Capacity Modification for Many-To-One Matching Problems

Author:
Arghyadip Chakraborty

Supervisor:
Prof. Prajakta Nimbhorkar

A thesis submitted in fulfillment of the requirements for the degree of Masters in Computer Science

Department of Computer Science
Chennai Mathematical Institute

June 2024

ABSTRACT

We study the problem of capacity modification in many-to-one stable matching problems, where one side consists of workers and the other side consists of firms with capacity constraints. First, we study how the set of stable matchings changes when some seats are assigned to the firms. We examine whether firms and workers can improve or worsen upon changing the capacities under worker-proposing and firm-proposing deferred acceptable algorithms.

Second, we study how to optimally increase the capacities of the firms so as to obtain a stable and perfect matching. We consider two common optimality criteria, one aiming to minimize the sum of capacity increase of all schools (abbrv. as MinSum) and the other aiming to minimize the maximum capacity increase of any school (abbrv. as MinMax). We obtain a complete picture in terms of computational complexity and further investigate the parameterized complexity and approximability.

Finally, we finish by answering some questions that emerged while reviewing the first two parts. We consider scenarios with bound on capacity increase scale, varying costs for firms to increase capacity, and targeting only selected (of initially unmatched) workers to match.

CONTENTS

1 INTRODUCTION

Many-to-one matching with two-sided preferences has various real-world applications such as school choice. i. e. placement of students to schools, college and university admission, hospital/residents programs, refugee resettlement or fair allocation in healthcare. In these scenarios, we are usually given two disjoint set of agents, W and F, such that each agent has a preference list over some members of the other set, and each agent in F has a *capacity constraint* that limits the maximum number of agents on the other side it can be feasibly matched with. The goal is to find a good matching (or assignment) between W and F without violating the capacity constraints. To unify the terminology, we consider the worker-firm allotment problem, and call the agents in W the workers and the agents in F the firms.

As to what defines a good matching, the answer varies from application to application. The simplest concept being that of a *perfect* matching, which ensures that every worker is matched, achieving which can be crucial. The arguably most prominent and well-known concept however is that of *stable* [GS62; GI89] matching, which ensures that no two agents form a blocking pair, i. e. they do not prefer to be matched with each other over their assigned partners. Stability is a key desideratum and has been a standard criterion for many matching applications. Remarkably, for any given capacities, a stable matching of workers and firms always exists and can be computed using the celebrated *deferred-acceptance* or *Gale-Shapley* algorithm [GS62; Rot84].

While the stable matching problem assumes fixed capacities, it is common to have flexible capacities in practice, particularly in settings with variable demand or popularity such as in vaccine distribution or course allocation. Flexibility refers to allowing the addition or removal of seats to firms that are either undersubscribed or oversubscribed, respectively. 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 [Sön97], who showed that under any stable matching algorithm, there exists a scenario where some firm is better off when its capacity is reduced.

Our focus of study will be the impact of *capacity modification* on the set of stable matchings. Next, we will study how *capacity modification* be done optimally to obtain stable and perfect matchings. Finally, we will study three other scenarios, with new parameters and constraints, and see how our initial approaches can be extended to solve the new problems.

1.1 THESIS OUTLINE

The remainder of this thesis is organized as follows

Chapter 2 provides the preliminary background, definitions and fundamentals (like Gale-Shapley Algorithm and Rural Hospital's Theorem) for stable and many-to-one matching problems.

Chapter 3 reviews Gokhale et al.'s [Gok+24] work, focusing on the impact of *capacity modification*. We study how the set of stable matchings respond to it.

Chapter 4 reviews Chen and Csáji's [CC24] work, focusing on the optimal capacity modification needed to obtain a stable and perfect matching. We study the computational complexity, parameterized complexity and approximability of the problem.

Chapter 5 answers the questions that arose while reviewing the last two chapters. We study scenarios with bound on capacity increase scale, varying costs for firms to increase capacity, and targeting only selected (of initially unmatched) workers to match.

2 PRELIMINARIES

For any positive integer $r \in \mathbb{N}$, let $[r] := \{1, 2, ..., r\}$. Given two integer vectors of $\boldsymbol{x}, \boldsymbol{y}$ of dimention t, i.e. $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{Z}^t$:

- we write $\boldsymbol{x} \leq \boldsymbol{y}$ iff $\boldsymbol{x}[i] \leq \boldsymbol{y}[i]$ for all $i \in [t]$
- $\boldsymbol{x} + \boldsymbol{y}$ denotes the addition vector $\boldsymbol{z}$, i.e. $\boldsymbol{z}[i] = \boldsymbol{x}[i] + \boldsymbol{y}[i]$ for all $i \in [t]$
- $\boldsymbol{x} \odot \boldsymbol{y}$ denotes the element-wise product vector $\boldsymbol{z}$, i.e. $\boldsymbol{z}[i] = \boldsymbol{x}[i] \cdot \boldsymbol{y}[i]$ for all $i \in [t]$
- $\lfloor \boldsymbol{x} \rfloor$ denotes the element-wise floor vector, i.e. $\lfloor \boldsymbol{x} \rfloor[i] = \lfloor \boldsymbol{x}[i] \rfloor$ for all $i \in [t]$
- $|\boldsymbol{x}|_1$ denotes the L_1 norm of $\boldsymbol{x}$, i.e. $|\boldsymbol{x}|_1 = \sum_{i \in [t]} \boldsymbol{x}[i]$
- $|\boldsymbol{x}|_\infty$ denotes the L_∞ norm of $\boldsymbol{x}$, i.e. $|\boldsymbol{x}|_\infty = \max_{i \in [t]} \boldsymbol{x}[i]$

A *preference list* (or *priority order*) $\succ$ over a set A is a linear order over A. We say that a is preferred to b if $a \succ b$.

