On Computing Pareto Stable Assignments

Ning Chen[1]

1 **Nanyang Technological University**
Singapore
`ningc@ntu.edu.sg`

Abstract

Assignment between two parties in a two-sided matching market has been one of the central questions studied in economics, due to its extensive applications, focusing on different solution concepts with different objectives. One of the most important and well-studied ones is that of stability, proposed by Gale and Shapley [8], which captures fairness condition in a model where every individual in the market has a preference of the other side. When the preferences have indifferences (i.e., ties), a stable outcome need not be Pareto efficient, causing a loss in efficiency. The solution concept Pareto stability, which requires both stability and Pareto efficiency, offers a refinement of the solution concept stability in the sense that it captures both fairness and efficiency.

We study the algorithmic question of computing a Pareto stable assignment in a many-to-many matching market model, where both sides of the market can have multiunit capacities (i.e., demands) and can be matched with multiple partners given the capacity constraints. We provide an algorithm to efficiently construct an assignment that is simultaneously stable and Pareto efficient; our result immediately implies the existence of a Pareto stable assignment for this model.

1998 ACM Subject Classification F.2.0 [Analysis of Algorithms and Problem Complexity] General

Keywords and phrases Algorithm, stable matching, Pareto efficiency

Digital Object Identifier 10.4230/LIPIcs.STACS.2012.384

1 Introduction

Two-sided matching markets have been extensively studied since the seminal work of Gale and Shapley on stable marriage [8], where there are a set of men and women, each with a strict preference ranking over members of the other side. A matching between the men and women is *stable* if there is no man-woman pair who both strictly prefer each other to their current partners. The concept of stability captures fairness condition for market participants and has had enormous influence on the design of real world matching markets [18]. The original marriage model, as well as many of its generalizations, have been thoroughly investigated.

A practical reality of matching markets is ties, or indifferences: Agents may not be able to strictly rank their prospective partners, i.e., they might be indifferent among some of them. The introduction of ties into preference lists dramatically changes the properties and structure of the set of stable matchings. For instance, man or woman-optimal stable matchings are no longer well-defined [19], and stable matchings need not all have the same cardinality. The problem of finding a maximum cardinality stable matching becomes NP-hard [11, 16], and much work has focused on finding approximation solutions [13, 17]. In addition, arguably more importantly, *stability no longer guarantees Pareto efficiency* (roughly speaking, Pareto efficiency means that no other feasible solution exists that improves some agent without

29th Symposium on Theoretical Aspects of Computer Science (STACS'12).
Editors: Christoph Dürr, Thomas Wilke; pp. 384–395
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany
SYMPOSIUM ON THEORETICAL ASPECTS OF COMPUTER SCIENCE

hurting everyone else)[1], an observation that has received much attention in the economics literature (e.g., [2, 22, 6, 1, 7]). In particular, as Erdil and Ergin [6] demonstrate, simply using a matching returned by the Gale-Shapley deferred acceptance algorithm can cause quite a severe loss in efficiency.

To capture both fairness and efficiency, Sotomayor [22] suggests *Pareto stability* (i.e., both stable and Pareto efficient) as a natural solution concept for matching markets in the presence of indifferences. A natural question is then whether such a matching always exists, and how to find one efficiently. Note that the presence of ties in the preference lists cannot be addressed by the standard trick of introducing small perturbations: If ties are broken arbitrarily, the set of stable matchings with respect to the new strict preferences can be strictly smaller than the set of stable matchings with respect to the original preferences with ties — that is, artificial tiebreaking does not preserve the set of stable matchings in the original problem, thus, may not generate a Pareto efficient matching.

This question has recently been addressed by Erdil and Ergin [6, 7] for the *many-to-one* matching model, where one side of the market can have multi-unit capacity. The authors showed an algorithm to find a Pareto stable assignment, when an agent's preference over subsets of neighbors is the natural partial order derived from preferences over individual neighbors. In recent years, there are a growing number of instances of *many-to-many* matching markets, such as online labor markets, course assignments, and the UK medical intern markets, where agents on both sides might want to transact with multiple agents from the other side. In course assignments, for instance, students may register for multiple courses and have preferences over them; on the other hand, courses may have implicit preferences over students according to their, e.g., years of study, majors.

