ by adding the man p such that the new preferences $\succ'$ agree with the old preferences $\succ$ on P and Q. Let μ_P and μ_Q be the men-optimal and women-optimal stable matchings, respectively, for $\mathcal{I}$, and let μ_P' and μ_Q' denote the same for $\mathcal{I}'$. Then, for every woman $q \in Q$, we have $\mu_P'(q) \succeq_q' \mu_P(q)$ and $\mu_Q'(q) \succeq_q' \mu_Q(q)$.*

Using Proposition 3.5 on the canonical one-to-one instance, we obtain that increasing a firm's capacity can never worsen the outcome of any worker under either worker-optimal or firm-optimal stable matching.

Corollary 3.6. *Let μ_W and μ_W' denote the worker-optimal stable matching before and after a firm increases its capacity by 1, and let μ_F and μ_F' be the corresponding firm-optimal matchings. Then, for all workers $w \in W$, $\mu_W'(w) \succeq_w \mu_W(w)$ and $\mu_F'(w) \succeq_w \mu_F(w)$.*

4 COMPUTATIONAL RESULTS

In this section, we will study the algorithmic aspects of capacity modification. We will take the perspective of a central planner who can modify the capacities of the firms to achieve a certain objective. We will focus on two natural (and mutually incomparable) objectives: (1) *Match a pair* (f^*, w^*), where the goal is to determine if a fixed firm f^* and a fixed worker w^* can be matched under *some* stable matching in the modified instance, and (2) *stabilize a matching* μ^*, where the goal is to check if a given matching μ^* can be realized as a stable outcome of the modified instance. These objectives have previously been studied in the one-to-one stable matching problem motivated by *control* problems [9, 19].

We will assume that the central planner can modify the firms' capacities in one of the following two natural ways: (1) By *adding* capacity, wherein the firms can receive some extra seats (the distribution can be unequal), and (2) by *deleting* capacity, wherein some of the existing seats can be removed. Under both addition and deletion problems, we will assume that there is a *global budget* $\ell \in \mathbb{N}$ that specifies the maximum number of seats that can be added (or removed) in aggregate across all firms.

The two objectives (match the pair and stabilize) and two actions (add and delete) together give rise to four computational problems. One of these problems—adding capacity to match a pair—is formally defined below. The other problems are defined analogously.

	Match the pair (f^*, w^*)		**Stabilize the matching μ^***	
	Add Capacity	Delete Capacity	Add Capacity	Delete Capacity
Unbudgeted	Poly time [Theorem 4.1]	Poly time [Theorem 4 in [17]]	Poly time [Theorem 7 in [17]]	Poly time [Theorem 9 in [17]]
Budgeted	NP-hard [Theorem 4.2]	NP-hard [Theorem 5 in [17]]	Poly time [Theorem 6 in [17]]	Poly time [Theorem 8 in [17]]

Table 2: Summary of our computational results for adding and deleting capacity under two problems: matching a worker-firm pair (columns 2 and 3) and stabilizing a given matching (columns 4 and 5). The top row contains the results for the *unbudgeted* problem (when only the aggregate change in firms' capacities is constrained) while the bottom row corresponds to the *budgeted* problem (with additional constraints on individual firms). The missing theorems and proofs are in the full version [17].

Add Capacity To Match Pair	
Given:	An instance $\mathcal{I} = \langle F, W, C, \succ \rangle$, a worker-firm pair (w^*, f^*), and a global budget $\ell \in \mathbb{N} \cup \{0\}$.
Question:	Does there exist a capacity vector $\overline{C} \in (\mathbb{N} \cup \{0\})^n$ such that $\overline{C} \geq C$, $\lvert\overline{C} - C\rvert_1 \leq \ell$, and f^* and w^* are matched in some stable matching of the instance $\mathcal{I}' = \langle F, W, \overline{C}, \succ \rangle$?