2.1 MANY-TO-ONE MATCHING PROBLEM

The MANY-TO-ONE Matching (MM) problem has as input:

- A set $W = \{w_1, w_2, ..., w_n\}$ of $n \in \mathbb{N}$ workers
- A set $F = \{f_1, f_2, ..., f_m\}$ of $m \in \mathbb{N}$ firms
- For each worker $w \in W$, a *preference list* $\succ_w$ over a subset of firms
- For each firm $f \in F$, a *priority order* $\succ_f$ over a subset of workers
- A capacity vector $\boldsymbol{q} \in \mathbb{N}^m$ which specifies the maximum number of workers allowed to be assigned to each firm

Thus an instance of the many-to-one matching problem is given by a tuple $\langle W, F, \succ, \boldsymbol{q} \rangle$, where $\succ = (\succ_x)_{x \in W \cup F}$. When all firms have unit capacities, i.e. $\boldsymbol{q} = 1^m$, the problem becomes a one-to-one matching problem. In that case, we will follow the man-woman terminology, 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 preferences.

Throughtout, 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 each worker $x \in W$ (resp. firm $x \in F$), let $A(x)$ denote the set of firm acceptable to worker x (resp. all workers in firm x's priority order). We assume no worker (resp. no firm) has an empty preference list (resp. priority order), since such agents can be ignored or removed from the problem instance without any effect. Note that we also assume that a worker w is *acceptable* to a firm f iff f is *acceptable* to w, otherwise such acceptances can be removed similarly. We can also model the acceptability relations with a bipartite graph $G = \langle W, F; E \rangle$, where E is the set of all pairs (w, f) such that w and f find each other acceptable.

2.2 RESPONSIVE PREFERENCES

The extension of a firm f's preference $\succ_f$ over subsets of workers is said to be *responsive* if for any subset $S \subseteq A(f)$ of workers:

- for all $w \in A(f) \setminus S, S \cup \{w\} \succ_f S$
- for all $w, w' \in A(f) \setminus S, S \cup \{w\} \succ_f S \cup \{w'\}$ iff $w \succ_f w'$
- $\succ_f$ is transitive

Thus, $\succ_f$ induces a partial order over the set of all subsets of acceptable workers. Throughout, we will assume that all firms have *responsive* preferences over subsets of workers. We will also define two subdomains of responsive preferences that are of interest to us:

Strongly Monotone A firm f has *strongly monotone* preferences if it prefers cardinality-wise larger subsets of workers, i. e. for any $S, T \subseteq A(f)$, $S \succ_f T$ if $|S| > |T|$

Lexicographic A firm f has *lexicographic* preferences if it prefers any subset of workers containing its favourite worker over any subset not containing it, subject to which, it prefers any subset containing its second-favourite worker over any subset not containing it, and so on. Formally, for any $S, T \subseteq A(f)$, $S \succ_f T$ if the most preferred worker (to $\succ_f$) in $S \Delta T$ is in S

2.3 MATCHING

Given a MM Instance $I = \langle W, F, \succ, \boldsymbol{q} \rangle$, a matching $\mu : W \rightharpoonup F$ is a (partial) map such that:

- each worker $w \in W$ is either unmatched, i. e. $\mu(w)$ is undefined, or matched to an acceptable firm, i. e. $\mu(w) \in A(w)$
- each firm f is matched with at most $\boldsymbol{q}[f]$ workers, i. e. $|\mu^{-1}(f)| \leq \boldsymbol{q}[f]$

Firms which are assigned less workers than their capacity are called *under-filled* or *under-subscribed*. A matching is said to be *perfect* is every worker is matched under it.

We say an worker $w \in W$ *weakly prefers* a matching μ to a matching σ if either $\mu(w) = \sigma(w)$ or $\mu(w) \succ_w \sigma(w)$. Similarly, a firm $f \in F$ *weakly prefers* a matching μ to a matching σ if either $\mu^{-1}(w) = \sigma^{-1}(w)$ or $\mu^{-1}(w) \succ_f \sigma^{-1}(w)$. Matching μ is called a *worker-optimal* (resp. *firm-optimal*) stable matching if every worker (resp. every firm) weakly prefers μ to all other stable matchings.

2.4 STABILITY & GALE-SHAPLEY ALGORITHM

A matching μ is said to be blocked by a worker-firm pair (w, f) if:

- $\mu(w) = \perp$ or $f \succ_w \mu(w)$
- $\exists S \subseteq \mu^{-1}(f)$ st. $S \cup \{w\} \succ_f \mu^{-1}(f)$ and $|S \cup \{w\}| \leq q[f]$

A matching is said to be *stable* if it is not blocked by any such worker-firm pairs. The set of stable matchings for a MM Instance I is noted by S_I.

The *Gale-Shapley* algorithm [GS62], also know as the *deferred-acceptance* algorithm, is the well-known method to find a stable matching. As the name suggests, one of the set is chosen to be proposing and each round an unmatched agent from the set *proposes* to their *most preferred* agent(s) on the other set. Agents from the other set accept their *most preferred* proposal(s) yet, and reject the others. Note that there cannot be more than $|E|$ proposals, thus the algorithm takes linear time in terms of input (E) size.