The generalization to multi-unit demand on both sides is nontrivial: The many-to-many stable matching problem behaves rather differently from the many-to-one and one-to-one models in terms of the properties and structure of the set of solutions [19, 5, 21, 22]. In this paper, we study Pareto stability with indifference for many-to-many matching, by considering the natural generalization of the Erdil and Ergin model to many-to-many market: What happens when agents have the same preference model over subsets of acceptable partners as in [6, 7], but agents on both sides can have capacities greater than one? Our main result is the following:

Theorem. *For many-to-many matching markets with indifferences, a Pareto stable assignment always exists and can be computed in polynomial time.*

1.1 Algorithmic Ideas

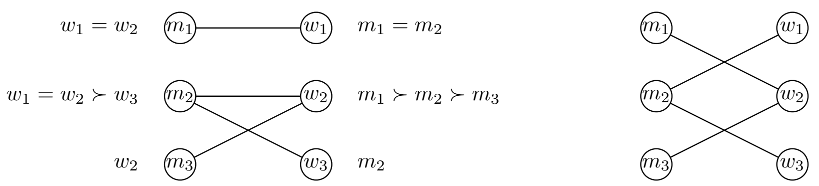
The algorithm of Erdil and Ergin [6, 7] for the many-to-one matching model depends on two observations: First, an assignment has a Pareto improvement (i.e., another assignment where no one gets worse off and at least one agent gets better off) only if the assignment graph does have an augmenting path or cycle (formal definitions refer to Section 2). Second, more critically, any Pareto improvement to a stable assignment preserves stability. These observations immediately imply an algorithm to find a Pareto stable assignment: Starting from any stable assignment, keep making Pareto improvements by eliminating augmenting

[1] For example, there are two men m_1, m_2 and two women w_1, w_2, where m_1 strictly prefers w_1 to w_2, but all others are indifferent amongst their possible partners. The matching $(m_1, w_2), (m_2, w_1)$ is stable, but not Pareto efficient since m_1 can be reassigned to w_1 and m_2 to w_2 without making anyone worse off.

paths and cycles until none remain, and the resulting matching will be both stable and Pareto efficient.

In the many-to-many setting, however, while the first property still holds, we observe that the second critical property fails: A Pareto improvement to a stable assignment need not preserve stability, as the following example shows.

▶ **Example 1** (Pareto improvement does not preserve stability). *Consider the example in the following figure, where m_2 and w_2 have capacity two each and other agents all have unit capacity. Preferences are specified next to each node, e.g., m_2 is indifferent between w_1 and w_2 and prefer both of them to w_3.*



It can be seen that the assignment on the left is stable, and the assignment on the right is a Pareto improvement where w_2 strictly improves her assignment and no one gets worse off. However, the assignment on the right is unstable as m_2 and w_2 would like to match with each other rather than w_3 and m_3 respectively, i.e., it is a blocking pair.

Since Pareto improvements need not preserve stability, all previous approaches (e.g., [6, 7, 4]) computing a Pareto stable assignment in variant models fail. Further, even the existence of a Pareto stable assignment in the many-to-many setting is unclear. We will give an explicit algorithm to compute a Pareto stable assignment, which implies the existence immediately. The above example already shows that the approach of starting with an arbitrary stable assignment and making Pareto improvements will not work, since this need not preserve stability. Further, for the given stable assignment in the above example (left), there is only one Pareto improvement (right); thus, the problem cannot be solved by a careful selection of Pareto improvements.

Instead of using a stability preserving Pareto improvement approach, our algorithm builds on the idea of Roth and Vande Vate [20], who provide an alternative to the deferred acceptance algorithm to compute a stable (one-to-one) matching. Their algorithm can be interpreted as follows: Assume that all women are present at the beginning, and men 'arrive' one by one. We start with the empty matching. When a new man m arrives, match him to a most preferred woman w with whom he forms a blocking pair, if any; if this woman was already matched to a man m', set m' free and consider him as the next arriving man; the algorithm runs iteratively until all men have arrived. Since every woman who changes her partner in this process gets a strict improvement and no woman ever becomes worse off, the algorithm terminates; the final matching is stable, since by construction the matching at every man's arrival is stable.

Our algorithm, like [20], assumes all women are available and considers men one by one (precisely, increases their capacities unit by unit). When the capacity of a man is increased by one, we do a sequence of reassignments to guarantee stability (with respect to the current considered capacities). Further, we ensure that no woman ever becomes worse off, and that some woman strictly improves her assignment in each phase. The algorithm hence will eventually terminate and lead to a stable matching; it remains to consider Pareto efficiency.

An important idea in our algorithm to derive Pareto efficiency is that in the process of reassignments, no augmenting cycles have ever been introduced in the matching; but, on the other hand, we allow the existence of augmenting paths. The key component of our algorithm is a subroutine for eliminating augmenting paths while preserving stability (and introducing no augmenting cycles). Having constructed a matching which is stable and contains no augmenting cycles, we apply the subroutine to eliminate augmenting paths in a stability preserving fashion, which finally yields a Pareto stable matching.

