Finding all stable pairs for the (many-to-many) Stable Matching

P. Eirinakis[1], D. Magos[2], I. Mourtos[1], P. Miliotis[1]

[1]Department of Management Science & Technology,
Athens University of Economics and Business, 76 Patission Ave., 104 34 Athens, Greece
email: {peir,mourtos,miliotis}@aueb.gr
[2]Department of Informatics, Technological Educational Institute of Athens
Ag. Spyridonos Str., 12210 Egaleo, Greece
email: dmagos@teiath.gr

Abstract

The many-to-many Stable Matching problem is defined in the context of a job market and asks for an assignment of workers (W) to firms (F) satisfying the quota of each agent and being stable, pairwise or setwise, with respect to given preference lists or relations. In this paper, we propose an algorithm identifying all stable worker-firm pairs in $O(m^2)$ steps where $m = \max\{|W|, |F|\}$. Further, we establish that this algorithm is appropriate under responsive group preferences (a) for pairwise stability and (b) for setwise stability when (in addition) workers or firms have strongly substitutable preferences. Computational results on random instances illustrate that removing non-stable pairs implies a substantial reduction in the preference lists.

1 Introduction

The *many-to-many Stable Matching* (MM) problem is naturally defined in the context of a job market in which each firm wants to hire a group of workers and each worker can be employed by several firms. Conventionally, F denotes the set of firms and W the set of workers. Let $m = \max\{|W|, |F|\}$. The *quota* q_f of a firm $f \in F$ denotes the maximum number of workers the firm can employ, while each worker $w \in W$ can be employed by at most q_w firms. A firm (worker) not fullfilling its quota is called undersubsribed. Furthermore, firms have preferences over individual workers and vice versa. These preferences are assumed to be strict and transitive, thus representable by ordered lists, called *preference lists.* We denote by $P(w)$ ($P(f)$) the preference list of worker w (firm f). For $f \in F$ ($w \in W$), the event that worker w_1 (firm f_1) ranks higher in $P(f)$ ($P(w)$) than worker w_2 (firm f_2) is denoted by $w_1 \succ_f w_2$ ($f_1 \succ_w f_2$). Preference lists need not be complete; if a firm f (worker w) is not in worker w's (firm f's) preference list, then f (w) prefers to leave a workspace empty than to employ w (f). The simplest form of the MM problem asks for a maximal set $M \subseteq W \times F$ satisfying the following two conditions:

(i) for no pair $(w, f) \in M$ there exist $\bar{w} \in W$ and $\bar{f} \in F$ such that $\bar{w}$ is more preferable than w by f *and* $\bar{f}$ is more preferable than f by w (stability condition),

(ii) the quota of each agent (i.e., worker/firm) is not exceeded (matching condition).

The MM problem is a generalization of the *Stable Admissions* (SA) problem which is, in turn, a generalization of the *Stable Marriage* (SM) problem. In the latter (former) case, also called the *one-to-one (one-to many)* Stable Matching problem, the quotas of the agents of both (one of the) sets are set to one. Both of these problems were introduced in a study of the mechanics of assigning students to colleges [7]. Besides the fact that MM is a proper generalization of SA (and SM), there are several real-life applications motivating its study. Primarily, it models several centralized markets, an example being the well-known medical interns' market in the U.K. [18], where each student must seek two positions, one in medicine and one in surgery. Also, most labor markets include certain many-to-many interacting agents, which are examined in terms of the MM model, since their study in terms of the (simpler) one-to-many framework produces significantly different results [5, Example 2].

The generalization of preferences and the stability condition, introduced in the literature, make the MM a more involved structure than SM. Under this setting, the preference lists are also generalized into *preference relations;* these represent the preferences of each firm f on groups of workers and vice versa. Thus in the MM case, the stability condition can be extended from pairs to *sets (groups)* of agents. That is, a matching M is called *setwise* (or *group*) stable if there is no subset of agents who by forming new partnerships only among themselves, possibly dissolving some partnerships of M to remain within their quotas and possibly keeping other ones, can all obtain a strictly preferred set of partners [19]. Note that preference relations include individual preferences as a group may be a singleton.

We examine three main types of preference relations that have been reported in the literature: *responsive* preferences [18], *substitutable* preferences [13] and *strongly substitutable* preferences [5]. To describe these terms, let $P^{\#}(f)$ denote the preference relation of the firm f and $M(f)$ the set of workers assigned to f in the matching M.

Definition 1 *The preference relation $P^{\#}(f)$ over sets of workers is responsive (to the preferences $P(f)$ over individual workers) if, whenever $M'(f) = M(f) \cup \{w_1\} \setminus \{w_2\}$ for $w_2 \in M(f)$ and $w_1 \notin M(f)$, the firm f prefers $M'(f)$ to $M(f)$ (denoted by $M'(f) \succ_f M(f)$) if and only if $w_1 \succ_f w_2$.*

The substitutable preferences comprise a weaker criterion. (In fact, it is the weakest condition under which the existence of stable matchings is guaranteed, according to [16].) Suppose that a firm f prefers a group of workers $\mathfrak{w}$ including a worker w. Then the firm has substitutable preferences, if, whenever some members of $\mathfrak{w}$ become unavailable, it still wants to employ a subgroup of $\mathfrak{w}$ that includes w (see [19] for a formal definition). In [5], a variant of the notion of substitutability, called *strong substitutability,* is introduced. That is, assume that hiring worker w is optimal when certain workers are available. Then, strong substitutability requires that hiring w must still be optimal even when a worse set of workers is available. Note that responsiveness implies substitutability but not strong substitutability. Figure 1 presents the relationship between responsive, substitutable and strong substitutable preferences (see examples in [5]).