Depending on whether the workers propose (worker-proposing deferred-acceptance or **WPDA**) or the firms propose (firm-proposing deferred-acceptance or **FPDA**), the outcome is a worker-optimal or a firm-optimal stable matching respectively.

Proposition 1 (Worker-optimal and Firm-optimal stable matchings [Rot84]). *Given any instance I, there exist (no necessarily distinct) stable matchings $\mu_W, \mu_F \in S_I$ such that for every stable matching $\mu \in S_I, \mu_W(w) \succ_w \mu(w) \succ_w \mu_F(w)$ for every worker $w \in W$ and $\mu_F(f) \succ_f \mu(f) \succ_f \mu_W(f)$ for every firm $f \in F$.*

2.5 CANONICAL ONE-TO-ONE INSTANCE

Given a many-to-one instance $I = \langle W, F, \succ, \boldsymbol{q} \rangle$ with responsive preferences, there exists an associated one-to-one instance $I' = \langle P, Q, \succ \rangle$ obtained by creating $\boldsymbol{q}[f]$ men for each firm f and one woman for each worker w. 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 I' maps to a unique stable matching in the many-to-one instance I, obtained by "compressing" the former matching in a natural way.

Proposition 2 (Canonical one-to-one instance [GS85])**.** *Given any many-to-one instance $I = \langle W, F, \succ, \boldsymbol{q} \rangle$, there exists a one-to-one instance $I' = \langle P, Q, \succ \rangle$ such that there is a bijection between the stable matchings of I and I'. Furthermore, the instance I' can be constructed in polynomial time.*

2.6 RURAL HOSPITALS THEOREM

The *rural hospitals theorem* is a well-know result for many-to-one stable matchings. It states that, for any fixed firm f, the number of workers matched with f is the same in every stable matching [Rot84]. Furthermore, if f is *under-filled* in any stable matching, then it is matched with the same set of workers in every stable matching [Rot86].

Proposition 3 (Rural Hospitals theorem [Rot84; Rot86])**.** *Given any instance I, any firm f, and any pair of stable matchings $\mu, \mu' \in S_I$, we have that $|\mu^{-1}(f)| = |\mu'^{-1}(f)|$. Furthermore, if $|\mu^{-1}(f)| < q[f]$ for some stable matching $\mu \in S_I$, then $\mu^{-1}(f) = \mu'^{-1}(f)$ for every stable matching $\mu' \in S_I$*

Thus given a MM instance, all stable matchings match the same set of workers and firms. We denote the set of all assigned and unassigned workers in a stable matching by W_a and W_u respectively. We also use the following notations:

$\Delta_x = \lvert A(x) \rvert$	length of the preference list of agent $x \in W \cup F$
$\Delta_u = \max\limits_{w \in W_u} \{\Delta_w\}$	length of the longest preference list among unassigned workers
$\Delta_W = \max\limits_{w \in W} \{\Delta_w\}$	length of the longest preference list among all workers
$\Delta_F = \max\limits_{f \in F} \{\Delta_f\}$	length of the longest preference list among all firms

3 IMPACT OF CAPACITY MODIFICATION

In this chapter, we explore how changing the capacity of a firm can affect the outcome of the firms and the workers, thus the set of stable matchings. Specifically, we consider **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 workers can worsen when a firm's capacity is increased. The results are summarized below:

	WPDA	**FPDA**
Can the firm improve?	Yes [Example 1]	Yes [Example 1]
Can the firm worsen?	Yes [Example 2]	Yes [Example 2]
Can all workers improve?	Yes [Example 1]	Yes [Example 1]
Can some worker worsen?	No [Corollary 1]	No [Corollary 1]

Table 1: Effect of a firm's capacity increase on itself and the workers

Note that the impact of decreasing capacity of a firm can be readily inferred from Table 1. If increasing capacity can improve the firm's outcome, then descreasing its capacity is equivalent to going back from the new to the old instance, which make it worse off.

3.1 CAN FIRMS AND WORKERS IMPROVE?

Example 1. *Consider an instance I with three workers w_1, w_2, w_3 and two firms f_1, f_2 with preferences:*

$$w_1, w_2, w_3 : f_1 \succ f_2 \;\Big|\; f_1, f_2 : w_1 \succ w_2 \succ w_3$$

Both firms have unit capacity, i. e. $\boldsymbol{q}[f_1] = \boldsymbol{q}[f_2] = 1$, then I has a unique stable matching:

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

Now suppose I' is the instance obtained by adding unit capacity to firm f_1, i. e. $\boldsymbol{q}'[f_1] = 2$, then I' also has a unique stable matching

$$\mu_2 = \{(\{w_1, w_2\}, f_1), (w_3, f_2)\}$$

Observe that all workers w_1, w_2, w_3 as well as the firm f_1 that increased its capacity are better off under the new matching μ_2, i. e. they prefer μ_2 over μ_1. Furthermore, as μ_1 and μ_2 are the only stable matchings for their corresponding instances, the result holds under both **WPDA** and **FPDA** algorithms. Hence both the firm and all workers can improve on capacity increase.

