Egerváry Research Group
on Combinatorial Optimization

Technical reportS

TR-2008-06. Published by the Egerváry Research Group, Pázmány P. sétány 1/C, H–1117, Budapest, Hungary. Web site: www.cs.elte.hu/egres . ISSN 1587–4451.

An algorithm for a super-stable roommates problem

Tamás Fleiner, Robert W. Irving, and
David F. Manlove

May 10, 2008

An algorithm for a super-stable roommates problem

Tamás Fleiner*, Robert W. Irving**, and David F. Manlove***

Abstract

In this paper we describe an efficient algorithm that decides if a stable matching exists for a generalized stable roommates problem, where, instead of linear preferences, agents have partial preference orders on potential partners. Furthermore, we may forbid certain partnerships, that is, we are looking for a matching such that none of the matched pairs is forbidden, and yet, no blocking pair (forbidden or not) exists.

To solve the above problem, we generalize the first algorithm for the ordinary stable roommates problem.

Keywords: stable marriages, stable roommates problem, polynomial time algorithm

1 Introduction

The study of stable matching problems were initiated by Gale and Shapley [3] who introduced the *stable marriage problem*. In this problem each of n men and n women have a linear preference order on the members of the opposite gender. We ask if there exists a marriage scheme in which no man and woman mutually

*Department of Computer Science and Information Theory, Budapest University of Technology and Economics, Magyar Tudósok körútja 2, Budapest, H-1117. `fleiner@cs.bme.hu`
Research was supported by the János Bolyai Research fellowship of the Hungarian Academy of Sciences, by the Royal Society of Edinburgh International Exchange Programme, and by the K69027 and K60802 OTKA grants.

**Department of Computing Science University of Glasgow G12 8QQ, UK. `rwi@dcs.gla.ac.uk` Supported by the Engineering and Physical Sciences Research Council grant EP/E011993/1.

***Department of Computing Science University of Glasgow G12 8QQ, UK. `davidm@dcs.gla.ac.uk` Supported by the Royal Society of Edinburgh / Scottish Executive Personal Research Fellowship, and Engineering and Physical Sciences Research Council grant GR/R84597/01 and EP/E011993/1.

prefer one another to their eventual partners. The authors prove that the so called deferred acceptance algorithm always finds a stable marriage scheme.

It is natural to ask the same question for a more general, nonbipartite (sometimes called: one sided) model, in which we have n agents with preference orders on all other agents. This is the so called *stable roommates problem*, and we are looking for a matching (i.e. a pairing of the agents) such that no two agents prefer one another to their eventual partners. Such a matching is called a *stable matching*. A significant difference between the stable marriage and the stable roommates problems is that for the latter, it might happen that no stable matching exists. The stable roommates problem was solved by Irving [4], with an efficient algorithm that either finds a stable matching or concludes that no stable matching exists for the particular problem. Later, Tan [8] used this algorithm to give a good characterization, that is, he proved that for any stable roommates problem, there always exists a so called *stable partition* (that can be regarded as a half integral, fractional stable matching) with the property that either it is a stable matching, or it is a compact proof for the nonexistence of a stable matching.

In both the stable marriage and the stable roommates problems strict preferences of the participating agents play a crucial role. However, in many practical situations, we have to deal with indifferences in the preference orders. Our model for this is that preference orders are partial (rather than linear) orders. We can extend the notion of a stable matching to this model in at least three different ways. One possibility is that a matching is *weakly stable* if no pair of agents a, b exists such that they mutually strictly prefer one another to their eventual partner. Ronn proved that deciding the existence of a weakly stable matching is NP-complete [6]. A more restrictive notion is that a matching is *strongly stable* if there are no agents a and b such that a strictly prefers b to his eventual partner and b does not prefer his eventual partner to a. Scott gave an algorithm that finds a strongly stable matching or reports if none exists in $O(m^2)$ time [7]. The most restrictive notion is that of super-stability. A matching is *super-stable* if there exist no two agents a and b such that neither of them prefers his eventual situation to being a partner of the other. In other words, a matching is super-stable, if it is stable for any linear extensions of the preference orders of the agents. For the case where indifference is transitive, Irving and Manlove gave an $O(m)$ algorithm to find a super-stable matching, if exists [5]. Interestingly, the algorithm has in two phases, just like Irving's [4], but its second phase is completely different. It is also noted there that the algorithm works without modification for the more general poset case.

