Strongly Stable Assignment

Ning Chen[1] and Arpita Ghosh[2]

[1] School of Physical and Mathematical Sciences,
Nanyang Technological University, Singapore.
ningc@ntu.edu.sg
[2] Yahoo! Research, Santa Clara, CA, USA.
arpita@yahoo-inc.com

Abstract. An instance of the stable assignment problem consists of a bipartite graph with arbitrary node and edge capacities, and arbitrary preference lists (allowing both ties and incomplete lists) over the set of neighbors. An assignment is strongly stable if there is no blocking pair where one member of the pair strictly prefers the other member to some partner in the current assignment, and the other weakly prefers the first to some partner in its current assignment.
We give a strongly polynomial time algorithm to determine the existence of a strongly stable assignment, and compute one if it exists. The central component of our algorithm is a generalization of the notion of the *critical set* in bipartite matchings to the *critical subgraph* in bipartite assignment; this generalization may be of independent interest.

1 Introduction

The classical stable marriage problem studies a setting with an equal number of men and women, each with a strict preference ranking over all members of the other side. Since the seminal work of Gale and Shapley on stable marriage [5], a number of variants of the stable matching problem have been studied, relaxing or generalizing different assumptions in the original model. One particularly practical generalization is to relax the requirement of strict and complete preferences over all alternatives: this gives rise to the stable marriage problem with *ties and incomplete lists* (SMTI), where a man can have indifferences, or ties, between women in his preference list, and need not rank all women (and similarly for women).

When preference lists have ties, different notions of stability can be defined depending on what qualifies as a blocking pair: a *weakly stable* matching [7, 9, 15] is one where there is no pair (i, j) such that *both* i and j strictly prefer each other to their matched partners; a *strongly stable* matching [7, 14] is one where there is no pair such that one member of the pair *strictly* prefers the other to its current partner in the matching, and the other member *weakly* prefers the first to its current partner in the matching. Unlike weakly stable matchings, a strongly stable matching need not always exist. Irving [7] gave a beautiful algorithm to solve the question of deciding whether or not a strongly stable matching exists and finding one if it does, and Manlove [14] extended the algorithm to the case of ties and incomplete lists.

Motivated by online matching marketplaces where buyers and sellers trade multiple items, we investigate the generalization of SMTI to assignment problems, where nodes on both sides of a bipartite graph have multiple units of capacity $c(i) \geq 1$ (as opposed to unit capacity in the stable marriage model) and edges in the graph have arbitrary capacities $c(i, j)$. We study the algorithmic question of finding *strongly stable assignments*— feasible assignments where there is no pair (i, j) such that both i and j weakly prefer allocating at least one additional unit on the edge (i, j), and at least one of i and j strictly prefers to do so. The *many-to-many matching* problem, a special case of assignment where at most one unit can be allocated between a pair of nodes, *i.e.*, $c(i, j) = 1$, turns out to be adequate to showcase most of the complexity introduced by the generalization from matching to assignment. For clarity, we therefore state all our results in terms of the many-to-many matching problem[3], and defer the generalization to assignment with arbitrary edge capacities to the full version of the paper.

The generalization from one-to-one matching, where nodes have unit capacity, to many-to-many matching with multi-unit node capacities, introduces significant algorithmic complexity to the problem of strong stability. To explain why, it is first necessary to understand the main idea behind the algorithms of Irving [7] and Manlove [14] for the unit capacity case: Men propose to women in decreasing order of preference, so that at each stage a man's proposals are (all) at the top of his current list and a woman's proposals are (all) at the bottom of her current list. The algorithm deletes pairs that can never occur in a strongly stable matching — this happens at one of two times, the first being when a woman receives a strictly better proposal (exactly as in the deferred acceptance algorithm). The second is when a woman is *over-demanded*, that is, there are multiple men in the engagement graph to whom this woman must be matched to avoid a blocking pair — when this happens, no man at this level, *i.e.*, the bottom of this woman's preference list, can be matched to her in a strongly stable matching, since these other men will form a blocking pair. The algorithms in [7, 14] delete all such pairs which can never occur in any strongly stable matching and then look for a strongly stable matching in a final engagement graph based on these modified lists.

In the many-to-many matching setting, there are two major complicating differences. The first difference is quite fundamental, and comes from the fact that the notion of a *critical set* — defined as the unique maximal subset of men with the largest deficiency, which is the difference between the size of a set and its neighborhood in a bipartite graph — does not generalize in the obvious way to many-to-many matching. With unit capacities, the set of over-demanded women turns out to be precisely the neighborhood of the critical set of men. The obvious generalization when nodes have multi-unit capacities is to define the deficiency

[3] The problem of finding strongly stable many-to-many matchings has been studied previously in [13]; see Section 1.1.

as the difference between the total capacity of a set and its neighborhood — that is, define $\delta(S) = \sum_{i \in S} c(i) - \sum_{j \in N(S)} c(j)$, where $N(S)$ is the set of neighbors of S, and extend the definition of the critical set to be the subset of men maximizing $\delta(S)$[4]. However, this obvious extension of the deficiency and the corresponding definition for the critical set does not work, and fails in two ways — first, the neighborhood of this set does not correctly identify the set of over-demanded women (Example 1). Second, it does not possess the property that the size of a maximum many-to-many matching is given by the total capacity of men minus this maximum deficiency (Example 2).