3.2 CAN FIRMS WORSEN?

Example 2. *Consider an instance I with three workers w_1, w_2, w_3 and two firms f_1, f_2 with lexicographic preferences:*

$$\begin{array}{r|l} & f_1 : \{w_1, w_2, w_3\} \succ \{w_1, w_2\} \succ \{w_1, w_3\} \succ \\ w_1 : f_2 \succ f_1 & \quad \{w_1\} \succ \{w_2, w_3\} \succ \{w_2\} \succ \{w_3\} \\ w_2, w_3 : f_1 \succ f_2 & f_2 : \{w_1, w_2, w_3\} \succ \{w_2, w_3\} \succ \{w_1, w_3\} \succ \\ & \quad \{w_3\} \succ \{w_1, w_2\} \succ \{w_2\} \succ \{w_1\} \end{array}$$

Each firm has unit capacity, i. e. $\boldsymbol{q}[f_1] = \boldsymbol{q}[f_2] = 1$, then there is a unique stable matching for I:

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

Now consider a new instance I' derived from I by increasing the capacity of firm f_1 by 1, i. e. $\boldsymbol{q}'[f_1] = 2$, then the stable matchings for I' are:

$$\mu_2 : \{(\{w_1, w_2\}, f_1), (w_3, f_2)\}$$
$$\mu_3 : \{(\{w_2, w_3\}, f_1), (w_1, f_2)\}$$

where μ_2 and μ_3 are respectively the firm-optimal and worker-optimal stable matchings for I'.

Next consider a new instance I'' derived from I' by increasing the capacity of firm f_2 by 1, i. e. $\boldsymbol{q}''[f_2] = 2$, then μ_3 is the unique stable matching for I''.

Observe that firm f_1 prefers μ_1 (unique and thus worker-optimal stable matching for I) over μ_3 and firm f_2 prefers μ_2 over μ_3 (unique and thus firm-optimal stable matching for I''). Thus under both **WPDA** and **FPDA** algorithms, a firm (resp. f_1 and f_2) can worsen upon increasing its capacity.

Proposition 4 ([KÜ06]). *Let μ and μ' denote the worker-optimal stable matching before and after a firm f with strongly monotone preference increases its capacity by 1. Then $\mu'(f) \succeq_f \mu(f)$*

We can show that the number of workers matched with a firm f cannot decrease upon capacity increase. We can also show that if the number of workers matched with a firm f does not change, then set of set of workers matched with f also remains the same. Thus, for a firm's outcome to change, it must be matched with strictly more workers in the new matching, and under strong monotonicity it will strictly prefer the new outcome. Thus under **WPDA** algorithm, a firm with stronly monotone preferences cannot worsen upon increasing its capacity.

Example 3. *Consider an instance I with two workers w_1, w_2 and two firms f_1, f_2 with strongly monotone preferences:*

$$\begin{array}{l|l} w_1 : f_2 \succ f_1 & f_1 : \{w_1, w_2\} \succ \{w_1\} \succ \{w_2\} \\ w_2 : f_1 \succ f_2 & f_2 : \{w_1, w_2\} \succ \{w_2\} \succ \{w_1\} \end{array}$$

Each firm has unit capacity, i.e. $\boldsymbol{q}[f_1] = \boldsymbol{q}[f_2] = 1$, then the firm-optimal stable matching for I is:

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

Now consider a new instance I' derived from I by increasing the capacity of firm f_2 by 1, i.e. $\boldsymbol{q}'[f_2] = 2$, then the firm-optimal stable matching for I' is:

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

Observe that f_2 prefers μ_1 over μ_2. Thus under **FPDA** algorithm, thus a firm with strongly monotone preference (namely f_2) can worsen on increasing its capacity.

3.3 CAN WORKERS WORSEN?

Proposition 5 ([GS85; RS90]). *Given any one-to-one instance $I = \langle P, Q, \succ \rangle$, let $I' = \langle P \cup \{p\}, Q, \succ' \rangle$ be another one-to-one instance derived from I by adding the man p such that the new preferences $\succ'$ agree with the old preferences $\succ$ over P and Q. Let μ_P and μ_Q be the men-optimal and women-optimal stable matchings (respectively) for I and let μ'_P and μ'_Q denote the same for I'. Then, for every woman $q \in Q$, we have $\mu'_P(q) \succeq'_q \mu_P(q)$ and $\mu'_Q(q) \succeq'_q \mu_Q(q)$.*

Increasing the capacity of a firm is equivalent to "adding a man" in the corresponding canonical one-to-one instance (Section 2.5). Due to the increased "competition" among the men, the outcomes of all women (correspondingly the outcome of all worker) weakly improves (Proposition 5). Thus under both **WPDA** and **FPDA** algorithms, the outcome of any worker can never worsen.

Corollary 1. *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 stable matching. 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

OPTIMAL CAPACITY MODIFICATION

In this chapter, we explore how to optimally increase the capacities of the firms so that we can obstain a *stable* and *perfect* matching. We focus on two decision problems, namely MinSum Cap Stable and Perfect and MinMax Cap Stable and Perfect abbreviated as MinSumSP and MinMaxSP respectively.

4.1 MINSUM CAPACITIES

Input : *A MM Instance* $I = \langle W, F, \succ, \boldsymbol{q} \rangle$*, a capacity bound* $k^+ \in \mathbb{N}$

Question (MinSumSP) **:** *Is there a capacity increase vector* $\boldsymbol{r}$ *with* $|\boldsymbol{r}|_1 \leq k^+$ *st.* $I' = \langle W, F, \succ, \boldsymbol{q} + \boldsymbol{r} \rangle$ *admits a stable and perfect matching?*

We call a capacity increase vector $\boldsymbol{r}$ *feasible* if it results in a solution and *optimal* is $|\boldsymbol{r}|_1$ is minimum among all feasible vectors. We also denote the minimum $|\boldsymbol{r}|_1$, i. e. the optimal capacity increase value by OPT.