The preference relations, described above, imply a partial ordering of stable matchings for each agent. This ordering depicts the preference of the agent over the matchings. In the case of individual preferences which imply *pairwise stability* (see (i) above) the standard criterion used is the so called *maxmin* criterion [2]. According to that criterion, a group of workers (firms) is preferred by a firm f (worker w) to some other group if the least preferred worker (firm) of the first group ranks higher in the preference list of firm f (worker w) to the least preferred worker (firm) of the other group. Hence, under the maxmin criterion, each agent has always a complete ordering of all stable matchings. A formal definition follows.

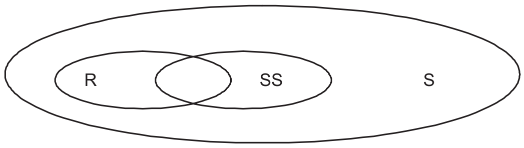

Figure 1: Relationship between responsive, substitutable and strong substitutable preferences

Definition 2 *Let M and M' be two arbitrary matchings. The firm $f \in F$ prefers M at least as M' (denoted as $M \succeq_f M'$) if either (i) $M(f) = M'(f)$ or (ii) $|M(f)| \geq |M'(f)|$ and* $\text{last}_M(f) \succ_f \text{last}_{M'}(f)$, *where* $\text{last}_M(f)$ $(\text{last}_{M'}(f))$ *denotes the least preferred worker of the group $M(f)$ $(M'(f))$ by f.*

A pair (w, f) is called *stable* if w is assigned to f in at least one stable matching. The problem studied here is the ALLSTABLEPAIRS problem, i.e. the problem of identifying all stable pairs. The usefulness of this information was first observed in [14], while a polynomial algorithm for solving this problem in the SM setting is described in [9]. In [6], an algorithm for identifying all stable pairs for the SA with individual preferences is presented.

In the current work, we present an algorithm for solving the ALLSTABLEPAIRS problem in the MM setting under individual preferences. In the sequel, we show that this algorithm can also be used in the presence of responsive group preferences under (a) pairwise stability, or, (b) setwise stability when (in addition to responsiveness) the agents of one of the sets (W or F) have strongly substitutable preferences. To show (a), we prove that the *lattice* of stable matchings formed when agents have individual preferences coincides with the corresponding object when preferences are responsive. This settles a question posed in [2] regarding the efficiency of the two criteria (i.e., maxmin versus responsive) when pairwise stability is considered.

An alternative definition to the ALLSTABLEPAIRS problem can be given in the context of Constraint Programming (CP). Thus, if we consider the MM problem as a *global* constraint, the ALLSTABLEPAIRS problem asks for establishing *generalized arc consistency* to that constraint (see [11] for related definitions). Such a view has been adopted in [8] and [15] for the SM and SA case, respectively. However, the main theme in all these papers is the presentation and comparison of different encodings, none of which directly refers to the MM version of the problem. Further the issue of consistency is only partially resolved, since the algorithms presented just establish arc consistency without identifying all (non-)stable pairs.

2 Background

Under pairwise stability, if a matching M is not stable then there exists a worker w and a firm f such that $(w, f) \notin M$ but both w and f prefer each other to their current partners in M [10, 18]. Such a pair is said to *block* the matching M, or, equivalently, (w, f) forms a *blocking pair.* Similarly, a matching M can be blocked by an individual agent if this agent is matched to a member of the other set not appearing in its preference list (i.e., he prefers to remain single than to obtain a partner not in its preference list). In the case of setwise stability, a matching M can be blocked by a coalition bigger than a simple worker-firm pair [18].

In the case of pairwise stability under the maxmin criterion [2] or under responsive preferences [1], the MM problem has a non-empty solution set, namely $\mathfrak{M}$. Among the members of $\mathfrak{M}$, there exists a matching in which every worker is at least as better-off (worse-off) under it as under any other matching. Such a matching is called *workers' optimal (pessimal).* In fact, a *worker-oriented dominance relation* can be defined on the set of stable matchings.

Remark 3 *For $M, M' \in \mathfrak{M}$ and $M \neq M'$,*

(i) M dominates M' ($M \succ_W M'$) if $M \succeq_w M'$ for all $w \in W$ and

(ii) $M \succ_w M'$ implies $M' \succ_f M$ for all workers $w \in W$ and firms $f \in F$ such that $(w, f) \in (M \setminus M') \cup (M' \setminus M)$ (and vice versa).