The aforementioned problems can be naturally generalized by considering *individual* budgets for the firms. For example, in the add capacity problem, in addition to the global budget ℓ, we can also have an individual budget ℓ_f for each firm f specifying the maximum number of additional seats that can be given to firm f. We call this generalization the *budgeted* version, and use the term *unbudgeted* to refer to the problem with only global—but not individual—budget. Formally, the budgeted version of Add Capacity to Match Pair problem is defined as follows:

Budgeted Add Capacity To Match Pair	
Given:	An instance $\mathcal{I} = \langle F, W, C, \succ \rangle$, a worker-firm pair (w^*, f^*), a global budget $\ell \in \mathbb{N} \cup \{0\}$, and an individual budget $\ell_f \in \mathbb{N} \cup \{0\}$ for each firm f.
Question:	Does there exist a capacity vector $\overline{C} \in (\mathbb{N} \cup \{0\})^n$ such that $\overline{C} \geq C$, $\lvert\overline{C} - C\rvert_1 \leq \ell$, $\lvert\overline{c}_f - c_f\rvert \leq \ell_f$ for each firm f, and f^* and w^* are matched in some stable matching of the instance $\mathcal{I}' = \langle F, W, \overline{C}, \succ \rangle$?

The consideration of individual budgets results in eight computational problems overall. Table 2 summarizes our results on the computational complexity of these problems.

A special case of the budgeted/unbudgeted problems is when the global budget is zero, i.e., $\ell = 0$. In this case, the capacities of the firms cannot be changed, and the goal is simply to check whether a worker-firm pair (w^*, f^*) are matched in some stable matching for the original instance $\mathcal{I}$, or whether a given matching μ^* is stable for $\mathcal{I}$. The latter problem is straightforward. To solve the former problem, it is helpful to consider the canonical one-to-one instance of the given instance $\mathcal{I}$. For the one-to-one stable matching problem, a polynomial-time algorithm is known for listing all man-woman pairs that are matched in one or more stable matchings [20]. Using the bijection between the stable matchings of the two instances (Proposition 2.3), we obtain an algorithm to check if the worker w^* is matched with any copy of firm f^* in any stable matching.

Thus, the zero budget case can be efficiently solved for all of the aforementioned problems. In the remainder of the section, we will consider the case of global budgets.

Adding Capacity to Match A Pair: Unbudgeted

Let us start with the problem of adding capacity to match a worker-firm pair (w^*, f^*) in the unbudgeted setting, i.e., with global but without individual budgets.

In order to check whether the worker-firm pair (w^*, f^*) can be matched in some stable matching in the given instance $\mathcal{I}$ by adding capacity to the firms, our algorithm (see Algorithm 1 in [17]) considers a modified instance $\mathcal{I}'$ where w^* and f^* are already matched, and checks if it is possible to construct a stable matching of the remaining agents satisfying some additional conditions.

More concretely, the algorithm considers the set of firms DF (short for "distracting firms") that the worker w^* prefers more than the firm f^*, and the set of workers DW (short for "distracting workers") that the firm f^* prefers more than w^*. Note that once the worker w^* is matched with the firm f^*, the firms in DF are the only ones that it could potentially form a blocking pair with (due to responsive preferences). Similarly, the workers in DW are the only ones that can block with f^* due to the forced assignment of w^* to f^*.

The algorithm creates the modified instance $\mathcal{I}'$ by truncating the preference lists of the firms in DF (respectively, the workers in DW) by having them declare all workers ranked below w^* (respectively, all firms ranked below f^*) as unacceptable. The truncation step is motivated from the following observation: In the original instance $\mathcal{I}$, there is a stable matching that matches (w^*, f^*) after adding capacities to the firms if and only if there exists a stable matching in the truncated instance $\mathcal{I}'$ such that, after the added capacities, all firms in the set DF are *saturated* (and thus, matched with workers they prefer more than w^*), and all workers in the set DW are *matched* (and thus, matched either with f^* or with firms they prefer more than f^*).

The key observation in our proof is that the desired matching exists in the truncated instance $\mathcal{I}'$ after adding capacities to the firms if and only if there exists a stable matching in the instance $\mathcal{I}'$ when the *entire* capacity budget is given to the firm f^*. This observation readily gives a polynomial-time algorithm. We defer the detailed proof of this observation to the full version [17].

Theorem 4.1. Add Capacity To Match Pair *can be solved in polynomial time.*

Adding Capacity to Match A Pair: Budgeted

Next, we will consider a more general problem where, in addition to the global budget of ℓ seats, we are also given an individual budget ℓ_f for each firm f specifying the maximum number of seats that can be added to the firm f. The goal, as before, is to determine if, after adding capacities as per the given budgets, it is possible to match the pair (w^*, f^*) under some stable matching.