1.2 Related Work

There is a vast literature studying various aspects of the original stable marriage model of Gale and Shapley [8], as well as many of its variants. For a nice review of the very large economics literature on the subject, see the book by Roth and Sotomayor [19] and the survey by Roth [18]; for an introduction to algorithmic and computational issues, see, for instance, the textbook by Gusfield and Irving [9] and the survey by Iwama and Miyazaki [12].

When preferences have indifferences, Irving [14] defined two different notions, weak stability and strong stability, to capture different levels of stability. These two solution concepts, while conceptually similar, have rather different properties. The stability considered in our paper corresponds to weak stability. The computer science literature has largely focused on, e.g., approximating the maximum cardinality weakly stable matching (e.g., [13, 17]) or computing strongly stable matchings (e.g. [14, 15, 10, 3]).

Another related work is [4] which also considers Pareto stable solutions in many-to-many settings; however, in the work of [4], every pair of agents can transact any number of units (e.g., money transfer) which is very different from the present paper where at most one unit can be assigned. Specifically, in the model of [4], the stability-preserving and augmenting path/cycle elimination properties still hold; thus, a Pareto stable solution exists trivially and the algorithm in Erdil and Ergin [6, 7] can be applied directly. The solution structures, algorithm ideas and technical details in our paper are all quite different from [6, 7, 4].

2 Preliminaries

In a two-sided marketplace, let M be the set of men and W be the set of women. Throughout this paper, we will use $m \in M$ to denote a man and $w \in W$ to denote a woman, and use $x, y, z \in M \cup W$ to denote any individual agent (man or woman). For each agent $x \in M \cup W$, let $c_x \in \mathbb{N}$ be his/her capacity, which is the maximum number of agents on the opposite side that can be matched to x. The presence of capacities allows us to assume, without loss of generality, that $|M| = |W| = n$, as dummy agents with $c_x = 0$ can be added to the market.

Each man $m \in M$ has a *preference* list P_m ranking individual women, denoted by $\succ$ and $=$, where $w_1 \succ w_2$ means that the man (strictly) prefers w_1 to w_2, and $w_1 = w_2$ means that m is indifferent between them. We say m weakly prefers w_1 to w_2 if either $w_1 \succ w_2$ or $w_1 = w_2$, denoted by $w_1 \succeq w_2$. Every two women in P_m are comparable and the preference is assumed to be transitive. The preference P_m gives individual women that are acceptable to m, and it may only contain a partial list of women (i.e., m does not want to be matched with any woman that is not on the list). For example, a possible preference list for m is $P_m : (w_1 = w_2 \succ w_3 = w_5)$: here, m is indifferent between w_1 and w_2, and prefers either of them to w_3, w_5, amongst which m is indifferent; he finds all other partners unacceptable. The preference list P_w for each woman $w \in W$ is defined similarly. Let $E = \{(m, w) \mid m \in P_w, w \in P_m\}$ be the set of mutually acceptable pairs. The problem then can be encoded as a bipartite graph $(M, W; E)$ where every vertex has a capacity and a preference over its neighbors.

Notice that the preference lists P_m and P_w defined above are over individual neighbors. Since agents can have capacities greater than one, we also need to define preferences over subsets of neighbors. For this, we adopt the preference model used by Erdil and Ergin [6, 7] in their work for many-to-one markets — the preference ordering over individuals defines a natural ranking over subsets of acceptable partners — given a subset $S \subseteq W$ and two women $w, w' \notin S$, m prefers $S \cup \{w\}$ to $S \cup \{w'\}$ if and only if m prefers w to w' (it is allowed that $w, w' = \emptyset$). In addition, the preference is transitive, i.e., if m prefers S_1 to S_2 and S_2 to S_3, it prefers S_1 to S_3 as well. The preferences of women are defined similarly. For example, if $P_m = (w_1 \succ w_2 \succ w_3 \succ w_4)$, then m prefers $\{w_1, w_2, w_3\}$ to $\{w_1, w_2, w_4\}$, also m prefers $\{w_1, w_3\}$ to $\{w_2\}$ (via $\{w_2, w_3\}$ in the middle). Note that this preference over subsets only constitutes a partial order — specifically, some subsets may not be comparable — for example, m cannot compare (or equivalently, is indifferent between) the sets $\{w_1, w_4\}$ and $\{w_2, w_3\}$.[2]

Given the preferences of all agents, our objective is to establish a multi-unit pairing between men and women, called an *assignment* (or b-matching). An assignment is denoted by $\mu = (\mu_{mw})_{m \in M, w \in W}$, where $\mu_{mw} = 1$ means that m and w are matched and $\mu_{mw} = 0$ otherwise. A *feasible* assignment is one that satisfies the following conditions: $\sum_w \mu_{mw} \leq c_m$ and $\sum_m \mu_{mw} \leq c_w$, and $\mu_{mw} = 1$ only if $(m, w) \in E$ (i.e., m and w are mutually acceptable). All assignments considered in this paper are feasible.