Also, the *join (supremum)* $M \vee M'$ and the *meet (infimum)* $M \wedge M'$ are stable matchings, where $M \vee M'$ ($M \wedge M'$) assigns to each worker w the best (worst) of the two sets of firms $M(w)$ and $M'(w)$, and to each firm f the worst (best) of the workers $M(f)$ and $M'(f)$. Under this dominance relation, it can be shown that the set of stable matchings forms a *distributive lattice* (for definitions see [1, 2]). The *greatest* element of the lattice corresponds to the workers' optimal (firms' pessimal) matching and the *least* element corresponds to the worker's pessimal (firms' optimal) matching.

The situation is more complicated when setwise stability is considered. In the simpler SA setting, a matching is setwise stable if and only if it is pairwise stable. This is not true in the case of MM. In [18] it is shown that, in many-to-many matching models with responsive (thus substitutable) preferences, a pairwise stable matching need not be setwise stable. Also in [4] it is observed that the lattices that arise in many-to-many matching markets under setwise stability need no longer be distributive. On the positive side, in [5] it is proved that the set of stable matchings, under setwise stability, forms a non-empty lattice identical to the one created under the pairwise stability when members of the one (other) set have (strong) substitutable preferences. That particular model actually encompasses standard one-to-many theory, in which one of the sets represents colleges and the other students, since it is trivially true that colleges (students) have (strong) substitutable preferences. In the same work, it is shown (contrary to [4, Example 5.2]) that the lattice formed by the set of stable matchings is distributive in the case of the preferences being strong substitutable [5].

Finding a stable matching is intuitively related to the identification of stable pairs. In [10], an extended implementation of the Gale/Shapley algorithm [7] (called EGS) is described that in the process of findin a stable solution, identifies and eliminates some (but not all) non-stable pairs. The *worker-oriented* version of that method (hereafter called WEGS) in the MM setting identifies the workers' optimal solution under pairwise stability with individual preference lists in $O(m^2)$ steps [2]. Upon termination of WEGS, the workers' preference lists are reduced in a way that (a) the worker w is assigned either its best q_w stable partners, which are the first q_w firms in her (reduced) list, or a set of fewer than q_w firms, and (b) each firm is assigned its worst set of 'stable' workers. Note that it is not necessarily true that, in the workers' optimal (i.e. firms' pessimal) solution, a firm that fills all of its q_f places is assigned its q_f worst partners (of course, this is trivially true if a worker is underemployed). However, if M_0 is the workers' optimal matching and M' any other stable matching, then every firm prefers all the workers assigned to it in M' to all of those assigned to it in M_0 but not in M'. Also, in [1, 2] it is observed that (a) each worker is employed in the same number of firms in all stable matchings, (b) exactly the same workers are underemployed (or unemployed) in all stable solutions, and (c) any worker who is underemployed in one stable matching is matched with precisely the same firms in all stable matchings.

As stated previously, even if we successively apply WEGS, the resulting reduced lists may still allow for non-stable pairs to appear. It turns out that the ALLSTABLEPAIRS problem can be solved by exploiting the lattice structure of $\mathfrak{M}$, as discussed next.

3 Individual preferences

Assume that the agents have individual preferences. The implications are that (a) stability is defined in a pairwise context and (b) the stable matchings can be ordered (to form a distributive lattice) with respect to the maxmin criterion. Given a stable matching M, let $r_M(w)$ be the first firm $f \in P(w)$ such that $(w, f) \notin M$ and $w \succ_f \text{last}_M(f)$, i.e. firm f is not assigned to worker w but prefers it to its least preferred worker. Note that such a firm exists as long as M is not the firm-optimal matching. Further, $\text{next}_M(w)$ denotes $\text{last}_M(r_M(w))$. Since M is stable, w prefers all firms assigned to him in M to $r_M(w)$ if such an $r_M(w)$ exists (for example in firms' optimal matching $r_M(w)$ does not exist for some $w \in W$). Next, we explore the possibility of assigning the worker w to $f = r_M(w)$ not knowing yet which of his/hers current assignments to break to be employed by f. Next, firm f (which prefers w to $\text{next}_M(w)$) fires $\text{next}_M(w)$ and employes w. Worker $\text{next}_M(w)$ (who is now underemployed) proposes to $r_M(\text{next}_M(w))$ and so on, until a worker w' (who may or may not be w) comes up twice. Let $\rho = (w_1, r_M(w_n)), (w_2, r_M(w_1)), \dots, (w_n, r_M(w_{n-1}))$ be an ordered list of pairs in a stable matching M such that $w_{i+1 \pmod n} = \text{next}_M(w_i)$, for all $i \in \{1, \dots, n\}$. Then ρ is a *(meta-)rotation exposed* in M, and we say that w (or f) is *in rotation* ρ if there is a pair (w, f) in the ordered list defining ρ. If $\rho = (w_1, f_1), (w_2, f_2), \dots, (w_n, f_n)$ is a rotation exposed in M, then M/ρ is called the *elimination* of rotation ρ from M and it denotes the matching M' derived from M if each worker w_i participating in ρ breaks her assignment to f_i and is employed by $f_{i+1(\text{mod}\, n)} = r_M(w_i)$, while everyone else is employed by exactly the same firms as in M.