Note that our algorithm for the unbudgeted problem assigns the entire additional capacity to the firm f^*, which may no longer be feasible in the budgeted problem. It turns out that, unless $P = NP$, no polynomial-time algorithm can be developed for this problem.

Theorem 4.2. Budgeted Add Capacity To Match Pair *is* NP-hard.

To prove Theorem 4.2, we leverage a result of Boehmer et al. [9] on control problems in the one-to-one stable matching problem (which, as per our convention, involves a matching between men and women). Specifically, Boehmer et al. [9] study the problem of adding a set of at most ℓ agents (men or women) such that in the resulting instance, a fixed man-woman pair are matched under some stable matching.

Interestingly, the reduction of Boehmer et al. [9] holds even when only men (but not women) are required to be added. Due to this additional feature, we slightly redefine the problem of Boehmer et al. [9] and call it Constructive-Exists-Add-Men. The formal definition of this problem is as follows:

Constructive-Exists-Add-Men	
Given:	An instance $\mathcal{I} = \langle P_{orig}, Q, \succ \rangle$, a set of addable men P_{add} with the preference relation $\succ$ defined over the entire set of agents $P_{orig} \cup P_{add} \cup Q$, a man-woman pair (p^*, q^*) from the original set of agents, and a budget $\ell \in \mathbb{N} \cup \{0\}$.
Question:	Does there exist a set $\overline{P} \subseteq P_{add}$ such that $\lvert\overline{P}\rvert \leq \ell$ and (p^*, q^*) is part of at least one stable matching in $\langle P_{orig} \cup \overline{P}, Q, \succ \rangle$?

The result of Boehmer et al. [9] shows that Constructive-Exists-Add-Men is NP-hard. We now use their result to show NP-hardness for Budgeted Add Capacity To Match Pair using the following straightforward construction: For each man in the set P_{orig}, we create a firm with capacity 1 and individual budget $\ell_f = 0$, while for each man in the addable set P_{add}, we create a firm with capacity 0 and individual budget $\ell_f = 1$. Adding a seat to an individual firm corresponds to adding the associated man. The equivalence now follows.

5 CAPACITY MODIFICATION V/S PREFERENCE MANIPULATION

So far, we have discussed qualitative (Section 3) and computational (Section 4) aspects of capacity modification from the perspective of a *central planner*. We will now adopt the perspective of a *firm* and compare the different manipulation actions available to it. Specifically, we will consider *preference manipulation* (abbreviated as `Pref`), wherein a firm can misreport its preference list without changing its capacity, and compare it with the two capacity modification actions we have already seen, namely `Add` and `Delete` capacity, wherein the firm can increase or decrease its capacity without changing its preferences. These actions are formally defined below.

- `Pref`: Under this action, a firm can report any *permutation* of its acceptable workers without changing its capacity.[6] That is, if a firm f's true preference is $\succ_f$, then $\succ'_f$ is a valid preference manipulation if for any worker w, $w \succ_f \emptyset$ if and only if $w \succ'_f \emptyset$.
- `Add`/`Delete`: Under `Add` (respectively, `Delete`), the firm f strictly increases (respectively, decreases) its capacity c_f by an arbitrary amount without changing its preferences.

Our goal is to examine which mode of manipulation—`Pref`, `Add`, or `Delete`—is always/sometimes more beneficial for the firm compared to the others under the FPDA and WPDA algorithms.

On first glance, each manipulation action may seem to offer a distinctive ability to the firm: `Add` allows the firm to either tentatively accept more proposals (under WPDA) or make more proposals (under FPDA), thus facilitating larger-sized (and possibly more preferable) matches. `Delete`, on the other hand, can allow a firm to be more selective, which, as we have seen in Section 3, can be advantageous in certain situations. Finally, `Pref` can allow a firm to trigger specific rejection chains, resulting in a potentially better set of workers. Given the unique advantage of each manipulation action, a systematic comparison among them is well motivated.