4.1.1 *Structural Properties*

Lemma 1. *Let* $I_1 = \langle W, F, \succ, \boldsymbol{q}_1 \rangle$ *and* $I_2 = \langle W, F, \succ, \boldsymbol{q}_2 \rangle$ *denote two MM-instances with the same set of workers and firms, and the same preferences and priority lists such that* $\boldsymbol{q}_1 \leq \boldsymbol{q}_2$*. Then the following hold.*

i. *Every worker weakly prefers the worker-optimal stable matching* μ_2 *in* I_2 *to the worker-optimal stable matching* μ_1 *in* I_1*.*

ii. *If a firm* f *is under-filled in* μ_1*, then* $\mu_2^{-1}(f) \subseteq \mu_1^{-1}(f)$

Proof. For (i), note that the weakly preferring relation is transitive. So it suffices to consider the case where only unit capacity is added to exactly one firm, the outcome of which we have already seen in Corollary 1 (also shown by Kominers [Kom20]).

For (ii), for the sake of contradiction, suppose there exists a worker $w \in \mu_2^{-1} \setminus \mu_1^{-1}(f)$, then by (i), $\mu_2(w) \succ_w \mu_1(w)$. But f is under-filled in μ_1, and thus it forms a blocking pair with w in μ_1, a contradiction. □

Algorithm 1 Create $I^*(\boldsymbol{v})$ and $\mu^*_{\boldsymbol{v}}$

Input: $\boldsymbol{v} \in \mathcal{V}$, a MM Instance I and the sets W_a, W_u **Output:** $I^*(\boldsymbol{v}), \mu^*_{\boldsymbol{v}}$
  Set $E = E' := (w, f) | w \in A(f) \wedge f \in A(w)$
  for each firm $f \in F$ **do**
    $n(f, \boldsymbol{v})$:= number of workers $\boldsymbol{v}$ assigns to f
    $\boldsymbol{q}'[f] := \max\{\boldsymbol{q}[f] - n(f, \boldsymbol{v}), 0\}$
  end for
  for $(w, f) \in E'$ with $w \in W_a$ **do**
    if there is a $w' \in W_u$ st. $f \succ_{w'} \boldsymbol{v}[w'] \wedge w' \succ_f w$ **then**
      Delete (w, f) from E'
    else if there is a $w' \in W_u$ st. $\boldsymbol{v}[w'] \succ_w f \wedge w \succ_{\boldsymbol{v}[w']} w'$ **then**
      Delete (w, f) from E'
    end if
  end for
  if there is a $w \in W_u$ with no remaining incident edges in E' **then**
    Return "No $I^*(\boldsymbol{v})$ instance"
  end if
  for $x \in W_a \cup F$ **do**
    Set $\succ'_x$ to be the restriction of $\succ_x$ to their remaining partners in $W_a \cup F$
  end for
  Set $I(\boldsymbol{v}) := \langle W_a, F, \succ', \boldsymbol{q}' \rangle$
  $\hat{\mu}_{\boldsymbol{v}}$:= worker-optimal stable matching of $I(\boldsymbol{v})$
  for $w \in W_a$ **do**
    $wo(w, \boldsymbol{v})$:= the worst remaining firm of w in $I(\boldsymbol{v})$
  end for
  $U(\boldsymbol{v})$:= unmatched workers in $\hat{\mu}_{\boldsymbol{v}}$
  $n_u(f, \boldsymbol{v}) := |\{w \in U(\boldsymbol{v}) \mid wo(w, \boldsymbol{v}) = f|$
  for for $w \in W$ **do**
    if $w \in W_u$ **then**
      $\mu^*_{\boldsymbol{v}}(w) := \boldsymbol{v}[w]$
    else if $w \in U(\boldsymbol{v})$ **then**
      $\mu^*_{\boldsymbol{v}}(w) := wo(w, \boldsymbol{v})$
    else
      $\mu^*_{\boldsymbol{v}}(w) := \hat{\mu}_{\boldsymbol{v}}(w)$
    end if
  end for
  for $f \in F$ **do**
    $\boldsymbol{q}^*[f] := \max\{n(f, \boldsymbol{v}) + |\hat{\mu}_{\boldsymbol{v}}(f)| + n_u(f, \boldsymbol{v}), \boldsymbol{q}[f]\}$
  end for
  Return $I^*(\boldsymbol{v}) := \langle W, F, \succ, \boldsymbol{q}^* \rangle, \mu^*_{\boldsymbol{v}}$

Given a MM Instance I, let $\hat{\mu}$ be the worker-optimal stable matching. Consider all possible vectors $\boldsymbol{v}$, whose coordinates are the unassigned workers and for each coordinate $w \in W_u$, the entry is an acceptable firm for the unassigned worker ($\boldsymbol{v}[w] \in A(w)$). Denote the set of such vectors by $\mathcal{V}$. Now, if we create $I^*(\boldsymbol{v})$ and $\mu^*_{\boldsymbol{v}}$ using Algorithm 1, we have the following lemma:

Lemma 2 ([CC24])**.** *Let $\mathcal{V}' \subseteq \mathcal{V}$ be the set of those vectors $\boldsymbol{v}$, such that $I^*(\boldsymbol{v})$ exists and $\mu^*_{\boldsymbol{v}}$ is stable in $I^*(v)$. Then, the sum of values of an optimal capacity increase vector for* MINSUMSP

$$OPT = \min_{v \in \mathcal{V}'} \left\{ \sum_{f \in F} (n(f, \boldsymbol{v}) + |\hat{\mu}_v(f)| + n_u(f, \boldsymbol{v}) - q[f])^+ \right\}$$

4.1.2 *Hardness Results*

Given any capacity increase vector $\boldsymbol{r}$, whether $I' = \langle W, F, \succ, \boldsymbol{q} + \boldsymbol{r} \rangle$ admits a stable and perfect matching can be checked in linear time using the *Gale-Shapley* algorithm. Hence, MINSUMSP is contained in NP.