2.1 Solution Concepts

We will consider the following solution concepts.

▶ **Definition 2** (Stability). We say that a feasible assignment $\mu = (\mu_{mw})$ is (pairwise) stable if there is no pair $(m, w) \in E$ (called a blocking pair), $\mu_{mw} = 0$, satisfying one of the following conditions:

- Both m and w have leftover capacity;
- m has leftover capacity and there is m', $\mu_{m'w} = 1$, such that w strictly prefers m to m'; or w has capacity remaining and there is w', $\mu_{mw'} = 1$, such that m prefers w to w';
- There are m' and w', $\mu_{mw'} = 1$ and $\mu_{m'w} = 1$, such that m strictly prefers w to w' and w strictly prefers m to m'.

Note that both members of a blocking pair are able to improve their assignments respectively by matching with each other (and possibly breaking some of the current assignments). A stable assignment always exists, and can be found using a variant of Gale-Shapley's deferred acceptance algorithm [8] for computing stable matchings (by making c_x copies for each individual $x \in M \cup W$ with the same preference list).

We next define Pareto efficiency. Roughly speaking, an assignment is Pareto efficient if there is no other feasible assignment where no agent is worse off, and at least one agent is strictly better off. The formal definition is given below.

[2] The preference we consider is called *responsive* preference in economics (see, e.g., [19]). This model of preferences with multi-unit capacity is both simple, since agents continue to only express preferences over individuals, and is arguably natural for settings where the benefit from a partner to an agent does not depend upon the agent's remaining partners.

▶ **Definition 3** (Pareto efficiency)**.** Given a feasible assignment $\mu = (\mu_{mw})$, let $S_x(\mu)$ be the subset of individuals assigned to x in μ. We say that $\mu' = (\mu'_{mw})$ is a Pareto improvement of μ if for all $x \in M \cup W$, x weakly prefers $S_x(\mu')$ to $S_x(\mu)$, and the preference is strict for at least one agent. An assignment μ is called Pareto efficient if it does not have any Pareto improvement.

Recall from Introduction that when preference lists contain ties, a stable assignment need not be Pareto efficient. This leads naturally to the concept of Pareto stability [22], which combines both Pareto efficiency and stability to provide a stronger solution concept to choose from amongst the set of feasible assignments.

▶ **Definition 4** (Pareto stability)**.** A feasible assignment is *Pareto stable* if it is both stable and Pareto efficient.

2.2 Characterization of Pareto Efficiency

Given the connection between matching and network flow, it is not surprising that the existence of augmenting paths and cycles in an assignment is closely related to whether it can be improved, i.e., its Pareto efficiency. The main difference in the context of stable assignment is that nodes have preferences in addition to capacities; thus, augmenting paths and cycles must improve not just the size of an assignment, but also its quality, as determined by node preferences. We first define augmenting path and cycle in the context of stable assignment.

▶ **Definition 5** (Augmenting Path)**.** Given an assignment $\mu = (\mu_{mw})$, we say that

$$[m_0, w_1, m_1, \ldots, w_\ell, m_\ell, w_{\ell+1}]$$

is an augmenting path if (i) $\sum_w \mu_{m_0 w} < c_{m_0}$ and $\sum_m \mu_{m w_{\ell+1}} < c_{w_{\ell+1}}$, (ii) $\mu_{m_k w_k} = 1$ and $\mu_{m_{k-1} w_k} = 0$ for all k, and (iii) m_k weakly prefers w_{k+1} to w_k and w_k weakly prefers m_{k-1} to m_k.

The first condition says that the capacities of m_0 and $w_{\ell+1}$ are not exhausted. The second condition says that pairs alternatively are not and are in the current assignment μ along the path. The last condition ensures that we are able to get a Pareto improvement by reassigning matches according to the augmenting path. That is, removing all pairs (m_k, w_k) and matching all pairs (m_k, w_{k+1}) give a feasible assignment, which is a Pareto improvement over μ (where no one is worse off and m_0 and $w_{\ell+1}$ are better off).