We compare the manipulation actions under two algorithms, WPDA and FPDA, and focus on a fixed firm f. An action X is said to *outperform* action Y (where $X, Y \in \{\texttt{Pref}, \texttt{Add}, \texttt{Delete}\}$) if there exists an instance such that the outcome for firm f when it performs X is strictly more preferable to it than that under Y.

An important insight from our analysis is that the usefulness of a manipulation action depends on a threshold on the firm's capacity which we call its *peak*. For fixed preferences of all agents and fixed capacities of the other firms, the peak of firm f is the size of the largest set of workers matched to f under any stable matching when f is free to choose its capacity $c_f \in \mathbb{N}$.

Formally, given an instance $\mathcal{I} = \langle F, W, C, \succ \rangle$, a firm $f \in F$ and any $b \in \mathbb{N}$, let $\mathcal{I}^b = \langle F, W, (C_{-f}, b), \succ \rangle$ denote the instance derived from $\mathcal{I}$ where the capacity of firm f is changed from c_f to b (and no other changes are made); here, C_{-f} denotes the capacities of firms other than f. Recall that the set of stable matchings for an instance $\mathcal{I}$ is denoted by $\mathcal{S}_{\mathcal{I}}$. The *peak* p_f for firm f is defined as the size of the largest set of workers f is matched with under any stable matching in the instance $\mathcal{I}^b$ for an arbitrary choice of b, i.e.,

$$p_f(\mathcal{I}) \coloneqq \max_{b \in \mathbb{N}, \mu \in \mathcal{S}_{\mathcal{I}^b}} |\mu(f)|.$$

Observe that when a firm's capacity is above its peak (i.e., $c_f > p_f$), it must necessarily be unsaturated in any stable matching. Similarly, by the rural hospitals theorem (Proposition 2.2), it follows that peak is the maximum number of proposals a firm receives under WPDA for an arbitrarily chosen capacity.

Figure 1 illustrates the comparison between the various manipulation actions under the FPDA and WPDA algorithms. Observe that in each of the three regimes in Figure 1—*below peak* (i.e., $c_f < p_f$),

[6] Manipulation via permutation has been studied by several works in the stable matching literature [18, 24, 25, 38, 41, 42].

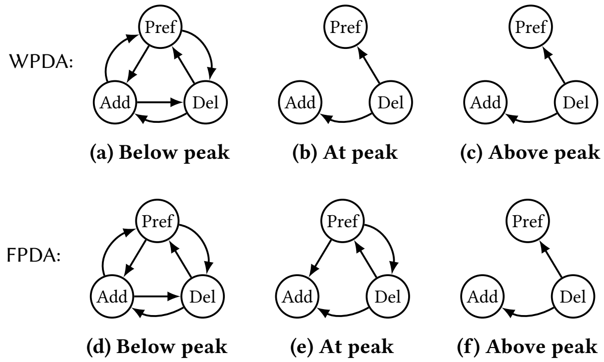

Figure 1: Manipulation trends for the WPDA **(top) and** FPDA **(bottom) algorithm in the below peak/at peak/above peak regimes. An arrow from action X to action Y denotes the existence of an instance where X is strictly more beneficial for the firm than Y. Each missing arrow from X to Y denotes that there is (provably) no instance where X is more beneficial than Y.**

at peak (i.e., $c_f = p_f$), and *above peak* (i.e., $c_f > p_f$)—there exist scenarios where `Delete` is strictly more beneficial than `Add` (and similarly, more beneficial than `Pref`). In fact, `Delete` is the only manipulation that can be beneficial above peak. The `Add` operation is only beneficial to a firm if its capacity is below peak irrespective of the matching algorithm. By contrast, `Pref` is beneficial to a firm at peak under FPDA but is unhelpful under WPDA.

In the rest of this section, we will discuss the comparison between `Delete` and `Pref` under the WPDA algorithm. In the full version [17], we discuss the other comparisons as well as the manipulation trends for strongly monotone preferences.

`Delete` vs `Pref`

Below Peak. When the capacity of a firm is below its peak (i.e., $c_f < p_f$), there exists an instance where `Pref` can outperform `Delete` (as well as `Add`) under WPDA. We defer the example of `Delete` outperforming `Pref` under the WPDA algorithm to [17].