We therefore need to appropriately extend the notion of critical set to the multi-unit capacity setting — as it turns out, a subset of men $S \subseteq A$ is no longer an adequate description for the extension of a critical set. Rather, we need to specify a *critical subgraph*, which is described by a partition of the set of men *and* their capacities, *as well as* a subset of women. In Section 2, we develop a definition of the critical subgraph that we show *retains both properties* of the critical set in unit-capacity matching, and show that the critical subgraph can be computed in polynomial time. We note that the extension of the critical set from bipartite matching to bipartite assignment is of independent interest, and is one of the major contributions of the paper: The critical set [12] plays a central role in algorithms for computing stable outcomes in matching markets, both without and with monetary transfers [7, 3, 1], by providing a way to identify the set of 'over-demanded' women (or items); our generalization of the critical set therefore might be useful in extending these market-clearing algorithms to marketplaces with multi-unit node capacities.

The second difference from [7] is that with unit capacity, the edges in an engagement graph are all at the same level for each node (top for men, bottom for women), whereas this is not the case with multi-unit capacities — when $c(i) \geq 1$, a man might need to propose to women of different levels in his preference list, and a woman might need to retain proposals from different levels, to meet their respective capacities. This means that not all edges in the engagement graph incident to a node are equally preferred by that node. Therefore, we cannot simply seek a maximum matching or attempt to identify the set of over-demanded women in an engagement graph as in [7], without appropriately processing it to account for the fact that not all edges are equal. In Section 3, we present our algorithm to determine the existence of a strongly stable assignment and compute one, if it exists. All proofs can be found in the full version of the paper [2].

1.1 Related Work

There has been much work on stable matchings in bipartite graphs focusing on different variants and applications of the original stable marriage problem. As mentioned earlier, our work extends the algorithms of Irving [7] and Manlove [14] for the one-to-one matching problem. Irving et al. [8] gave a strongly stable matching algorithm for the many-to-one matching problem (*i.e.*, nodes on one side of the graph can have multi-unit capacities); the algorithm was improved

[4] In fact, this is exactly the definition used in [13].

later by Kavitha et al. [11]. For other related algorithmic problems that arise from the study of stable matchings, see Gusfield and Irving [6] and two recent survey papers by Iwama and Miyazaki [10] and Roth [16]. Economic properties for stable matchings are discussed in [17, 19].

The most obvious work related to ours is [13], which studies the problem of finding strongly stable many-to-many matchings, *i.e.*, the special case of our assignment model with unit edge capacities $c(i, j) = 1$. Unfortunately, however, the algorithm proposed in [13] is incorrect, both in terms of the processing of the engagement graph to account for the fact that the edges incident to a node belong to multiple levels on its preference list, as well as in terms of identifying over-demanded women (that algorithm uses the extension of the critical set based on the difference between total capacities). We give explicit examples showing two different points at which the algorithm in [13] fails in the full version of the paper [2].

2 The Critical Subgraph

This section develops the notion of the *critical subgraph*, generalizing the critical set from unit-capacity matching to the setting with multi-unit capacities. As in the rest of the paper, for simplicity we restrict ourselves to many-to-many matchings: Given a bipartite graph $G = (A, B; E)$ with node capacities $c(k) \geq 1$ for $k \in A \cup B$, a *many-to-many* matching[5] of G is a subset of edges such that each node k is matched to at most $c(k)$ pairs. Let $d_G(k)$ denote the degree of node k in G. Assume without loss of generality that $c(k) \leq d_G(k)$, since a node cannot be matched to more neighbors than its degree in G. Given a subset S of nodes, we use $N_G(S)$ (or simply $N(S)$) denote the set of neighbors of S in G.

2.1 The Critical Set

Given a bipartite graph $G = (A, B; E)$ where all nodes have unit capacity, the *critical set* is the (unique) maximal subset with the largest deficiency, where the *deficiency* of $S \subseteq A$ is defined as the difference between the size of S and the size of its neighborhood in B, *i.e.*, $\delta(S) \triangleq |S| - |N(S)|$. The critical set is closely related to maximum matchings [12, 7] — the size of a maximum matching in G is given by $|A| - \delta(X)$, where $X = \arg\max_{S \subseteq A} \{|S| - |N(S)|\}$ is the critical set of A, and $\delta(X)$ is its deficiency.