▶ **Definition 6** (Augmenting Cycle)**.** Given an assignment $\mu = (\mu_{ij})$, we say that

$$[m_1, w_2, m_2, \ldots, w_\ell, m_\ell, w_1, m_1]$$

is an augmenting cycle if (i) $\mu_{m_k w_k} = 1$ and $\mu_{m_k w_{k+1}} = 0$ for all k (where $w_{\ell+1} = w_1$) (ii) m_k weakly prefers w_{k+1} to w_k and w_k weakly prefers m_{k-1} to m_k, and at least one of these preferences is strict.

Again, we are able to match all pairs (m_k, w_{k+1}) and unmatch all pairs (m_k, w_k) in an augmenting cycle to get a Pareto improvement. For a given assignment, an augmenting path or cycle can be found easily by a network flow approach.

The following lemma characterizes the relation between stable assignment and augmenting path and cycle (its proof is the same as the one for many-to-one matching market [6]).

▶ **Lemma 7.** *A feasible assignment is Pareto efficient if and only if it has no augmenting path or cycle.*

3 Algorithm

In this section, we will give an efficient algorithm to compute a Pareto stable assignment. By the above characterization Lemma 7, it suffices to find a stable assignment without containing any augmenting path and cycle. We will apply the idea of the Roth and Vande Vate algorithm [20] that computes a stable one-to-one matching as the high level structure of our algorithm: Initially all individuals are available; women are with *full* capacities and men are with *null* capacity. We increase capacities of men unit by unit, and do a number of reassignments in the process. In the course of the algorithm, the current assignment always has the following invariants (with respect to the current capacities):

- **Stability preserving:** *it is always stable.*
- **No augmenting cycle:** *it does not contain any augmenting cycle.*
- **Women improving:** *the assignments of all women do not get worse off and overall keep improving (this implies that the algorithm always terminates).*

Why do we need to maintain the invariant that the algorithm contains no augmenting cycles, whereas it is allowed to have augmenting paths? Observe that the reason that a Pareto improvement may not preserve stability is that the path or cycle corresponding to the Pareto improvement contains a matched pair (m, w) where both m and w are also matched to a less preferred agent, say w' and m'. When the match (m, w) is removed in the reassignment process of the augmenting path/cycle, even though m and w could receive better partners in the path or cycle, they will prefer to be matched to each other instead of w' and m' respectively. For augmenting path, however, we can always start reassignment from one side of the path (say, the man), and stop proceeding along the path when we reach such a woman w (then (m', w) is unmatched and the process restarts). In this stability-preserving process, a woman becomes strictly better off. However, for the pair (m, w) in an augmenting cycle, we would need to release both (m', w) and (m, w') to preserve stability. That is, we would no longer have the monotonically improving property for women's assignments, which is critical to the analysis of the algorithm.

The high-level structure of the algorithm is described below:

PARETO-STABLE-ALG

1. `Initialization`
   - `there are no assigned edges (i.e.,` $\mu = 0$`) between` M `and` W
   - `all women have their full capacities available`
   - `let` $\mathbf{d} = (d_m)_{m \in M}$ `be a virtual capacity vector of men; initially` $d_m = 0$ `for` $m \in M$
2. `While there is` $m \in M$ `such that` $d_m < c_m$
   - `run` INCREASE-CAP$(\mathbf{d})$
3. `While there is an augmenting path` P
   - `run` ELIMINATE-PATH(P)
4. `Return the final assignment`

Note that in the algorithm, $\mu = (\mu_{mw})_{m \in M, w \in W}$ and $(d_m)_{m \in M}$ are global variables in both subroutines. The first subroutine, INCREASE-CAP, increases the virtual capacity of a man by one and does a number of reassignments to ensure the three invariants listed above (in particular, it guarantees that the assignment is stable for the increased virtual capacity vector). The second subroutine, ELIMINATE-PATH, eliminates all possible augmenting paths to derive a Pareto efficient assignment in a stability preserving fashion. The two subroutines are not completely independent: We may call the second subroutine in the process of the first one, and vice versa. After all augmenting paths have been eliminated, by Lemma 7, the returned assignment is Pareto stable.

While the algorithm may look a bit complicated as the two subroutines may call each other, the fact that no women ever get worse off in the process implies a simple, but critical, structure of the algorithm: we iteratively do a sequence of reassignments to improve women's assignments while preserving stability and containing no augmenting cycle. If at any moment in the algorithm a woman's assignment gets strictly improved, no matter at which stage the algorithm is, we terminate that thread immediately and go to Step (2) of the main algorithm to repeat the process given the current virtual capacity vector $\mathbf{d}$. Such monotonically improving property is crucial to the analysis of the algorithm.

We will describe the two subroutines in detail in the following subsections. (All discussions are with respect to the considered virtual capacity vector.) In the algorithm, for any (augmenting) cycle C and a pair $(m,w) \in C$, we use $C \setminus \{(m,w)\}$ to denote the path by removing pair (m,w) from C.