Example 5.1 (`Pref` *outperforms* `Delete` *and* `Add` *under* WPDA*).* Consider an instance $\mathcal{I}$ with three firms f_1, f_2, f_3 and four workers w_1, w_2, w_3, w_4. The firms have unit capacities (i.e., $c_1 = c_2 = c_3 = 1$) and have lexicographic preferences given by

$$w_1 : f_2 \succ f_1 \succ f_3 \succ \emptyset \qquad f_1 : w_4 \succ w_1 \succ w_2 \succ w_3$$
$$w_2, w_3 : f_1 \succ f_2 \succ f_3 \succ \emptyset \qquad f_2 : w_3 \succ w_2 \succ w_1 \succ w_4$$
$$w_4 : f_3 \succ f_1 \succ f_2 \succ \emptyset \qquad f_3 : w_1 \succ w_4 \succ w_2 \succ w_3$$

Under the WPDA algorithm, firm f_1 is matched with $\{w_1\}$. If f_1 uses `Add` by switching to any capacity $c_1 \geq 2$, its WPDA match is the set $\{w_2, w_3\}$. It is easy to verify that the peak for firm f_1 is $p_f(\mathcal{I}) = 2$. Thus, under $\mathcal{I}$, the capacity of firm f_1 is below peak.

If f_1 uses `Pref` in the instance $\mathcal{I}$ by misreporting its preferences to be $w_4 \succ w_2 \succ w_3 \succ w_1$, then its WPDA match is $\{w_4\}$, which is more preferable for f_1 (according to its true preferences) than its match under `Add`. On the other hand, using `Delete` in the instance $\mathcal{I}$ (by reducing the capacity to $c_1 = 0$) is the worst outcome for f_1 as it is left unmatched. □

At Peak. When the capacity of the firm is equal to the peak (i.e., $c_f = p_f$), `Pref` becomes unhelpful under WPDA. This is because in this case, the number of proposals received by the firm under the WPDA algorithm is equal to its capacity. Thus, regardless of which permutation of the acceptable workers it reports, the firm does not reject any worker and is therefore matched with the same set.

Above Peak. When the capacity of the firm is above its peak, `Pref` again turns out to be unhelpful under WPDA. To show this, we first make the following observation: For any above-peak instance, the set of workers matched with firm f in *any* stable matching is the same as its worker-optimal match in the at-peak instance.

Lemma 5.2. *Let $\mathcal{I} = \langle F, W, C, \succ \rangle$ be an instance with a firm f such that $c_f > p_f$. Let μ^* be the worker-optimal stable matching of the at-peak instance $\mathcal{I}^{p_f}$, and μ be any stable matching of any above-peak instance $\mathcal{I}^b$ (where $b > p_f$). Then, $\mu(f) = \mu^*(f)$.*

In particular, Lemma 5.2 shows that the set of proposals made to the firm f under WPDA algorithm stays the same at or above peak.

Aziz et al. [6] have shown that under WPDA algorithm, a necessary condition for beneficial preference manipulation by a firm is that it must be saturated and receive more proposals than its capacity. By combining Lemma 5.2 with the observation of Aziz et al. [6], we get that `Pref` is ineffective in the at-peak and above-peak regimes under the WPDA algorithm. In fact, using Lemma 5.2, we can make a similar inference for *any* stable matching algorithm.

Theorem 5.3. *Under any stable matching algorithm, a firm cannot improve via preference manipulation* (`Pref`) *if its capacity is strictly greater than its peak.*

The proof of Lemma 5.2 and Theorem 5.3 can be found in the full version [17].

6 CONCLUDING REMARKS

We studied capacity modification in the many-to-one stable matching problem from qualitative, computational, and strategic perspectives, and provided a comprehensive set of results. Going forward, it would be interesting to explore algorithms for capacity modification when *both* add and delete operations are allowed. It would also be relevant to study situations where a firm can simultaneously misreport its preferences and change its capacity [27]. Finally, experiments on synthetic or real-world data to evaluate the frequency of availability of various manipulation actions is another natural direction to explore [36].

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their helpful comments. We are also grateful to Manshu Khanna for pointing out the relevant work of Dur and Van der Linden [14]. RV acknowledges support from DST INSPIRE grant no. DST/INSPIRE/04/2020/000107 and SERB grant no. CRG/2022/002621. Part of this work was done as the SURA (Summer Undergraduate Research Award) project of SG and SS during May-July 2023.