The motivation of our present work is to give a direct algorithm to this kind of stable matching problem by generalizing Irving's original algorithm. This latter algorithm works in such a way that it keeps on deleting edges of the underlying graph until a (stable) matching is left. It turnes out that deleting an edge is too harsh a transformation, we need a finer one as well. For this reason, we extend our model and we also allow forbidden edges. And, instead of deleting, we will also forbid certain edges during the algorithm. Although a stable matching problem with forbidden edges is a special case of the poset problem (for each forbidden edge add a parallel copy and declare them equal in the preference orders), it is an interesting problem in itself. Dias *et al.* gave an $O(m)$ algorithm to the stable marriage problem with forbidden pairs [1].

Our present problem, the super-stable matching problem with forbidden edges is known to be polynomial-time solvable. Fleiner *et al.* exhibited a reduction of this problem to 2-SAT [2]. However, this reduction does not give much information about the structure of super-stable matchings. In particular, it is not obvious if there exists a "short proof" for the nonexistence of a super-stable matching, just like Tan's stable partition [8] works for the ordinary stable roommates problem. Our direct approach may be useful to find such a certificate.

To formalize our problem, we define a *preference model* as a triple $(G, F, \mathcal{O})$, where $G = (V, E)$ is a graph, the set F of *forbidden edges* is a subset of the edge set E of G, and $\mathcal{O} = \{<_v : v \in V\}$, where $<_v$ is a partial order on the star $E(v)$ of v (that is, the set of those edges of G that are incident with vertex v). It is convenient to think that we deal with a market situation: vertices of G are the acting agents and edges of G represent possible partnerships between them. Partial order $<_v$ is the preference order of agent v on his possible partnerships. Parallel edges are allowed in G: the same two agents may form different partnerships, that may yield different profits for them. A subset M of E is a *matching* if edges of M do not share a vertex, that is, each agent participates in at most one partnership. Matching M is a *stable* (we omit the super prefix for convenience), if $M \subseteq E \setminus F$ (in other words, no edge of M is forbidden, that is, all edges of M are *free*), and each edge e of E is *dominated by* M, that is, there is an edge $m \in M$ and a vertex $v \in V$ such that $m \leq_v e$. If M is a matching and e is not dominated by M then e is a *blocking edge of* M. The *stable roommates problem with partial orders and forbidden pairs* is the decision problem on an input preference model whether it has a stable matching or not.

Note, that in the standard terminology, agents have preferences on possible partners, rather than on partnerships. It is easy to see that in our approach, this corresponds to the case where graph G in the preference model is simple. We also have a slightly different way of defining stability via dominance. Traditionally, we first define the notion of blocking and then we say that a stable matching is a matching that has no blocking edge. Also note that the stable roommates problem is the special case where G is simple, $F = \emptyset$, and each order $<_v$ is linear.

2 The generalized algorithm

Let us fix a preference model $(G_0, F_0, \mathcal{O}_0)$, as the input of our algorithm. We should find a stable matching, if it exists. The algorithm works step by step.

In each step, it transforms the actual model $(G_i, F_i, \mathcal{O}_i)$ to a simpler model $(G_{i+1}, F_{i+1}, \mathcal{O}_{i+1})$ in such a way that the answer to the latter problem is a valid answer to the former one, as well. That is, after the transformation no new stable matching can emerge and if there was a stable matching in the former model, then there should also be one in the new model. We use three kind of transformations: we forbid edges, we delete forbidden edges and we restrict the model.

If e is a free edge of G_i, then *forbidding* e means that $G_{i+1} := G_i$, $F_{i+1} := F_i \cup \{e\}$ and $\mathcal{O}_{i+1} := \mathcal{O}_i$. The algorithm may forbid e if either no stable matching contains e or if e is not contained in all stable matchings. After such a forbidding, there is a stable matching in $(G_i, F_i, \mathcal{O}_i)$ if and only if there is one in $(G_{i+1}, F_{i+1}, \mathcal{O}_{i+1})$, and any stable matching of $(G_{i+1}, F_{i+1}, \mathcal{O}_{i+1})$ is a stable matching of $(G_i, F_i, \mathcal{O}_i)$. Forbidding a subset E' of E means that we simultaneously forbid all edges of E'.

If e is a forbidden edge of G_i then *deleting* e means that we delete e from G_i to get G_{i+1}, $F_{i+1} := F_i \setminus \{e\}$, and the partial orders in $\mathcal{O}_{i+1}$ are the restrictions of the corresponding partial orders of $\mathcal{O}_i$, to the corresponding stars of G_{i+1}. The algorithm may delete e if there exists no matching in $(G_i, F_i, \mathcal{O}_i)$ that is blocked only by e. This implies that the set of stable matchings in $(G_i, F_i, \mathcal{O}_i)$ and in $(G_{i+1}, F_{i+1}, \mathcal{O}_{i+1})$ is the same.