Theorem 1. MINSUMSP *is NP-complete; hardness remains even if $q = 1^m$, $\Delta_W \leq 4$, $\Delta_u = 2$ and $\Delta_F = 3$. If $\Delta_u \leq 1$ or $\Delta_F \leq 2$ then* MINSUMSP *becomes polynomial-time solvable.*

Proof. For the polynomial results, consider $\Delta_u \leq 1$, then there is just one possible assignment vector for the unassigned workers, and by Lemma 2 the problem can be solved in polynomial time.

Next, assume that $\Delta_F \leq 2$. Since every firm has atleast one seat, it must be assigned atleast one worker in every initial stable matching as otherwise by Lemma 1 (ii) we can ignore such firms. Thus if an unassigned worker is assigned to a firm, it already has an initially assigned worker, and no other assigned or unassigned worker would have a justified envy. Hence, each assignment vector corresponds to a good capacity increase vector and we only need to check whether $k^+ \geq |W_{un}|$.

Now for the hardness result we have a reduction from the NP-complete VERTEX COVER problem [CC24] □

Theorem 2. MINSUMSP *does not have any constant-factor approximation algorithm unless P=NP. This holds even if the preference and priority lists are derived from a master list*

Proof. We have a reduction from the NP-hard SET COVER problem [CC24] □

Theorem 3. MINSUMSP *is $W[1]$-hard wrt. the capacity bound k^+*

Proof. We have a parameterized reduction from the MULTI-COLORED CLIQUE problem, which is $W[1]$-hard wrt. the solution size h [CC24] □

$$\begin{aligned}
\min \sum_{f\in F} r_f \quad & \text{subject to} \\
|W|\cdot \sum_{f'|f'\succeq_w f} x_{(w,f')} + \sum_{w'|w'\succ_f w} x_{(w',f)} \geq q[f]+r_f \quad & \forall (w,f)\in E \\
\sum_{f\in A(w)} x_{(w,f)} = 1 \quad & \forall w\in W \\
\sum_{w\in A(f)} x_{(w,f)} \leq q[f]+r_f \quad & \forall f\in F \\
x_{(w,f)} \in \{0,1\} \quad & \forall (w,f)\in E \\
r_f \in \mathbb{N} \quad & \forall f\in F
\end{aligned}$$

Table 2: IP formulation for MinSumSP

4.1.3 *Algorithmic Results*

First, Table 2 gives us an an Integer Programming formulation for MinSumSP.

Lemma 3 ([CC24]). *The optimal solution to the Integer Program in Table 2 gives an optimal solution for* MinSumSP

Next, based on Lemma 2, we have a simple greedy approximation algorithm.

Algorithm 2 $|W_u|$-approximation

Input: MM Instance I
  $\mu := \phi$
  $L := W_u$ = set of unmatched workers in the worker-optimal stable matching
  Delete the workers $w \in W_u$ from I
  while $L \neq \phi$ **do**
    Choose the next (at most) c workers in L, add them to I and define $\mathcal{V}$ as in Lemma 2
    for all $v \in \mathcal{V}$ **do**
      Compute $I^*(v)$ and μ^*_v using Algorithm 1 if it exists
    end for
    Let $v \in \mathcal{V}'$ be the vector where the smallest aggregate capacity increase is needed

    Update the capacities according to $I^*(v)$
    Set $I := I^*(v)$ and $\mu := \mu^*_v$
  end while
  Return μ

Theorem 4. MinSumSP *admits an $|W_u|$-approx. algorithm. Futhermore, it admits a polynomial time $\lceil |W_u/c| \rceil$-approx. algorithm for any constant c.*

Proof. Fix a constant c, then in each iteration of the while loop, we choose at most c unassigned workers. Thus $|\mathcal{V}| \leq \Delta_u^c$ and the running time is $\Delta_u^c \cdot O(|E|)$. Now there are at most $\lceil |W_u|/c \rceil$ iterations of the while loop, and in each iteration the additional number of seats required is at most OPT. Hence, the total capacity increase of the algorithm is at most $OPT \cdot \lceil |W_u|/c \rceil$ and we have a polynomial time $\lceil |W_u|/c \rceil$-approx. algorithm. □

Checking all possible assignment vectors gives up the following result:

Theorem 5. MINSUMSP *can be solved in* $\Delta_u^{|W_u|} \cdot (n+m)^{O(1)}$ *time and hence is FPT wrt.* $(|W_u|, \Delta_u)$.