In many-to-many matching, nodes can have arbitrary capacities $c(\cdot) \geq 1$. We begin with two examples that show that the obvious generalization of the deficiency of a set $S \subseteq A$ — defining it as the difference between the total capacity of nodes in that set and the total capacity of its neighbors, *i.e.*, $\sum_{i \in S} c(i) - \sum_{j \in N(S)} c(j)$, and defining the critical set to be the (unique) subset maximizing this quantity — fails to capture two important properties of the corresponding definition for unit capacity matching: the neighborhood of the critical set does not correctly identify the set of "over-demanded" women, and the deficiency no longer relates to the size of the maximum matching.

[5] It is also called simple b-matching and is a well-studied concept [18].

An *over-demanded* woman j is one for whom there is some maximum matching in which j has an unmatched neighbor with leftover capacity. Specifically, j is over-demanded if there is a maximum matching M where there is an edge $(i, j) \in E$ such that $(i, j) \notin M$ and the number of matched neighbors of i in M is less than his capacity $c(i)$. The first example illustrates that the neighborhood of the critical set defined this way does not correctly identify the set of over-demanded women when $c(\cdot) \geq 1$.

Example 1. Consider graph $G = (A, B; E)$ where $A = \{i_1, i_2, i_3, i_4\}$ with node capacities $(2, 1, 1, 1)$ and $B = \{j_1, j_2, j_3\}$ with capacities $(2, 1, 1)$. Connect (i_1, j_1), (i_1, j_2) and (i_2, j_1), and connect all nodes in $\{i_2, i_3, i_4\}$ to all nodes in $\{j_2, j_3\}$. The unique subset maximizing $\sum_{i \in S} c(i) - \sum_{j \in N(S)} c(j)$ is A, with neighborhood B. However, j_1 is never over-demanded in any maximum matching, since both edges (i_1, j_1) and (i_2, j_1) belong to every maximum many-to-many matching (recall that at most one edge, or one unit of capacity, can be assigned between any pair (i, j) in a many-to-many matching).

The next example shows that in addition, the size of the maximum matching is not related to the deficiency defined according to the difference of capacities, *i.e.*, it is not true that the size of a maximum matching is given by $\sum_{i \in A} c(i) - \max_{S \subseteq A} \left\{ \sum_{i \in S} c(i) - \sum_{j \in N(S)} c(j) \right\}$ when $c(i), c(j) \geq 1$.

Example 2. Let $(A_1; B_1)$ be a complete bipartite graph with $A_1 = \{i_1, \ldots, i_{10}\}$, $B_1 = \{j_1, \ldots, j_{10}\}$, and $c(i_k) = 10$ for $i_k \in A_1$, $c(j_k) = 4$ for $j_k \in B_1$. Let $(A_2; B_2)$ be another complete bipartite graph with all unit capacity nodes $A_2 = \{i'_1, \ldots, i'_n\}$, $B_2 = \{j'_1, \ldots, j'_{2n}\}$ (n is an arbitrary number). $G = (A, B)$ consists of these two graphs plus an extra node j_0 with capacity $c(j_0) = 11$ (*i.e.*, $A = A_1 \cup A_2$ and $B = B_1 \cup B_2 \cup \{j_0\}$) and edges connecting all nodes in A to j_0. Clearly, any maximum matching of G has size $n + 50$ (e.g., fully match all nodes in B_1 to A_1, all nodes in A_2 to B_2, and j_0 to all nodes in A_1). The maximum deficiency of G is given by A_1 and its neighbor set $B_1 \cup \{j_0\}$, with value $\sum_{i \in A_1} c(i) - \sum_{j \in B_1} c(j) - c(j_0) = 100 - 40 - 11 = 49$. However, $\sum_{i \in A} c(i) - 49 = 100 + n - 49 = n + 51$, which is not the size of the maximum matching.

To appropriately extend the notion of the critical set to multi-unit capacities, we first state the following lemma for the unit capacity case, allowing an alternative view of the critical set.

Lemma 1. *Let X be the critical set of A and $Y = N(X)$ be its neighbor set. X and Y define a partition of the graph $G = (A, B)$ into two subgraphs $G_1 = (S, T)$ and $G_2 = (X, Y)$, where $S = A \setminus X$ and $T = B \setminus Y$. Then any maximum matching M_1 of G_1 has size $|S|$, and any maximum matching M_2 of G_2 has size $|Y|$. Further, $M_1 \cup M_2$ gives a maximum matching of G with size $|S| + |Y| = |A| - (|X| - |Y|)$.*

2.2 Critical Subgraph

We now extend the notion of critical set to the multi-unit capacity setting: when $c(i), c(j) \geq 1$, the critical set $X \subseteq A$ is defined not just by the identities of the nodes in X, but also by a vector of associated reduced capacities. In addition, we cannot simply choose the neighborhood of X to define the deficiency (and the set of over-demanded women): we will need a different partition of the nodes in B as well. We define the critical subgraph of a bipartite graph G below.