3.1 Subroutine One: Capacity Increment

The first subroutine that increases virtual capacities of the men is the following.

Increase-Cap($\mathbf{d}$)

1. Pick an arbitrary man m with $d_m < c_m$
2. Let $d_m \leftarrow d_m + 1$, i.e., increase the virtual capacity of m by one
3. Let $S = \{w \mid (m,w)$ is a blocking pair$\}$
4. Let $T = \{w \in S \mid m$ prefers $w \succeq w'$ for any $w' \in S\}$
5. If $T = \emptyset$ (i.e., there is no blocking pair), <u>return</u>
6. Otherwise
   a. If there exists $w \in T$ such that adding match (m,w) does not introduce any augmenting cycle
      - pick such a woman w'
      - add match (m,w')
   b. Otherwise
      - pick an arbitrary $w' \in T$
      - let C be a potential augmenting cycle by adding (m,w')
      - let $P = \left[m \xrightarrow{C\setminus\{(m,w')\}} w'\right]$ be the path from m to w' through $C \setminus \{(m,w')\}$
      - run Eliminate-Path(P)
   c. If w' (defined either in Step (6.a) or (6.b)) is over-matched (i.e., matched to more than $c_{w'}$ neighbors)
      - let m' be a least preferred man matched to w' where deleting (m',w') does not introduce an augmenting cycle
      - delete match (m',w')
      - let $d_{m'} \leftarrow d_{m'} - 1$
      - <u>return</u>
   d. Otherwise, <u>return</u>

When the virtual capacity of m is increased by one, there might be some blocking pairs, among which the subroutine tries to match m to one that he prefers most ($w' \in T$ in the above description). However, this could introduce potential augmenting cycles (Step 6(b)).

Instead of matching m and w' directly, the subroutine considers a potential augmenting cycle C incurred by (m, w') and tries to do reassignments according the other path from m to w' along the cycle. Finally, if w' is over-matched, then we delete one of her least preferred assignments without incurring any augmenting cycles and delete the virtual capacity of that man by one. This guarantees that the assignment remains stable, and the assignment of w' strictly improves.

The existence of m' in Step 6(c) is guaranteed by the following lemma.

▶ **Lemma 8.** *Given a stable assignment without augmenting cycles, for any woman w let $S \subseteq M$ be the subset of men matched to w to whom w is least preferred. Then there is $m \in S$ such that deleting match (m, w) does not introduce any augmenting cycle.*

3.2 Subroutine Two: Augmenting Path Elimination

Consider any given stable assignment, assume that there is an augmenting path $P = [m_0, w_1, m_1, \dots, w_\ell, m_\ell, w_{\ell+1}]$, where (m_i, w_i) is in the assignment and (m_i, w_{i+1}) is not. Note that it is possible that an individual x (either a man or a woman) or a pair (x, y) appears more than once in P. In this subsection, when we refer to an individual $x \in P$ or a pair $(x, y) \in P$, we denote the corresponding one at that position of P.

Before describing the subroutine, we will first consider a truncation process, which deletes some pairs in a given augmenting path according to different appearances of the same agent and will will be used in the subroutine.

3.2.1 Truncation.

For a given augmenting path P, we consider the following truncation function.

TRUNC-PATH(P)

1. **while** one of the following "if" conditions holds
   - If there is m such that $P = [\dots, m, w_1, \dots, w_2, m, \dots]$ and m weakly prefers w_1 to w_2
     - truncate $P = [\dots, m, \; \sout{(w_1, \dots, w_2, m,)} \; \dots]$
   - If there is w such that $P = [\dots, w, m_1, \dots, m_2, w, \dots]$ and w weakly prefers m_2 to m_1
     - truncate $P = [\dots, w, \; \sout{(m_1, \dots, m_2, w,)} \; \dots]$
2. Return path P

It can be seen that if TRUNC-PATH(P) is executed, by the rules of the truncation, no pair (x, y) can appear more than once after truncation. However, it is still possible that an individual appears more than once (e.g., when m strictly prefers w_2 to w_1, we do not truncate the two occurrences of m). The truncation process is necessary in our algorithm; in particular, it is important to the analysis of termination of the algorithm.

We have the following observation.

▶ **Lemma 9.** *For any given augmenting path P,* TRUNC-PATH(P) *returns an augmenting path as well.*

3.2.2 Elimination.

We next describe the subroutine to eliminate augmenting paths while preserving the three invariants listed at the beginning of the section. Note that for any augmenting path, its one side must be a man and the other side must be a woman. The subroutine starts from the man side and considers pairs one by one. Hence, for any man-woman pair in the path, the objective is to match them; and for any woman-man pair in the path, the objective is to unmatch them.