This material is based upon work supported by the National Science Foundation under Grant No. DMS-1928930 and by the Alfred P. Sloan Foundation under grant G-2021-16778, while SN was in residence at the Simons Laufer Mathematical Sciences Institute (formerly MSRI) in Berkeley, California, during the Fall 2023 semester.

REFERENCES

[1] Atila Abdulkadiroğlu, Parag A Pathak, and Alvin E Roth. 2005. The New York City High School Match. *American Economic Review* 95, 2 (2005), 364–367.

[2] Atila Abdulkadiroğlu, Parag A Pathak, Alvin E Roth, and Tayfun Sönmez. 2005. The Boston Public School Match. *American Economic Review* 95, 2 (2005), 368–371.

[3] Atila Abdulkadiroğlu and Tayfun Sönmez. 2003. School Choice: A Mechanism Design Approach. *American Economic Review* 93, 3 (2003), 729–747.

[4] Narges Ahani, Tommy Andersson, Alessandro Martinello, Alexander Teytelboym, and Andrew C Trapp. 2021. Placement Optimization in Refugee Resettlement. *Operations Research* 69, 5 (2021), 1468–1486.

[5] Haris Aziz, Jiayin Chen, Serge Gaspers, and Zhaohong Sun. 2018. Stability and Pareto Optimality in Refugee Allocation Matchings. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. 964–972.

[6] Haris Aziz, Hans Georg Seedig, and Jana Karina von Wedel. 2015. On the Susceptibility of the Deferred Acceptance Algorithm. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*. 939–947.

[7] Federico Bobbio, Margarida Carvalho, Andrea Lodi, Ignacio Rios, and Alfredo Torrico. 2023. Capacity Planning in Stable Matching: An Application to School Choice. In *Proceedings of the 24th ACM Conference on Economics and Computation*. 295.

[8] Federico Bobbio, Margarida Carvalho, Andrea Lodi, and Alfredo Torrico. 2022. Capacity Variation in the Many-to-one Stable Matching. *arXiv preprint arXiv:2205.01302* (2022).

[9] Niclas Boehmer, Robert Bredereck, Klaus Heeger, and Rolf Niedermeier. 2021. Bribery and Control in Stable Marriage. *Journal of Artificial Intelligence Research* 71 (2021), 993–1048.

[10] Eric Budish, Gérard P Cachon, Judd B Kessler, and Abraham Othman. 2017. Course Match: A Large-Scale Implementation of Approximate Competitive Equilibrium from Equal Incomes for Combinatorial Allocation. *Operations Research* 65, 2 (2017), 314–336.

[11] Jiehua Chen and Gergely Csáji. 2023. Optimal Capacity Modification for Many-To-One Matching Problems. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*. 2880–2882.

[12] Shradha Chettri. [n.d.]. New courses in 19 colleges will increase 2,000 seats in DU. *The Hindustan Times* ([n. d.]). https://www.hindustantimes.com/delhi/new-courses-in-19-colleges-will-increase-2-000-seats-in-du/story-aKie2NTKKAB3qyJnciyNAK.html

[13] Lester E Dubins and David A Freedman. 1981. Machiavelli and the Gale-Shapley Algorithm. *The American Mathematical Monthly* 88, 7 (1981), 485–494.

[14] Umut Dur and Martin Van der Linden. 2021. Capacity Design in School Choice. *Available at SSRN 3898719* (2021).

[15] David Gale and Lloyd S Shapley. 1962. College Admissions and the Stability of Marriage. *The American Mathematical Monthly* 69, 1 (1962), 9–15.

[16] David Gale and Marilda Sotomayor. 1985. Some Remarks on the Stable Matching Problem. *Discrete Applied Mathematics* 11, 3 (1985), 223–232.

[17] Salil Gokhale, Shivika Narang, Samarth Singla, and Rohit Vaish. 2024. Capacity Modification in the Stable Matching Problem. *arXiv preprint arXiv:2402.04645* (2024).

[18] Sushmita Gupta, Kazuo Iwama, and Shuichi Miyazaki. 2016. Total Stability in Stable Matching Games. In *Proceedings of the 15th Scandinavian Symposium and Workshops on Algorithm Theory*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 23:1–23:12.