Definition 1 (Critical subgraph). *Given a bipartite graph $G = (A, B; E)$ with node capacities $c(k) \geq 1$ for $k \in A \cup B$, for any $i \in A$ and $S \subseteq B$, let $d_S(i) = |\{j \in S \mid (i, j) \in E\}|$ denote the degree of i restricted on S. We say $S \subseteq B$ is a* perfect subset *if there is a maximum matching of G such that every $i \in A$ is matched to $c_S(i) = \min\{c(i), d_S(i)\}$ pairs in S. That is, S is perfect if this maximum matching matches each i to the maximum possible number of neighbors in S. Let $S^* \subseteq B$ be the unique (Corollary 1)* maximal perfect subset *(i.e., S^* is not a proper subset of any other perfect set). (Note that if there is no perfect subset, $S^* = \emptyset$.)*

Define the critical capacity *of A to be $\mathbf{x} = (x(i))_{i \in A}$ where $x(i) = c(i) - c_{S^*}(i)$. Let $X = \{i \in A \mid x(i) > 0\}$ and $Y = B \setminus S^*$. We define (X, Y) to be the* critical subgraph *of G (i.e., an induced subgraph given by X and Y), with capacity $x(i)$ for $i \in X$ and capacity $c(j)$ for $j \in Y$. The* deficiency *of the critical subgraph is defined to be $\sum_{i \in X} x(i) - \sum_{j \in Y} c(j)$.*

For instance, in Example 1, the maximal perfect set is $S^* = \{j_1\}$; the critical capacity is $\mathbf{x} = (1, 0, 1, 1)$; and the critical subgraph is given by $X = \{i_1, i_3, i_4\}$ and $Y = \{j_2, j_3\}$. In Example 2, the maximal perfect set is $S^* = B_2 \cup \{j_0\}$; and the critical subgraph is given by $X = A_1$ and $Y = B_1$, where the critical capacity of each node in X is 9. Note that in both examples, the size of the maximum matching (4 and $50 + n$, respectively) is exactly the total capacity of A (5 and $100 + n$, respectively), minus the deficiency of the critical subgraph (1 and 50, respectively); this is formalized in Corollary 2.

Properties. We first state the following fundamental lemma, which says that to prove a perfect set S, instead of showing a globally maximum matching of G as required by the definition, it suffices to find a locally maximum matching of S.

Lemma 2. *Given a bipartite graph $G = (A, B; E)$ and a subset of nodes $S \subseteq B$, if there is a maximum matching M in the subgraph $(X, \mathbf{c}_S; S)$, where $X = \{i \in A \mid c_S(i) > 0\}$, such that every node $i \in X$ is matched to $c_S(i)$ pairs, then there is a maximum matching of G containing M. Hence, S is a perfect set.*

This lemma allows us to prove that the maximal perfect set is unique, which implies that the critical subgraph and the critical capacity are uniquely defined as well (Corollary 1). Corollaries 2 and 3 below generalize Lemma 1 to multi-unit capacities, showing that the critical subgraph correctly captures the set of over-demanded vertices and the deficiency.

Corollary 1. *If $S_1 \subseteq B$ and $S_2 \subseteq B$ are two perfect subsets, then $S_1 \cup S_2$ is a perfect subset as well. Hence, there is a unique maximal perfect subset.*

Corollary 2. *Let $S \subseteq B$ be the maximal perfect set of graph $G = (A, B; E)$ and $x(i) = c(i) - c_S(i)$. Consider two subgraphs $G_1 = (X_1, \mathbf{c}_S; S)$ where $X_1 = \{i \in A \mid c_S(i) > 0\}$, and $G_2 = (X_2, \mathbf{x}; Y)$ where $X_2 = \{i \in A \mid x(i) > 0\}$ and $Y = B \setminus S$. Then any maximum matching M_1 of G_1 has size $\sum_{i \in X_1} c_S(i)$, and any maximum matching M_2 of G_2 has size $\sum_{j \in Y} c(j)$. Further, $M_1 \cup M_2$ gives a maximum matching of G with size equal to the total capacity of A minus the deficiency of the critical subgraph.*

$$\sum_{i \in X_1} c_S(i) + \sum_{j \in Y} c(j) = \sum_{i \in A} c(i) - \left(\sum_{i \in X_2} x(i) - \sum_{j \in Y} c(j) \right)$$

The above expression is similar in appearance to (though not the same as) the characterization of maximum size b-matchings (Theorem 21.4, [18]); however, our focus here is to identify the set of over-demanded women.