Rotations were introduced in [12] and used in [9] to develop an algorithm for solving the ALLSTABLEPAIRS problem in the SM setting. Rotations have also been used in the SA [6] and the MM context [3]. To employ rotations for solving the problem in the MM case, we present a series of statements, some of which extend known results and are therefore provided without a proof.

Lemma 4 *When a rotation ρ is eliminated from a stable matching M, the resulting matching M/ρ is stable and is dominated by M.*

Corollary 5 *Every worker-firm pair produced by eliminating a rotation is stable.*

The following result [9, Lemma 3] extends in the MM case.

Lemma 6 *Let ρ denote a rotation exposed in the stable matching M. Then, there is no stable matching 'between' $M, M' = M/\rho$ (i.e., M, M' correspond to adjacent points in the lattice).*

Lemma 7 *Let $\rho = (w_1, f_1), \dots, (w_n, f_n)$ be a rotation exposed in the stable matching M and, for some $i \in \{1, \dots, n\}$, let $f \in P(w_i)$ ($w \in P(f_i)$) such that* $\text{last}_M(w_i) \succ_{w_i} f \succ_{w_i} \text{last}_{M/\rho}(w_i)$ $(\text{last}_{M/\rho}(f_i) \succ_{f_i} w \succ_{f_i} \text{last}_M(f_i))$. *Then (w_i, f) (respectively, (w, f_i)) is not a stable pair.*

Proof. We illustrate the result only with respect to the reference lists of workers (i.e. for pair (w_i, f)). Observe that, in w_i's list, f ranks lower than f_i and higher than $f_{i+1(\text{mod}\, n)}$, where $f_i = \text{last}_M(w_i)$ and $f_{i+1(\text{mod}\, n)} = r_M(w_i) = last_{M/\rho}(w_i)$. The procedure for exposing a rotation implies that $r_M(w_i)$ is the first firm ranking lower than $last_M(w_i)$ in w_i's list, which prefers w_i

to its least preferred worker in M. Since f ranks higher than $r_M(w_i)$, f prefers $last_M(f)$ to w_i (if not, it would itself be $r_M(w_i)$), i.e. $last_M(f) \succ_f w_i$.

Now suppose (w_i, f) is a stable pair in some stable matching M'. That implies that w_i is better-off in M, since it prefers f_i to f. However, so is f, since it prefers $last_M(f)$ to $last_{M'}(f)$, where $last_{M'}(f)$ may only be w_i or some worker ranking lower than w_i in f's list. It follows that $M \succ_{w_i} M'$ and $M \succ_f M'$, which contradicts Remark 3(ii). ■

Let hereafter M_0 (M_z) denote the workers' optimal (pessimal) stable matchings, respectively. The following theorem, generalizes [17, Theorem 4].

Theorem 8 *Any stable matching can be obtained from the workers' optimal by successively exposing and eliminating rotations.*

Proof. Let $M \neq M_0$. Since M is non-worker-optimal, there exists some worker w such that $M_0(w) \succ_w M(w)$, i.e. $last_{M_0}(w) \succ_w last_M(w)$. Thus, consider a rotation ρ exposed in M_0 that includes w. Within the procedure of exposing ρ, assume that worker w proposes to a choice worse than $last_M(w)$. That implies that w is rejected by $last_M(w)$ at some point during the procedure of exposing ρ. Equivalently, $last_M(w)$ is 'between' $last_{M_0}(w)$ and $r_{M_0}(w)$, i.e. $last_{M_0}(w) \succ_w last_M(w) \succ_w r_{M_0}(w)$. But then, Lemma 7 yields pair $(w, last_M(w))$ as non-stable, which contradicts the assumption that matching M is stable.

Therefore, exposing a rotation in M_0 can only produce a stable matching M', in which no worker has a worse choice than in M, while the procedure can be repeated if $M' \neq M$. Since, after exposing a rotation, at least two workers are employed by a worse choice, but not a choice worse than their choice in M, matching M is obtainable after eliminating a finite number of rotations. ■

Corollary 9 *Starting from the worker-optimal stable matching, the firm-optimal stable matching can be obtained by a number of successive exposures and eliminations of rotations.*

Theorem 10 *Let $M_0, M_1, \ldots, M_z$ be a sequence of stable matchings such that $M_i \succ_W M_{i+1}$ (M_i dominates M_{i+1}) for $i = 0, \ldots, z-1$, and there is no stable matching 'between' M_i and M_{i+1}. Then, every stable pair appears in at least one of the stable matchings of this sequence.*

Proof. For $w \in W$ and $f \in F$, let (w, f) be a stable pair appearing in no matching $M_i, i = 0, \ldots, z$. It follows that $f \neq last_{M_i}(w) = f_i$ for all $i = 0, \ldots, z$. Since $M_i \succ_W M_{i+1}$ for $i = 0, \ldots, z-1$, either $f_i = f_{i+1}$ or $f_i \succ_w f_{i+1}$. Thus, there exist two matchings in the sequence, namely M_i and M_{i+1} such that $f_i \succ_w f \succ_w f_{i+1}$.