[19] Sushmita Gupta and Pallavi Jain. 2023. Manipulation With (Out) Money in Matching Market. *Available at SSRN 4289699* (2023).

[20] Dan Gusfield. 1987. Three Fast Algorithms for Four Problems in Stable Marriage. *SIAM J. Comput.* 16, 1 (1987), 111–128.

[21] Dan Gusfield and Robert W Irving. 1989. *The Stable Marriage Problem: Structure and Algorithms*. MIT press.

[22] Onur Kesten. 2012. On Two Kinds of Manipulation for School Choice Problems. *Economic Theory* 51 (2012), 677–693.

[23] 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.

[24] Hirotatsu Kobayashi and Tomomi Matsui. 2009. Successful Manipulation in Stable Marriage Model with Complete Preference Lists. *IEICE Transactions on Information and Systems* 92, 2 (2009), 116–119.

[25] Hirotatsu Kobayashi and Tomomi Matsui. 2010. Cheating Strategies for the Gale-Shapley Algorithm with Complete Preference Lists. *Algorithmica* 58, 1 (2010), 151–169.

[26] Fuhito Kojima. 2007. When Can Manipulations be Avoided in Two-Sided Matching Markets?–Maximal Domain Results. *The BE Journal of Theoretical Economics* 7, 1 (2007).

[27] Fuhito Kojima and Parag A Pathak. 2009. Incentives and Stability in Large Two-Sided Matching Markets. *American Economic Review* 99, 3 (2009), 608–627.

[28] Hideo Konishi and M Utku Ünver. 2006. Games of Capacity Manipulation in Hospital-Intern Markets. *Social Choice and Welfare* 27, 1 (2006), 3–24.

[29] Taro Kumano and Morimitsu Kurino. 2022. Quota Adjustment Process. (2022). https://ies.keio.ac.jp/en/publications/22210/

[30] David Manlove. 2013. *Algorithmics of Matching under Preferences*. Vol. 2. World Scientific.

[31] Thanh Nguyen and Rakesh Vohra. 2018. Near-Feasible Stable Matchings with Couples. *American Economic Review* 108, 11 (2018), 3154–3169.

[32] Alvin E Roth. 1982. The Economics of Matching: Stability and Incentives. *Mathematics of Operations Research* 7, 4 (1982), 617–628.

[33] Alvin E Roth. 1984. The Evolution of the Labor Market for Medical Interns and Residents: A Case Study in Game Theory. *Journal of Political Economy* 92, 6 (1984), 991–1016.

[34] Alvin E Roth. 1985. The College Admissions Problem is Not Equivalent to the Marriage Problem. *Journal of Economic Theory* 36, 2 (1985), 277–288.

[35] Alvin E Roth. 1986. On the Allocation of Residents to Rural Hospitals: A General Property of Two-Sided Matching Markets. *Econometrica* (1986), 425–427.

[36] Alvin E Roth and Elliott Peranson. 1999. The Redesign of the Matching Market for American Physicians: Some Engineering Aspects of Economic Design. *American Economic Review* 89, 4 (1999), 748–780.

[37] Alvin E Roth and Marilda A Oliveira Sotomayor. 1990. *Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis*. Cambridge University Press.

[38] Weiran Shen, Yuan Deng, and Pingzhong Tang. 2021. Coalitional Permutation Manipulations in the Gale-Shapley Algorithm. *Artificial Intelligence* 301 (2021), 103577.

[39] Tayfun Sönmez. 1997. Manipulation via Capacities in Two-Sided Matching Markets. *Journal of Economic Theory* 77, 1 (1997), 197–204.

[40] Tayfun Sönmez. 1999. Can Pre-Arranged Matches be Avoided in Two-Sided Matching markets? *Journal of Economic Theory* 86, 1 (1999), 148–156.

[41] Chung-Piaw Teo, Jay Sethuraman, and Wee-Peng Tan. 2001. Gale-Shapley Stable Marriage Problem Revisited: Strategic Issues and Applications. *Management Science* 47, 9 (2001), 1252–1267.

[42] Rohit Vaish and Dinesh Garg. 2017. Manipulating Gale-Shapley Algorithm: Preserving Stability and Remaining Inconspicuous. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. 437–443.