Corollary 3. *Let $S \subseteq B$ be the maximal perfect set and $Y = B \setminus S$. For any $j \in Y$, we have $d_G(j) > c(j)$. Further, there exists a maximum matching M where there is $i \in N_G(j)$ such that $(i, j) \notin M$ and i is under-assigned, i.e., it has fewer than $c(i)$ neighbors in M. That is, Y is precisely the set of over-demanded nodes.*

Recall that in the definition of a perfect set S, we only need to find one maximum matching in which every node $i \in A$ is matched to $c_S(i)$ pairs in S. The following claim says that if this holds for one matching, it holds for every matching — that is, if we require the set S to satisfy this requirement in *every* maximum matching, we obtain the same collection of perfect sets — so that the two definitions are equivalent.

Lemma 3. *Given a graph $G = (A, B; E)$, let $S \subseteq B$ be the maximal perfect set. Let $c_S(i) = \min\{c(i), d_S(i)\}$. Then for any maximum matching of G, each $i \in A$ is matched to exactly $c_S(i)$ neighbors in S.*

Computation. The maximal perfect set and critical subgraph can be computed by the following algorithm.

Critical-Subgraph

Given $G = (A, B; E)$, with capacities $c(k) \leq d_G(k)$ for all $k \in A \cup B$

1. Compute an arbitrary maximum many-to-many matching M in graph $G = (A, B; E)$
2. For each $i \in A$, set $x(i) = c(i) - d_M(i)$, where $d_M(i)$ is the degree of $i \in A$ in M
3. Let $X = \{i \in A \mid x(i) > 0\}$ and $Y = \emptyset$
4. While there are $i_0 \in X$ and $j_0 \in B \setminus Y$ such that edge $(i_0, j_0) \notin M$
   - add $Y \leftarrow Y \cup \{j_0\}$
   - let $X \leftarrow X \cup \{i\}$ for each edge $(i, j_0) \in M$ matched to j_0 by M
5. Return $S = B \setminus Y$ and (X, Y)

Note that the algorithm can start with an *arbitrary* maximum many-to-many matching, and recursively adds nodes into X and Y using essentially what are alternating paths with respect to the matching M (all such paths can be found in linear time using, for instance, breadth first search). Therefore, the running time of the algorithm is equivalent to finding a maximum many-to-many matching, e.g., $O(m^2 n)$ by Edmonds-Karp algorithm [4] of finding a maximum flow, where m is the number of edges and n is the number of nodes. We will show that the output of the algorithm is independent of the choice of the maximum matching M, and correctly computes the maximal perfect set, and therefore, critical subgraph.

Theorem 1. *The subset $S = B \backslash Y$ and (X, Y) returned by algorithm* Critical-Subgraph *are the maximal perfect set and critical subgraph, respectively.*

3 Algorithm for Strongly Stable Assignment

An instance of the bipartite stable many-to-many matching problem consists of two disjoint sets A (men) and B (women), where each node has a preference ranking over nodes in the other side. We give all definitions for man-nodes, *i.e.*, nodes in A; all definitions for B are symmetric. We denote the preferences of a node $i \in A$ via a preference list $\mathcal{L}(i)$ over nodes in B. The preference lists can have *ties*, *i.e.*, i need not have strict preferences over all $j \in \mathcal{L}(i)$, and can be *incomplete*, *i.e.*, $\mathcal{L}(i)$ need not include all $j \in B$. (We assume that lists $\mathcal{L}(\cdot)$ have been processed so that $i \in \mathcal{L}(j)$ if and only if $j \in \mathcal{L}(i)$.) We use $\succ_i$ and $\succeq_i$ to denote the preferences of i: if $j \succ_i j'$, then i strictly prefers j to j'; if $j \succeq_i j'$, then i weakly prefers j to j'. Each node i only wants to be assigned to nodes on his preference list $\mathcal{L}(i)$; he has a capacity $c(i)$, which is the maximum number of pairs that can be feasibly assigned to i.

Definition 2 (Strong stability). *Given a feasible many-to-many matching M, we say $(i, j) \notin M$ is a* blocking *pair for M if one of the following conditions holds:*

- *both i and j have leftover capacity in M, and belong to each others' preference lists.*
- *i has leftover capacity in M and there is $(i', j) \in M$ such that $i \succeq_j i'$; or j has leftover capacity and there is $(i, j') \in M$ such that $j \succeq_i j'$.*
- *there are $(i', j), (i, j') \in M$ such that either $j \succ_i j', i \succeq_j i'$ or $j \succeq_i j', i \succ_j i'$.*

M is strongly stable *if it does not admit a blocking pair.*

That is, a pair (i, j) blocks M if by matching with each other, at least one of them will be strictly better off and the other will not become worse off. In general, a strongly stable matching need not exist, even with unit capacities (*i.e.*, $c(\cdot) = 1$). We next give an algorithm for determining the existence of a strongly stable many-to-many matching and computing one (if it exists), based on the critical subgraph described in the previous section. For convenience, we will use many-to-many matching and matching interchangeably throughout this section.