Clearly, (w, f) appears in some other stable matching M, i.e. $M \neq M_i$ for $i = 0, \ldots, z$. Since the set of stable matchings forms a distributive lattice, we may construct matching $M' = M_i \wedge (M \vee M_{i+1})$. Observe that M' is a stable matching in which w is assigned to f, thus being different from both M_i and M_{i+1}, with $last_{M_i}(w) \succ_w last_{M'}(w) \succ_w last_{M_{i+1}}(w)$. Under the maxmin criterion, that implies $M_i \succ_W M' \succ_W M_{i+1}$, i.e. a contradiction to the hypothesis that there is no stable matching 'between' M_i and M_{i+1}. ■

Note that the underemployed workers (unsubscribed firms) participate in no rotations, since they are employed by precisely the same set of firms (workers) in all stable matchings. The following is an extension of [12, Lemma 4.7] (see [5] for a proof).

Lemma 11 *In any MM instance, no (w, f) pair can belong to two different rotations.*

Let us denote as FEGS the firm-oriented version of EGS that identifies the firms' optimal solution under pairwise stability. Algorithm 1 solves the ALLSTABLEPAIRS problem in the

Algorithm 1 Finding all stable pairs

```
Initially, the status of all pairs (w, f) is 'undefined';
Run WEGS and FEGS to produce reduced preference lists and M_0, M_z;
Use the reduced preference list to update the status of pairs (w, f);
i ← 0;
while ∃(w, f) with undefined status do
  Expose ρ_i in M_i;
  M_{i+1} ← M_i/ρ_i;
  Set the status of pairs (w, f) ∈ M_{i+1} to 'stable';
  for all f ∈ ρ_i do
    for all w such that last_{M_i/ρ_i}(f) ≻_f w ≻_f last_{M_i}(f) do
      Set the status of pair (w, f) to 'non-stable';
    end for
  end for
  for all w ∈ ρ_i do
    for all f such that last_{M_i/ρ_i}(w) ≻_w f ≻_w last_{M_i}(w) do
      Set the status of pair (w, f) to 'non-stable';
    end for
  end for
  Update preference lists;
end while
Output preference lists;
```

MM setting when agents have individual preferences. The algorithm assigns to each pair $(w, f) \in W \times F$ a status flag which can receive the values 'stable', 'non-stable', or 'undefined'. Initially, the status of all pairs is set to 'undefined'. The execution of WEGS and FEGS causes some of the pairs to obtain the status 'stable' (i.e. those appearing at M_0 and M_z) and some the status 'non-stable'. Starting from M_0, the algorithm exposes and eliminates rotations. Each time a new stable matching is reached, the pairs belonging to it receive the status 'stable' whereas the pairs indicated by Lemma 7 have their status set to 'non-stable' . The algorithm terminates when no pair remains with an 'undefined' status. Each time a pair (w, f) is found to be non-stable w, f are deleted from each other's list.

Theorem 12 *Algorithm 1 solves the ALLSTABLEPAIRS problem in* $O(m^2)$ *steps.*

Proof. Matching M_0 (M_z), together with the reduced preference lists is produced by WEGS (FEGS). As stated above, the algorithm starts exposing and eliminating rotations, thus creating a sequence of stable matchings $M_0, M_1, ..., M_z$ (Lemma 4). By Lemma 6, there is no stable matching 'between' M_i and M_{i+1} for all $i \in \{0, ..., z-1\}$. Since every stable pair is bound to appear in at least one of the matchings of this sequence (Theorem 10), Algorithm 1 reveals all stable pairs upon completion. (Note that the algorithm may terminate before deriving M_z as long as the status of all pairs has been defined.) The preference list of each agent, once Algorithm 1 terminates, contains only the mates that appear together with him in at least one stable matching. Thus, Algorithm 1 is correct.

Concerning the complexity of this algorithm, recall that WEGS and FEGS require $O(m^2)$ steps to create M_0 [2]. Exposing and eliminating a rotation can easily be implemented at constant time per pair by employing a stack (see [10, Section 4.2.4]). Since each pair appears in at most one rotation (Lemma 11) and the number of pairs is $O(m^2)$, Algorithm 1 requires $O(m^2)$

steps for exposing and eliminating all rotations, i.e. for computing matchings $M_1, ..., M_{z'}$, $z' \leq z$. ■

Since each pair should be examined at least once in order to determine its stability, Algorithm 1 is of the lowest possible complexity, i.e. it is asymptotically optimal.

4 Group preferences

In the setting of group preferences, the ALLSTABLEPAIRS problem amounts to computing all stable assignments of each agent, i.e. all sets of firms (workers) assigned to each worker (firm) in any stable matching. First, we show that this can be accomplished by Algorithm 1 under the maxmin criterion and for responsive group preferences.

Let L denote the distributive lattice corresponding to an MM instance. A sequence of stable matchings $P = M_0, M_1, \ldots, M_z$ is called a *maximal path of* L, if $M_{i+1} = M_i/\rho_i$ where ρ_i is a rotation exposed in matching $M_i, i = 0, ..., z-1$. Lemma 6 implies that there exists no stable matching 'between' M_i and M_{i+1}, i.e. the notion of a *maximal path* is analogous to that of a *maximal chain* of a general ring of sets [10]. Denote also as $M(w)$ the assignment to worker w in matching M, i.e. $M(w) = \{f : (w, f) \in M\}$.