ELIMINATE-PATH(P)

1. Assume $P = [m^*, w_1, m_1, \ldots, w^*]$
2. Let $e = (m^*, w_1)$ be the first pair on path P
3. **while** $e \neq \emptyset$
   - If e is not a match (i.e., $e = (m, w)$)
     - if adding match (m, w) does not introduce an augmenting cycle
       a. add match (m, w)
       b. if w is not over-matched, <u>return</u>
       c. if w strictly prefers m to a current partners
          * let m' be a least preferred man matched to w where deleting (m', w) does not introduce an augmenting cycle (by Lemma 8, such m' exists)
          * delete match (m', w)
          * let $d_{m'} \leftarrow d_{m'} - 1$
          * <u>return</u> to Step 2 of the main algorithm PARETO-STABLE-ALG to run INCREASE-CAP
       d. else let e be the next pair after (m, w) in P
     - otherwise
       e. let $C = [m, w'_1, m'_1, \ldots, w'_k, m'_k, w, m]$ be such a potential cycle if adding (m, w)
       f. expand $P = \left[m^*, \ldots, m, w'_1 \xrightarrow{C\setminus\{(m,w)\}} m'_k, w, \ldots, w^*\right]$
       g. truncate $P = \left[m^*, \ldots, \text{TRUNC-PATH}\left(m, w'_1 \xrightarrow{C\setminus\{(m,w)\}} m'_k, w, \ldots, w^*\right)\right]$
       h. let e be the first pair returned by the TRUNC-PATH
   - If e is a match (i.e., $e = (w, m)$)
     - if deleting match (w, m) does not introduce an augmenting cycle
       i. delete match (w, m)
       j. let e be the next pair after (w, m) in P
     - otherwise
       k. run the above Steps (e,f,g,h)
          (switching the notations of m and w (except m^* and w^*))

The subroutine tries to add and delete matches one by one along pairs in the path P. If the current considered pair is a man-woman pair (i.e., $e = (m, w)$), the subroutines matches them if it does not introduce any augmenting cycle. If the assignment of w is strictly improved (i.e., the condition in Step (3.b) or (3.c) is satisfied), the subroutine terminates. Note that at this point the subroutine may not completely eliminate the augmenting path, however, the overall assignment of the woman gets strictly improved and the process restarts at the capacity increment stage. If matching m and w will introduce a potential augmenting cycle, instead of adding the match directly, the subroutine takes a "detour" and considers the other path from m to w along the cycle and expands it to the path P (Step 3(f); by the following Lemma 10, it is a valid expansion). Then the subroutine will do a truncation from m to the end of the path P and restarts the process by considering the first pair returned by the truncation (its first individual must be m). The subroutine performs similarly if the considered pair is a woman-man pair.

We first establish the following observations.

▶ **Lemma 10.** *The expansion of path P in Step (3.f) is a well-defined augmenting path.*

We have the following key claim, which implies that the subroutine always terminates.

▶ **Lemma 11** (Main). *The subroutine* Eliminate-Path(P) *terminates in finite number of steps for any augmenting path* P*.*

3.3 Analysis of the Algorithm

Again, the high level structure of the algorithm is to increase capacities of men and eliminate augmenting paths. While the algorithm may look involved, as the virtual capacity is not always monotonically increasing (e.g., in Step 6(c) of Increase-Cap and Step 3(c) of Eliminate-Path, we actually need to reduce the virtual capacities) and two subroutines may call each other, there is a simple, but crucial, idea behind the algorithm: the assignments of women keep improving (this is the exact reason that we do not want to introduce any augmenting cycle in the course of the algorithm). Therefore, at any moment of the algorithm, if a woman's assignment gets improved (e.g., Step 6(c) of Increase-Cap and Step 3(b), 3(c) of Eliminate-Path), the algorithm will abandon the current subroutine and restart the whole process (i.e., capacity increment and augmenting path elimination) starting from the current virtual capacity vector. Since every woman can improve her assignment at most n^2 times (as her capacity is at most n and every unit capacity can be improved at most n times), the whole algorithm will terminate.