3.1 Algorithm

The algorithm Strong-Match starts with men proposing to women at the head of their current lists, where the *head* of a man's preference list $\mathcal{L}(i)$ is the set of all women tied at the top level in $\mathcal{L}(i)$. Each proposal from i to j translates to adding an edge (i, j) to the bipartite *engagement graph* G on the sets A and B.

We say a man $i \in \mathcal{L}(j)$ is *dominated* in a woman j's preference list if $|\{i' \mid i' \succ_j i, i' \in N_G(j)\}| \geq c(j)$, *i.e.*, the number of j's neighbors in G that she strictly prefers to i exceeds her capacity. Every time a woman receives a proposal, she breaks all engagements with men who are now *dominated* in her list, *i.e.*, the edge (i, j) is removed from the engagement graph G and i and j are deleted from $\mathcal{L}(j)$ and $\mathcal{L}(i)$ respectively. Men continue proposing until their degree in the engagement graph G is greater than or equal to their capacity, or there are no women left in their preference list. This part of the algorithm is exactly like the algorithms for finding a strongly stable matching with unit capacity [7, 14]. Note, however, that because of the multi-unit capacities, a node need not be indifferent amongst its neighbors in our engagement graph G, *i.e.*, it can have neighbors in G from different levels in its preference list, which never happens in the unit capacity matching version. As the algorithm proceeds, the lists $\mathcal{L}(\cdot)$ shrink: men's neighbors get progressively worsen, and women receive progressively better proposals.

Once all men have finished making proposals, Strong-Match processes the engagement graph G to account for the fact that edges incident to a node belong to different levels. We define the sets $\mathcal{P}_G(i)$ and $\mathcal{I}_G(i)$ below: $\mathcal{P}_G(i)$ is the set of "preferred" neighbors of i that i *must* be matched to in a strongly stable matching in G (if one exists), and $\mathcal{I}_G(i)$ is the set of "indifferent" neighbors of i that i may or may not be matched to in a strongly stable matching in G.

Definition 3 **($\mathcal{P}_G(i)$, $\mathcal{I}_G(i)$ and $E_{\mathcal{P},\mathcal{P}}$, $E_{\overline{\mathcal{I},\mathcal{I}}}$).** *Given an engagement graph $G = (A, B)$ produced by* Strong-Match*, and a node $i \in A$, divide i's neighbors in G into levels $L_1, \ldots, L_m$ according to $\mathcal{L}(i)$, where i is indifferent between all nodes in the same level L_k and strictly prefers each L_k to L_{k+1}. Let $r^* = \max\{r \mid \sum_{k=1}^{r} |L_k| \leq c(i)\}$. Then $\mathcal{P}_G(i) = \{j \mid (i, j) \in G, j \in L_k, k = 1, \ldots, r^*\}$, and $\mathcal{I}_G(i) = \{j \mid (i, j) \in G, (i, j) \notin \mathcal{P}_G(i)\}$ (and similarly for $j \in B$). That is,*

- *If i has more neighbors than his capacity $c(i)$ (i.e., $d_G(i) > c(i)$), $\mathcal{P}_G(i)$ consists of the neighbors in $L_1, \ldots, L_{m-1}$ (by the rule of the algorithm, i stops proposing when his degree in G is greater than or equal to $c(i)$). $\mathcal{I}_G(i)$ consists of neighbors at level L_m.*
- *If i's degree in G is less than or equal to $c(i)$ (i.e., $d_G(i) \leq c(i)$), all his neighbors belong to $\mathcal{P}_G(i)$ (in this case, i will propose to all women in $\mathcal{L}(i)$), and $\mathcal{I}_G(i) = \emptyset$.*

Let
$$E_{\mathcal{P},\mathcal{P}} = \{(i, j) \in G \mid j \in \mathcal{P}_G(i), \text{ and } i \in \mathcal{P}_G(j)\}$$
be the set of edges such that both nodes belong to each others' $\mathcal{P}_G(\cdot)$-groups, and
$$E_{\overline{\mathcal{I},\mathcal{I}}} = \{(i, j) \in G \mid j \in \mathcal{P}_G(i), \text{ or } i \in \mathcal{P}_G(j)\}$$
be the set of edges where at least one of i or j is in the $\mathcal{P}_G(\cdot)$-group of the other.

Note that by definition, $|\mathcal{P}_G(i)| \le c(i)$ and $|\mathcal{P}_G(j)| \le c(j)$. Further, $\mathcal{P}_G(\cdot)$ can be empty (when $|L_1| > c(\cdot)$ or the node has no neighbors in G); if this occurs, all neighbors (if any) of the corresponding node are at the same level (by the algorithm).