Theorem 13 *Every rotation appears exactly once on every maximal path P of lattice L.*

Proof. It suffices to establish that every rotation is exposed at least once in P. For a rotation ρ not exposed while deriving P, Lemma 11 implies that a pair (w, f) appearing in ρ appears in no other rotation exposed while deriving P. But then, pair (w, f) is non-stable (Theorem 10), i.e. a contradiction to Corollary 5. ■

An implication of the previous theorem is that the all maximal paths have the same length $z + 1$, where z is the number of rotations exposed in all stable matchings.

Theorem 14 *A complete order of all stable assignments is obtained for worker w (firm f) by listing all stable assignments given to w (f) during the procedure of deriving a maximal path P.*

Proof. We show the result with respect to workers. Assume that assignment $M_{i+1}(w)$ is not included in worker w's list after exposing and eliminating all rotations. Let also $M_{i+1} = M_i/\rho_i$, pair (w, f) be appearing in rotation ρ_i and $f' = r_{M_i}(w)$. By Lemma 11, pair (w, f) appears in at most one rotation that is ρ_i, while Theorem 13 implies that ρ_i is exposed in some matching M_k appearing on P.

If $M_i = M_k$, $M_k/\rho_i = M_{i+1}$ thus $M_{i+1}(w)$ is actually included in worker w's list, i.e. a contradiction to the hypothesis. Otherwise ($M_i \neq M_k$), let $M_{k+1} = M_k/\rho_i$ thus yielding $last_{M_{i+1}}(w) = last_{M_{k+1}}(w) = f'$. For $M_{k+1}(w) = M_{i+1}(w)$, $M_{k+1}(w)$ is included in worker w's list since M_{k+1} belongs to P, i.e. a contradiction. For $M_{k+1}(w) \neq M_{i+1}(w)$, observe that $M_{k+1}(w)$ and $M_{i+1}(w)$ cannot be compared under the maxmin criterion since $last_{M_{k+1}}(w) = last_{M_{i+1}}(w)$, therefore one of them is not stable. It is not difficult to see that both cases yield a contradiction. ■

Evidently, Algorithm 1 can compute a maximal path if it is set to terminate only after obtaining the firms' optimal matching M_z. Therefore, this algorithm finds all stable assignments for all agents participating in any stable matching under the maxmin criterion, simply by retaining the workers (firms) assigned to each firm (worker) at each of the matchings $M_0, M_1, \ldots, M_z$. Moreover, all assignments of each worker (firm) are completely ranked in the (inverse) order they are identified. Hence the following.

Theorem 15 *Algorithm 1 computes all stable assignments of every agent for every instance of MM in $O(m^2)$ steps.*

Now, let us examine the case where group preferences are responsive to individual preferences and the stability criterion involves only pairs. The following theorem extends [18, Theorem 5.27].

Theorem 16 *Let M, M' be two stable matchings. If $M(w) \succ_w M'(w)$ for some worker w, then $f \succ_w f'$ for all $f \in M(w)$ and $f' \in M'(w) \backslash M(w)$.*

Theorem 16 together with Definition 1 lead to the following.

Corollary 17 *Let $P^{\#}(w)$ denote the preference relation of worker w over groups of firms which is responsive to his preferences $P(w)$ over individual firms. Then for every pair of stable matchings M and M', $M(w) \succ_w M'(w)$ under $P^{\#}(w)$ if and only if $M(w) \succ_w M'(w)$ under $P(w)$.*

Thus, Corollary 17 shows that the set of stable matchings is invariant to changes in the preference relations $P^{\#}(w)$ as long as these preferences remain responsive to the preferences $P(w)$ over individuals. A question posed in [2] is whether the maxmin criterion is significantly different in practice from responsive preferences, under pairwise stability. The following theorem answers this question in the negative. Recall that $\mathfrak{M}$ denotes the set of solutions of an MM instance.

Theorem 18 *The distributive lattice formed by $\mathfrak{M}$ under responsive preferences with pairwise stability is identical to the one formed by $\mathfrak{M}$ under the maxmin criterion.*

Proof. Consider an MM instance with preference lists $\mathbb{P}$ and recall that the matchings comprising $\mathfrak{M}$ form a distributive lattice L, under the maxmin criterion with pairwise stability [1]. Consider another MM instance, along with its associated set of solutions $\mathfrak{M}'$, having the exact same agents as $\mathfrak{M}$ but responsive preference relations $\mathbb{P}^{\#}$ that include the corresponding lists $\mathbb{P}$. We may safely assume that each agent's preference list contains the complete order of all stable assignments of that particular agent in $\mathfrak{M}'$ (computed by Algorithm 1), followed by the individual preference list $\mathbb{P}$. Then, observe that Corollary 17 implies a dominance relation for $\mathfrak{M}'$ identical to the one defined for $\mathfrak{M}$ (Remark 3), while the supremum and the infimum of any two $M, M' \in \mathfrak{M}'$ can also be defined as in the case of $\mathfrak{M}$. In other words, $\mathfrak{M}'$ is identical to $\mathfrak{M}$ and so are the corresponding lattices. ■

By Theorem 18, it becomes evident that pairwise stable matchings can be identified in the case of responsive group preferences without the complete knowledge of them, i.e. only the preferences over individuals are required. Note that, although the number of assignments included in a preference relation $P^{\#}$ can be exponential in m, Theorem 15 implies that the number of stable assignments is bounded by the number of rotations, which is trivially bounded by m^2. Therefore, the ALLSTABLEPAIRS problem can be solved using Algorithm 1 in the presence of responsive preference relations, under pairwise stability.