It is easy to see that the three invariants listed at the beginning of the section are maintained in the course of the algorithm. Indeed, the last two (no augmenting cycle and women not worse off) hold trivially as they are guaranteed by the algorithm itself. For stability, in the subroutine Increase-Cap, when increasing the virtual capacity of m by one, we try to match m with a most preferred woman w where (m, w) forms a blocking pair. If w is not over-matched, then the resulting assignment is still stable. Otherwise, we delete a match (m', w) where m' is a least preferred man matched to w and reduce the virtual capacity of m' by one (Step (6.c) of Increase-Cap); this implies that the resulting assignment is still stable with respect to the new capacity vector. For the second subroutine Eliminate-Path, stability comes from the definition of augmenting path and the fact that when we delete a match (w, m), we know that m must be a least preferred man matched to w and w was over-matched (otherwise, when we add the match right before (w, m), the assignment of w gets strictly improved and the subroutine will run Step (3.b) or (3.c) to terminate). Therefore, the final returned assignment is stable.

By the rule of the algorithm Pareto-Stable-Alg, when it terminates there is no augmenting path. By the invariant that there is no augmenting cycle, we know that the returned assignment is Pareto efficient. We conclude with the following result.

▶ **Theorem 12.** *The algorithm* Pareto-Stable-Alg *computes a Pareto stable assignment in polynomial time.*

Acknowledgements

I am grateful for Arpita Ghosh for many valuable discussions and suggestions, and also thank for the anonymous reviewers for their helpful comments.

References

1 A. Abdulkadiroglu, P. Pathak, A. E. Roth, *Strategy-Proofness versus Efficiency in Matching with Indifferences: Redesigning the NYC High School Match*, American Economic Review, V.99(5), 1954-1978, 2009.

2 A. Abdulkadiroglu, T. Sonmez. *School Choice: A Mechanism Design Approach*, American Economic Review, V.93(3), 729-747, 2003.
3 N. Chen, A. Ghosh, *Strongly Stable Assignment*, ESA 2010, 147-158.
4 N. Chen, A. Ghosh, *A Market Clearing Solution for Social Lending*, IJCAI 2011, 152-157.
5 F. Echenique, J. Oviedo, *A Theory of Stability in Many-to-Many Matching Markets*, Theoretical Economics. V.1(2), 233-273, 2006.
6 A. Erdil, H. Ergin, *What's the Matter with Tie-breaking? Improving Efficiency in School Choice*, American Economic Review, V.98(3), 669-689, 2008.
7 A. Erdil, H. Ergin, *Two-Sided Matching with Indifferences*, working paper.
8 D. Gale, L. S. Shapley, *College Admissions and the Stability of Marriage*, American Mathematical Monthly, V.69, 9-15, 1962.
9 D. Gusfield, R. W. Irving, *The Stable Marriage Problem: Structure and Algorithms*, MIT Press, 1989.
10 T. Kavitha, K. Mehlhorn, D. Michail, K. E. Paluch, *Strongly Stable Matchings in Time O(nm) and Extension to the Hospitals-Residents Problem*, ACM Transactions on Algorithms, V.3(2), 2007.
11 K. Iwama, D. Manlove, S. Miyazaki, Y. Morita, *Stable Marriage with Incomplete Lists and Ties*, ICALP 1999, 443-452.
12 K. Iwama, S. Miyazaki, *Stable Marriage with Ties and Incomplete Lists*, Encyclopedia of Algorithms, 2008.
13 K. Iwama, S. Miyazaki, N. Yamauchi, *A 1.875-Approximation Algorithm for the Stable Marriage Problem*, SODA 2007, 288-297.
14 R. W. Irving, *Stable Marriage and Indifference*, Discrete Applied Mathematics, V.48, 261-272, 1994.
15 D. F. Manlove, *Stable Marriage with Ties and Unacceptable Partners*, Technical Report TR-1999-29, University of Glasgow, 1999.
16 D. F. Manlove, R. W. Irving, K. Iwama, S. Miyazaki, Y. Morita, *Hard Variants of Stable Marriage*, Theoretical Computer Science, V.276(1-2), 261-279, 2002.
17 E. McDermid, *A 3/2-Approximation Algorithm for General Stable Marriage*, ICALP 2009, 689-700.
18 A. E. Roth, *Deferred Acceptance Algorithms: History, Theory, Practice, and Open Questions*, International Journal of Game Theory, 537-569, 2008.
19 A. E. Roth, M. Sotomayor, *Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis*, Cambridge University Press, 1992.
20 A. Roth, J. Vande Vate, *Random Paths to Stability in Two-Sided Matching*, Econometrica, V.58, 1475-1480, 1990.
21 M. Sotomayor, *Three Remarks on the Many-to-Many Stable Matching Problem*, Mathematical Social Sciences, V.38(1), 55-70, 1999.
22 M. Sotomayor, *The Pareto-Stability Concept is a Natural Solution Concept for Discrete Matching Markets with Indifferences*, International Journal of Game Theory, 2010.