We divide all edges in G into $(\mathcal{P},\mathcal{P})$, $(\mathcal{P},\mathcal{I})$, $(\mathcal{I},\mathcal{P})$, and $(\mathcal{I},\mathcal{I})$ types, where an edge (i,j) is, for example, a $(\mathcal{P},\mathcal{I})$ type if $j \in \mathcal{P}_G(i)$ and $i \in \mathcal{I}_G(j)$, and so on (note that these sets change through the course of the algorithm, as the engagement graph changes). If a strongly stable matching is to be found using only the edges in the current engagement graph G, it must contain all edges in the subset $E_{\overline{\mathcal{I},\mathcal{I}}}$ defined above (*i.e.*, edges where at least one endpoint strictly prefers, *i.e.*, needs to be matched to, the other), since otherwise such an edge gives a blocking pair for the matching. The algorithm therefore attempts to remove all edges in $E_{\overline{\mathcal{I},\mathcal{I}}}$, *i.e.*, the $(\mathcal{P},\mathcal{P})$, $(\mathcal{P},\mathcal{I})$, and $(\mathcal{I},\mathcal{P})$ types, from G without exceeding the capacity of any node.

By the definition of the sets $\mathcal{P}_G(\cdot)$, all $(\mathcal{P},\mathcal{P})$ edges can be removed from G without violating any node's capacity (*i.e.*, every node has adequate capacity for all edges in $E_{\mathcal{P},\mathcal{P}}$ since $|\mathcal{P}_G(i)| \le c(i)$ and $|\mathcal{P}_G(j)| \le c(j)$). Next, we proceed to $(\mathcal{P},\mathcal{I})$ edges: in Step 5(a) of the algorithm, the graph H_1 only contains $(\mathcal{P},\mathcal{I})$ edges. For every woman who does not have adequate capacity $c_{H_1}(j) = c(j) - |\mathcal{P}_G(j)|$ in H_1, for all the $(\mathcal{P},\mathcal{I})$ edges incident to her, we delete all pairs in her bottom level in G (and below) in Step 5(a), since if such a pair occurs in a strongly stable matching, it would be blocked by one of her neighbors in H_1. (The capacity of j in H_1 is defined this way to ensure that j can be matched to all her $\mathcal{P}_G(j)$ neighbors without exhausting her capacity, *i.e.*, pushing her remaining capacity below 0.)

Finally, in Step 5(b), the algorithm removes all $(\mathcal{I},\mathcal{P})$ edges to form H_2, since if such an edge cannot be included in a strongly stable matching in G, the corresponding pair blocks that matching. Note that every woman's remaining capacity is nonnegative after the removal of all these edges in $E_{\overline{\mathcal{I},\mathcal{I}}}$ (Step 5.(a) of the algorithm has already dealt with all nodes j who do not have enough leftover capacity for $(\mathcal{P},\mathcal{I})$ edges). However, a man's capacity might be smaller than the number of edges removed before forming H_2 (in this case, no strongly stable matching exists).

The graph H_2 contains only $(\mathcal{I},\mathcal{I})$ edges, so that finally all edges belong to the same level for each node. However, note that the remaining capacity of nodes in H_2 can be greater than one — thus our problem of identifying over-demanded women is still different from [7, 14], where all nodes have unit capacity in addition to having all neighbors at the same level in their preference list. Note that if all men cannot be fully matched in H_2, there is a blocking pair in every maximum matching in H_2: any under-assigned man will form a blocking pair with some neighbor in H_2 to whom he is not matched. To continue, we therefore need to identify every woman j who is over-demanded in H_2, and delete all pairs from the bottom level of each such over-demanded woman. By Corollary 3, the set of these over-demanded women is given precisely by the critical subgraph. We therefore use the algorithm CRITICAL-SUBGRAPH to identify the critical subgraph in H_2 and delete all such pairs in Step 5(b).

STRONG-MATCH

1. Set each woman $j \in B$ to be *unmarked*
2. Initialize the engagement graph $G = (A, B; E)$ where $E(G) = \emptyset$
3. While there is $i \in A$ such that $d_G(i) < c(i)$ and i has a non-empty list
   - for each $j \in B$ at the head of the list $\mathcal{L}(i)$
     - remove j from $\mathcal{L}(i)$ and add (i, j) to $E(G)$
     - if j is fully-engaged (*i.e.*, $d_G(j) \geq c(j)$), set j to be *marked*
       - for each dominated man i' on j's list, delete pair $i' \leftrightarrow j$
4. Set $b(i) = c_G(i)$, $b(j) = c_G(j)$ for $i \in A, j \in B$
5. (a) Construct graph H_1 from G by removing all edges in $E_{\mathcal{P},\mathcal{P}}$ and $\mathcal{I}_G(i)$ for each $i \in A$. If $H_1 \neq \emptyset$
   - set the capacity of each $j \in B$ in H_1 to be $c_{H_1}(j) = c(j) - |\mathcal{P}_G(j)|$
   - if there is $j \in B$ such that $d_{H_1}(j) > c_{H_1}(j)$
     - for each such $j \in B$ with $d_{H_1}(j) > c_{H_1}(j)$
       - set j to be *marked* and let $i \in N_{H_1}(j)$ be a neighbor of j in H_1
       - for each man i' with $i \succeq_j i'$, delete pair $i' \leftrightarrow j$
     - goto Step 3