Proof. From Lemma 2, we know that to find the optimal capacity increase vector, it is enough to iterate over $\mathcal{V}$, compute μ_v^* if it exists and choose the best among the matchings that are stable. For each $v \in \mathcal{V}$, $n_u(v)$ can be computed in $O(|E|)$ time and $|\mathcal{V}| \leq \Delta_u^{|W_u|}$. Hence, we can iterate through all $v \in \mathcal{V}$ and find the optimal solution in time $O(\Delta_u^{|W_u|} \cdot |E|)$. □

4.2 MINMAX CAPACITIES

Input : *A MM Instance* $I = \langle W, F, \succ, \boldsymbol{q} \rangle$*, a capacity bound* $k^{max} \in \mathbb{N}$

Question (MINMAXSP) **:** *Is there a capacity increase vector* $\boldsymbol{r}$ *with* $|\boldsymbol{r}|_\infty \leq k^{max}$ *st.* $I' = \langle W, F, \succ, \boldsymbol{q} + \boldsymbol{r} \rangle$ *admits a stable and perfect matching?*

Algorithm 3 Algorithm for MINMAXSP

Input: MM Instance I
  $r[f] := 0$ for all $f \in F$
  $\mu :=$ worker-optimal stable matching
  while μ does not match all workers **do**
    $r[f] = r[f] + 1$
    $\mu :=$ worker-optimal stable matching with capacities $q + r$
  end while
  Return μ

Theorem 6. MINMAXSP *can be solved, and the corresponding student-optimal stable and perfect matching can be found, in polynomial time.*

Proof. From Lemma 1 we know that increasing a firm's capacity weakly improves every worker's situation. Assume r is the optimal capacity increase vector, with $|r|_\infty = k^{max}$, then if we have $r[f] = k^{max}$ for all $f \in F$, then the worker-optimal stable matching should match all workers and is best for the workers for any capacity increase vector with $|r|_\infty \leq k^{max}$. Hence, the algorithm finds the optimal k^{max}, and a corresponding worker-optimal stable matching. □

5 OUR CONTRIBUTION

In this chapter, we investigate some questions that emerged while reviewing the last two chapters. We will see how the existing ways can be extended to tackle more realistic challenges, which take more parameters and constraints into account.

5.1 QUESTION 1

What if there is a bound on the capacity increase scale, say κ% of the initial value? Formally:

Input : *A MM Instance $I = \langle W, F, \succ, \boldsymbol{q} \rangle$, a capacity increase scale bound $\kappa \in \mathbb{R}$*

Question : *Is there a capacity increase scale α with $\alpha \leq \kappa$ st. $I' = \langle W, F, \succ, \boldsymbol{q} + \lfloor \alpha \boldsymbol{q} \rfloor \rangle$ admits a stable and perfect matching?*

Algorithm 4 Binary search on $[0, \kappa]$

```
Input: MM Instance I, capacity increase scale bound κ, error bound ϵ
  Set low := 0, high := κ, α := 0
  while high − low > ϵ do
    α := (low+high)/2
    I' := ⟨W, F, ≻, q + ⌊αq⌋⟩
    if I' admits a stable and perfect matching then
      low = α
    else
      high = α
    end if
  end while
  Return α
```

The simplest approach is to divide $[0, \kappa]$ into (say) n partitions, and perform a linear search by picking a point in each partition. We can get more accurate values of α by increasing the number of partitions, since the difference between each point gets smaller. A better approach is to use a binary search on $[0, \kappa]$ with an error bound $\epsilon > 0$ (Algorithm 4) or a certain number of rounds n. Since checking whether I' admits a stable and perfect matching can be done in linear time using the *Gale-Shapley* algorithm, all of the above approaches run in polynomial time.

5.2 QUESTION 2

What is each firm has different cost associated to its unit capacity increase? Formally:

5.2.1 MinSum Cost-Based Capacities

Input : *A MM Instance* $I = \langle W, F, \succ, \boldsymbol{q} \rangle$*, a cost vector* $\boldsymbol{c} \in \mathbb{N}^m$*, a capacity bound* $k^+ \in \mathbb{N}$

Question (MinSumCSP) **:** *Is there a capacity increase vector* $\boldsymbol{r}$ *with* $|\boldsymbol{c} \odot \boldsymbol{r}|_1 \leq k^+$ *st.* $I' = \langle W, F, \succ, \boldsymbol{q} + \boldsymbol{r} \rangle$ *admits a stable and perfect matching?*

If we set $\boldsymbol{c} = 1^m$, then the problem reduces to MinSumSP, which we know is NP-complete (Theorem 1). Thus MinSumCSP is NP-complete. So we try to extend the algorithmic results from MinSumSP to account for costs.

The Integer Program can be simply extended by changing the optimize function to $\min \sum_{f \in F} \boldsymbol{c}[f] \cdot r_f$ while the other constraints remain the same.

$$|W| \cdot \sum_{f' | f' \succeq_w f} x_{(w,f')} + \sum_{w' | w' \succ_f w} x_{(w',f)} \geq q[f] + r_f \;\forall (w,f) \in E$$

The first constraint of the IP Table 2 (above) is to ensure that:

- either the worker w is assigned to some more preferred firm f', i. e.

$$\sum_{f' | f' \succeq_w f} x_{(w,f')} = 1$$

- or the firm f is already full with workers more preferred than w, i. e.

$$\sum_{w' | w' \succ_f w} x_{(w',f)} \geq q[f] + r_f$$

$|W|$ is chosen to scale the first term greater than $q[f] + r_f$ and can be conviniently replace by either smaller parameters Δ_F or Δ_f.

Algorithm 1 can be used without any change, by accounting for the cost in Lemma 2 as follows:

$$OPT = \min_{v \in \mathcal{V}'} \left\{ \sum_{f \in F} \boldsymbol{c}[f] \cdot (n(f,v) + |\hat{\mu}_v(f)| + n_u(f,v) - q[f])^+ \right\}$$

This also implies that the approximation and FPT results naturally follow, since they directly use Lemma 2.