In the case of setwise stability, the question of finding all stable assignments becomes far more complex. However, a special case remains tractable through Algorithm 1, again because the set of stable matchings $\mathfrak{M}$ forms a distributive lattice. In [5], it is shown that if the workers' preferences are substitutable and the firms' preferences are strong substitutable (or vice-versa), then $\mathfrak{M}$ is the same under both setwise and pairwise stability, and forms a lattice. Since responsive preferences are also substitutable (Figure 1), it is easy to derive the following through Theorem 18.

Theorem 19 *The distributive lattice formed by $\mathfrak{M}$ under setwise stability, when both workers and firms have responsive preferences and at least one set of agents has also strong substitutable preferences, is identical to the one formed by $\mathfrak{M}$ under the maxmin criterion.*

It follows easily that, in the case of setwise stability implied by Theorem 19, Algorithm 1 computes all stable assignments in $O(m^2)$ steps.

5 Computational Results

This section discusses the application of Algorithm 1 to a set of randomly generated instances of MM. The code is written in VB.NET in the Microsoft Visual Studio .NET Professional 2003, while computational results are obtained on an Intel(R) Core(TM) 2 Duo CPU T7500 processor (2.20 GHz) with 2 GB of RAM under Windows Vista(TM).

The results are summarized in Table 1. Each row in this table corresponds to a different MM problem and presents the average results obtained on 10 randomly generated instances. Workers and firms have complete preference lists, i.e. each of the $|W| \cdot |F|$ pairs appears in two preference lists. Further, the quota of each worker (firm) is a random integer drawn uniformly from the interval $\{1, \ldots \max q_w\}$ ($\{1, \ldots \max q_f\}$). Those instance features are depicted in the first four columns of Table 1. The next two columns illustrate the time (in seconds) required by Algorithm 1 and the percentage of pairs that are deleted, respectively. The following two columns illustrate the percentage of time consumed and the percentage of pairs deleted, respectively, by the procedure of exposing and eliminating rotations (i.e. the remaining percentage of time and pairs deleted concern the WEGS and FEGS algorithms that comprise the first step of Algorithm 1). Thus, these two columns indicate the additional reduction achieved through Algorithm 1 (i.e. the percentage of non-stable pairs not identified by WEGS or FEGS) and the associated computational effort for finding all rotations. The penultimate column presents the number of rotations identified, while the last column provides the average number of stable assignments per worker, i.e. the average number of subsets of F that would not be deleted in any preference relation satisfying the criteria of Theorems 18 or 19.

Observe first that the reduction in the preference lists is substantial in all instances. This illustrates the importance of the proposed algorithm in terms of reducing the solution space. Thus, Algorithm 1, as a preprocessing step, could significantly enhance the performance of an algorithm searching for an ‘optimal’ solution or enumerating all solutions. Moreover, this finding provides an insight on the structure of MM, given that no theoretical bound on the number of stable pairs is known. Notice also that the reduction in the preference lists remains approximately the same for different values of $|F|$ and $|W|$ but decreases as quotas increase, since larger quotas imply more solutions and therefore fewer non-stable pairs.

Most of this reduction is achieved by using WEGS and FEGS, which also consume the larger percentage of total time. Still, the remainder of Algorithm 1 that utilizes the concept of rotations has a significant impact, with the percentage of non-stable pairs identified through exposing rotations (i.e. not found WEGS or FEGS) being ‘proportional’ to the percentage of the time required. For example, in the instance where $(|F|, |W|, \max q_f, \max q_w) = (100, 1000, 200, 20)$, 66% of a total of 10^5 pairs is deleted within less than a second; among those pairs, 90% is removed by using WEGS or FEGS in 76% of total time, while 10% is removed by the procedure of exposing rotations in 24% of total time. The ‘time percentage over pair percentage’ ratio (related to exposing rotations) actually increases in large instances (e.g. for $|W| = |F| = 2000$), although the number of rotations increases drastically, especially for larger quotas.

Algorithm 1 performs equally well in the presence of group preferences, since the number of stable assignments per worker (last column of Table 1) is always smaller than 20% of m.

One can easily construct responsive preference relations $\mathbb{P}^{\#}$, where the total number of subsets of firms contained in all workers' relations is $O\left(|W| \cdot \binom{|F|}{\max q_w}\right)$. It follows that Algorithm 1 can lead to an enormous reduction of preference relations when Theorems 18 or 19 become applicable.

Overall, the current work, apart from providing an efficient algorithm for the ALLSTABLE-PAIRS problem in the MM setting, has important computational implications under both individual and group preferences. Furthermore, Algorithm 1 is applicable for any type of preference relations that gives rise to a distributive lattice identical to the one formed under the maxmin criterion. Whether more such types exist remains to be further investigated.