If U is a proper subset of the vertex set of G_i then *restriction to* U means that G_{i+1} is the graph we get from G_i after deleting all vertices outside U, F_{i+1} is the subset of F_i that is spanned by G_{i+1}, and the partial orders of $\mathcal{O}_{i+1}$ are the restricted partial orders of $\mathcal{O}_i$ to the corresponding stars of G_{i+1}.

We shall use different kinds of steps throughout the algorithm. There is a certain hierarchy of them: the next step of the algorithm always has the highest priority among those steps that can be executed. To describe these step types, we say that edge $e = E_i(v)$ of G_i (forbidden or not) is a *first choice edge of* v, if there is no edge $f \in E_i(v) \setminus F_i$ with $f <_v e$ (i.e., if no free edge can dominate e at vertex v). Note that there can be more than one 1st choices of v present.

0th priority (proposal) step If $e = vw$ is a 1st choice of v then orient e from v to w, and $(G_i, F_i, \mathcal{O}_i) = (G_{i+1}, F_{i+1}, \mathcal{O}_{i+1})$.

Clearly, the set of stable matchings does not change by a proposal step. We shall call the 1st choice arcs we create by the proposal steps *1-arcs*. Note that it is possible that a 1-arc is bioriented.

After the algorithm have found all 1-arcs, it looks for a

1st priority (mild rejection) step If 1-arc e of G_i points to v and $E_i(v) \ni f \not<_v e$ (that is, f is not better than e according to v in G_i) then forbid f.

Obviously, if f is in some matching M then $e \notin M$, and hence e (being a first choice at its other end) blocks M. So f cannot be in a stable matching, we can forbid it. Eventually, we have to delete edges and the algorithm does this only the following way.

2nd priority (firm rejection) step If some free 1-arc e of G_i points to v and $e <_v f \in E_i(v)$ (e is better than f according to v in G_i) then we delete f.

Note that the above f is already forbidden by a 1st priority step. Assume that f blocks matching M, hence, in particular, $e \notin M$. But e is a first choice of its other endvertex, thus e is also blocking M. So deleting f does not change the set of stable matchings of the preference model.

Note that the so called 1st phase steps in Irving's algorithm [4] for the stable roommates problem are the special cases of our proposal and firm rejection steps. It is true for the stable roommates problem that as soon as no more 1st phase steps can be executed, the preference model has the so called first-last property: if some edge $e = uv$ is a first choice of u, then e is the last choice of v. A generalization of this property holds in our setting. Assume that the algorithm cannot execute a 0th, 1st or 2nd priority step for $(G_i, F_i, \mathcal{O}_i)$. Let V_0 denote the set of those vertices of G_i that are not incident with any free edges, V_1 stand for the set of those vertices of G_i that are incicent with a bioriented free 1-arc and V_2 refer to the set of the remaining vertices of G_i. The following properties are true.

Theorem 2.1. *Assume that no proposal or rejection step can be made in G_i, and let V_0, V_1 and V_2 be defined as above.*

If $v \in V_1 \cup V_2$ then there is a unique 1*-arc entering v and there is a unique* 1*-arc leaving v, and all these* 1*-arcs are free. There is no edge of G_i leaving V_0. Bioriented free* 1*-arcs form a matching M_1 that covers V_1, and no more edges are incident with V_1 in G_i.*

M is a stable matching of $(G_i, F_i, \mathcal{O}_i)$ if and only if the following properties hold:
(1) each vertex of V_0 is isolated and *(2) $M_1 \subseteq M$ and*
(3) $M \setminus M_1$ is a stable matching of the model restricted to V_2.

Proof. Let $v \in V_1 \cup V_2$. By definition, there is at least one free edge incident with v, hence there is at least one free 1-arc leaving v. On the other hand, no proposal or rejection step (mild or firm) can be made in G_i, hence at most one free 1-arc enters v. By definition, no free 1-arc enter vertices of V_0, and this means that 1-arcs leaving vertices of $V_1 \cup V_2$ enter this very same vertex set. Consequently, there is a unique free 1-arc leaving and entering each vertex of $V_1 \cup V_2$. Can there be a forbidden 1-arc e incident with a vertex v of $V_1 \cup V_2$? The answer is no: such an arc cannot enter v, as otherwise v would be able to reject. So $e = uv$ is a 1-arc from $V_1 \cup V_2$ to V_0. But v is not incident with any free arcs by definition, thus vu is a 1-arc that enters vertex u of $V_1 \cup V_2$, contradiction. Hence all 1-arcs that are incident with $V_1 \cup V_2$ are free.