5.2.2 MinMax Cost-Based Capacities

Input : *A MM Instance* $I = \langle W, F, \succ, \boldsymbol{q} \rangle$*, a cost vector* $\boldsymbol{c} \in \mathbb{N}^m$*, a capacity bound* $k_{max} \in \mathbb{N}$

Question (MinMaxCSP) **:** *Is there a capacity increase vector* $\boldsymbol{r}$ *with* $|\boldsymbol{c} \odot \boldsymbol{r}|_\infty \leq k^{max}$ *st.* $I' = \langle W, F, \succ, \boldsymbol{q} + \boldsymbol{r} \rangle$ *admits a stable and perfect matching?*

Algorithm 5 Algorithm for MinMaxCSP

Input: MM Instance I
  $r[f] := 0$ for all $f \in F$
  $C := 0$
  $\mu :=$ worker-optimal stable matching
  while μ does not match all workers **do**
    $C := C + 1$
    $r[f] = \lfloor C / \boldsymbol{c}[f] \rfloor$
    $\mu :=$ worker-optimal stable matching with capacities $q + r$
  end while
  Return μ

Extending Algorithm 3 for MinMaxCSP is quite easy. In each itertion of the while loop, instead of increasing each firm's capacity, we increase the total cost by 1 and set the firms' capacities accordingly. Checking whether I' admits a stable and perfect matching can be done in linear time, and thus Algorithm 5 solves for MinMaxCSP in polynomial time. Note that if $gcd(\boldsymbol{c}) = \gcd\{\boldsymbol{c}[f_1], \boldsymbol{c}[f_2], ..., \boldsymbol{c}[f_m]\} \neq 1$, then Algorithm 5 can complete faster by updating $C := C + \gcd(\boldsymbol{c})$ in each iteration.

5.3 Question 3

What if we want to match some k-subset of unmatched workers? Formally:

Input : *A MM Instance* $I = \langle W, F, \succ, \boldsymbol{q} \rangle$*, a k-subset* $W' \subseteq W_u$ *of unmatched workers, a capacity bound* $k^+ \in \mathbb{N}$

Question : *Is there a capacity increase vector* $\boldsymbol{r}$ *with* $|\boldsymbol{r}|_1 \leq k^+$ *st.* $I' = \langle W, F, \succ, \boldsymbol{q} + \boldsymbol{r} \rangle$ *admits a stable and perfect matching which matches every worker in* W'*?*

The above problem is similary NP-complete, as choosing $W' = W_{un}$ makes it equivalent to MinSumSP. We can modify Algorithm 2 by setting $L := W'$ to solve for above problem, giving us a $\lceil |W'|/c \rceil$-approximation algorithm for any constant c. Similary, it is also FPT wrt. $(|W'|, \Delta_u)$.

BIBLIOGRAPHY

[CC24] Jiehua Chen and Gergely Csáji. "Optimal Capacity Modification for Many-To-One Matching Problems". In: (2024). DOI: 10.48550/arXiv.2302.01815.

[GS62] David Gale and Lloyd S. Shapley. "College Admissions and the Stability of Marriage". In: *The American Mathematical Monthly* 69.1 (1962), pp. 9–15. ISSN: 00029890, 19300972. DOI: 10.1080/00029890.1962.11989827.

[GS85] David Gale and Marilda Sotomayor. "Some remarks on the stable matching problem". In: *Discrete Applied Mathematics* 11.3 (1985), pp. 223–232. ISSN: 0166-218X. DOI: 10.1016/0166-218X(85)90074-5.

[Gok+24] Salil Gokhale et al. "Capacity Modification in the Stable Matching Problem". In: *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*. AAMAS '24. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2024, pp. 697–705. ISBN: 9798400704864. DOI: 10.48550/arXiv.2402.04645.

[GI89] Dan Gusfield and Robert W Irving. *The Stable Marriage Problem: Structure and Algorithms*. Cambridge, MA, USA: The MIT Press, 1989. ISBN: 9780262515528. URL: https://mitpress.mit.edu/9780262515528/the-stable-marriage-problem/.

[Kom20] Scott Duke Kominers. *Respect for Improvements and Comparative Statics in Matching Markets*. Tech. rep. Harvard University, 2020. URL: http://www.scottkom.com/articles/Kominers_Comparative_Statics.pdf.

[KÜ06] Hideo Konishi and M. Utku Ünver. "Games of capacity manipulation in hospital-intern markets". In: *Social Choice and Welfare* 27.1 (2006), pp. 3–24. ISSN: 01761714, 1432217X. DOI: 10.1007/s00355-006-0097-z.

[Rot84] Alvin E. Roth. "The Evolution of the Labor Market for Medical Interns and Residents: A Case Study in Game Theory". In: *Journal of Political Economy* 92.6 (1984), pp. 991–1016. DOI: 10.1086/261272.

[Rot86] Alvin E. Roth. "On the Allocation of Residents to Rural Hospitals: A General Property of Two-Sided Match ing Markets". In: *Econometrica* 54.2 (1986), pp. 425–427. ISSN: 00129682, 14680262. DOI: 10.2307/1913160.

[RS90] Alvin E. Roth and Marilda A. Oliveira Sotomayor. *Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis*. Econometric Society Monographs. Cambridge University Press, 1990. DOI: 10.1017/CCOL052139015X.

[Sön97] Tayfun Sönmez. "Manipulation via Capacities in Two-Sided Matching Markets". In: *Journal of Economic Theory* 77.1 (1997), pp. 197–204. ISSN: 0022-0531. DOI: 10.1006/jeth.1997.2316.