   (b) Construct graph H_2 from G by removing all edges in $E_{\overline{\mathcal{I}},\overline{\mathcal{I}}}$; reducing $b(i)$ and $b(j)$ by 1 each for every removed edge (i, j) (remove all nodes with non-positive $b(i)$ and their incident edges from H_2)
   - set each node's capacity in H_2 to be $c_{H_2}(i) = b(i)$ and $c_{H_2}(j) = b(j)$
   - find the critical subgraph $X \subseteq A$ and $Y \subseteq B$ of H_2
   - if $Y \neq \emptyset$
     - for each woman $j \in Y$
       - set j to be *marked* and let $i \in N_{H_2}(j)$ be a neighbor of j in H_2
       - for each man i' with $i \succeq_j i'$, delete pair $i' \leftrightarrow j$
     - goto Step 3
6. Reset $b(i) = c_G(i), b(j) = c_G(j)$ for $i \in A, j \in B$

   (a) Construct graph G' from G by removing all edges in $E_{\overline{\mathcal{I}},\overline{\mathcal{I}}}$; set $b(i) \leftarrow b(i) - 1$ and $b(j) \leftarrow b(j) - 1$ for each removed edge $(i, j) \in E_{\overline{\mathcal{I}},\overline{\mathcal{I}}}$; let the capacity of each node in G' be $b(\cdot)$

   (b) If $b(i) \geq 0$ for all $i \in A$
   - let M' be any maximum matching in G', and let $M = M' \cup E_{\overline{\mathcal{I}},\overline{\mathcal{I}}}$
   - for each $j \in B$, if the following conditions hold
     - if j is *marked*, it is matched to $c(j)$ pairs in M
     - if j is *unmarked*, it is matched to $c_G(j)$ pairs in M

     then return M as a strongly stable matching

   (c) Else no strongly stable matching exists for the given instance

Theorem 2. *Algorithm* STRONG-MATCH *determines the existence of a strongly stable assignment and computes one (if it does) in strongly polynomial time*[6] $O(m^3 n)$.

References

1. N. Chen, X. Deng, A. Ghosh, *Competitive Equilibria in Matching Markets with Budgets*, ACM SIGecom Exchanges, V.9.1, 2010.
2. N. Chen, A. Ghosh, *Strongly Stable Assignment*, Full version available on arXiv.
3. G. Demange, D. Gale, M. Sotomayor, *Multi-Item Auctions*, Journal of Political Economy, V.94(4), 863-872, 1986.
4. J. Edmonds, R. Karp, *Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems*, Journal of the ACM, V.19(2), 248-264, 1972.
5. D. Gale, L. S. Shapley, *College Admissions and the Stability of Marriage*, American Mathematical Monthly, V.69, 9-15, 1962.
6. D. Gusfield, R. W. Irving. *The Stable Marriage Problem: Structure and Algorithms*, MIT Press, 1989.
7. R. W. Irving, *Stable Marriage and Indifference*, Discrete Applied Mathematics, V.48, 261-272, 1994.
8. R. W. Irving, D. Manlove, S. Scott, *Strong Stability in the Hospitals/Residents Problem* STACS 2003, 439-450.
9. K. Iwama, D. Manlove, S. Miyazaki, Y. Morita, *Stable Marriage with Incomplete Lists and Ties*, ICALP 1999, 443-452.
10. K. Iwama, S. Miyazaki, *Stable Marriage with Ties and Incomplete Lists*, Encyclopedia of Algorithms, 2008.
11. 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.
12. C. L. Liu, *Introduction to Combinatorial Mathematics*, McGraw-Hill, 1968.
13. V. S. Malhotra, *On the Stability of Multiple Partner Stable Marriages with Ties*, ESA 2004, 508-519.
14. D. F. Manlove, *Stable Marriage with Ties and Unacceptable Partners*, Technical Report TR-1999-29, University of Glasgow, 1999.
15. 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.
16. A. E. Roth, *Deferred Acceptance Algorithms: History, Theory, Practice, and Open Questions*, International Journal of Game Theory, 537-569, 2008.
17. A. E. Roth, M. Sotomayor, *Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis*, Cambridge University Press, 1992.
18. A. Schrijver, *Combinatorial Optimization*, Springer, 2003.
19. M. Sotomayor, *Three Remarks on the Many-to-Many Stable Matching Problem*, Mathematical Social Sciences, V.38(1), 55-70, 1999.

[6] We note that it might be possible to use the techniques in [11] (as well as faster algorithms for max-flow) to improve the runtime of our algorithm; we do not focus on optimizing the runtime in this paper.