Table 1: Computational results

Instance				Algorithm 1		Exposing Rotations			
|F|	|W|	maxq$_f$	maxq$_w$	Time	Pairs(%)	Time(%)	Pairs(%)	#Rot.	#Assign.
100	100	5	5	<1	90	14	6	90	7
100	100	20	20	<1	67	20	8	199	14
100	100	50	50	<1	40	11	3	155	11
100	500	25	5	<1	88	16	8	402	8
100	500	100	20	<1	67	22	9	825	14
100	500	250	50	<1	39	12	4	708	12
100	1000	50	5	<1	88	17	9	802	8
100	1000	200	20	<1	66	24	10	1643	15
100	1000	500	50	<1	39	12	9	1324	12
100	2000	100	5	1	88	17	9	1546	8
100	2000	400	20	3	67	23	9	3110	15
100	2000	1000	50	4	39	12	4	2566	12
500	500	25	25	1	87	27	15	1538	43
500	500	100	100	2	64	30	14	3011	82
500	500	250	250	2	36	17	7	2707	73
500	1000	50	25	4	86	30	18	2873	45
500	1000	200	100	5	63	33	17	5631	88
500	1000	500	250	7	35	19	8	4973	78
500	2000	100	25	13	86	32	19	5472	47
500	2000	400	100	19	63	33	17	10406	88
500	2000	1000	250	24	36	17	7	8581	73
1000	1000	50	50	11	86	34	22	5026	95
1000	1000	200	200	15	62	34	18	9509	179
1000	1000	500	500	22	35	19	8	8307	156
2000	2000	100	100	98	85	36	25	15038	197
2000	2000	400	400	138	61	37	21	29399	382
2000	2000	1000	1000	180	34	20	9	25325	327

References

[1] Alkan A.: On the properties of stable many-to-many matchings under responsive preferences, *Current trends in economics: theory and applications* **8**, edited by A. Alkan, C.D. Aliprantis, N.C. Yianellis, Studies in Economic Theory, Springer Verlag, (1999).

[2] Baiou M., Balinski M.: Many-to-Many Matching: stable polyandrous polygamy (or polygamous polyandry). *Discrete Applied Mathematics* **101**, 1-12 (2000).

[3] Bansal V., Agrawal A., Malhotra V.S.: Polynomial time algorithm for an optimal stable assignment with multiple partners". *Theoretical Computer Science* **379 (3)**, 317-328 (2007).

[4] Blair C.: The lattice structure of the set of stable matchings with multiple partners, *Mathematics of Operations Research* **13** (4), 619-628 (1988)

[5] Echenique F., Oviedo J.: A theory of stability in Many-to-Many matching markets, *Theoretical Economics* **1**, 233-273 (2006).

[6] Eirinakis P., Magos D., Mourtos I., Miliotis P.: Hyperarc Consistency for the Stable Admissions problem, In *Proceedings of ICTAI'07,* **1,** 239-242 (2007).

[7] Gale D., Shapley L.S.: College Admissions and the Stability of Marriage. *The American Mathematical Monthly* **69**, 9-15 (1962).

[8] Gent I.P., Irving R.W., Manlove D.F., Prosser P., Smith B.M.: A constraint programming approach to the stable marriage problem. In *Proceedings of CP'01*, *Lecture Notes in Computer Science* **239**, 225 – 239, Springer-Verlag (2001).

[9] Gusfield D.: Three fast algorithms for four problems in stable marriage. *SIAM Journal on Computing* **16**, 111-128 (1987).

[10] Gusfield D., Irving R.W.: The stable marriage problem: Structure and Algorithms. *MIT Press*, Cambridge, MA (1989).

[11] Hooker, J.N.: Logic Based Methods for Optimization, *Wiley*, NY (2000).

[12] Irving R., Leather P.: The complexity of counting stable marriages. *SIAM Journal on Computing* **15**, 655-667 (1986).

[13] Kelso A.S., Crawford V.: Job matching, coalition formation and gross substitutes, *Econometrica* **50** (6), 1483–1504 (1982).

[14] Knuth D.E. Marriage Stables. *Montreal University Press,* Montreal (1976).

[15] Manlove D.F., O'Malley G., Prosser P., Unsworth C.: A Constraint Programming approach to the Hospitals / Residents problem. In *Proceedings of the CPAIOR'07, Lecture Notes in Computer Science* **4510**, 28-43 (2007).

[16] Martínez R., Massó J., Neme A., Oviedo J.: An algorithm to compute the full set of many-to-many stable matchings. *Mathematical Social Sciences* **47**, 187-210 (2004)

[17] McVitie D.G., Wilson L.B.: The Stable Marriage Problem. *Communications of the ACM* **14**, 486-490 (1971).

[18] Roth A.E., Sotomayor M.A.O.: Two-sided Matching: a study in game theoretic modelling and analysis, *Economic Society Monographs* **18**, Cambridge University Press (1990).

[19] Sotomayor M.A.O.: Three remarks on the many-to-many stable matching problem, *Mathematical Social Sciences* **38**, 55-70 (1999).