Let $u \in V_0$ and $e = uv$ be an edge of G_i. Clearly e is a 1-arc and e is forbidden by the definition of V_0, so $v \in V_0$ holds. This means that all edges incident with a vertex of V_0 are completely inside V_0.

If v is in V_1 then there is a unique 1-arc a that leaves v, so a must bioriented by the definition of V_1. If $e = uv$ is an edge of G_i then either $e = a$ or e is not a first choice of v, hence $a \prec_v e$ holds. But in this case v should delete e in a firm rejection step as a is a 1-arc entering v. This argument shows that edges of G_i that are incident with V_1 are all bioriented and form a matching M_1 covering V_1.

Assume now that M is a stable matching of G_i. No edge of G_i incident with a vertex of V_0 can block M, hence V_0 consists of isolated vertices. As M is not blocked by an edge of M_1, edges of M_1 all belong to M. As there is no edge of G_i that leaves V_2, edges of M in V_2 form a stable matching of the restricted model to V_2.

Let now M_2 be a stable matching of the model restricted to V_2 and assume that V_0 consists of isolated vertices. Let $M := M_2 \cup M_1$. Clearly M is a matching. If some edge e blocks M then e cannot be incident with V_0, as these vertices are isolated, and e cannot have a vertex in V_1 either, as vertices of V_1 are only incident with edges of M_1. Hence e is an edge within V_2, contradicting to the fact that M_2 is a matching. □

If some vertex of V_0 is not isolated then the algorithm stops and concludes that no stable matching exists. If this is not the case, then another possibility is that $V_2 = \emptyset$. This case the algorithm stops, and reports that there is a stable matching. To construct one, the algotithm takes M_1 and completes it to a stable matching of the original preference model with the previously listed other matchings of type M_1. Theorem 2.1 justifies both these terminations. If none of the above cases hold then $V_2 \neq \emptyset$ and we make a

3rd priority (restriction) step: if $V_0 \cup V_1 \neq \emptyset$ then we restrict the model to V_2. By Theorem 2.1, it is enough to find a stable matching for the restricted G_{i+1}: if there is such a matching M', then $M' \cup M_1$ is a stable matching of G_i. If no stable matching exists after the restriction, then there was no stable matching even before it.

Assume that in $(G_i, F_i, \mathcal{O}_i)$, the algorithm can execute no 0th, 1st or 2nd or 3rd priority step. An edge $e \in E_i(v)$ is a *second choice of* v if $e >_v f \notin F$ implies that f is *the* 1st choice of v. In other words, e is a second choice, if the only free edge that dominates e at v is the unique 1-arc leaving v. Note that every vertex v of G_i is incident with at least one free second choice edge: in the "worst case" it is the unique 1-arc pointing to v.

4th priority step If $e = vw$ is a second choice of v then (counterintuitively) orient e from w to v. Arcs created at this step are called *2-arcs*. As we do not modify the preference model ($G_{i+1} = G_i, F_{i+1} = F_i$ and $\mathcal{O}_{i+1} = \mathcal{O}_i$), the set of stable matchings does not change by a 4th priority step.

What is the meaning of a 2-arc? Let, vv' and uu' be 1-arcs and $u'v$ be a 2-arc. As vv' is the only free edge dominating $u'v$ at v, we get that if uu' is present in a stable matching M then uu' does not dominate uv', hence $vv' \in M$ follows. In other words, 2-arcs represent implications on 1-arcs. This allows us to build an implication structure on the set of 1-arcs.

In this structure, two 1-arcs e and f are called *sm-equivalent*, if there is a directed cycle D formed by 1-arcs and 2-arcs in an alternating manner such that

D contains both e and f. (Note that D may use the same vertex more than once.) Sm-equivalence is clearly an equivalence relation and if C is an sm-class and M is a stable matching then either C is disjoint from M or C is contained in M.

Beyond determining sm-equivalence classes, 2-arcs yield further implications between sm-classes: if uu' is a 1-arc of sm-class C and vv' is a 1-arc of sm-class C' and $u'v$ is a 2-arc, then sm-class C "implies" sm-class C' in such a way that if C is not disjoint from stable matching M then M contains both classes C and C'. Assume that sm-class C is on the top of this implication structure, i.e. C is not implied by any other sm-class (but C may imply certain other classes). Formally, we have that

$$\begin{gathered}\text{if } vv' \text{ is a 1-arc of } C \text{ and } w'v \text{ is a 2-arc} \\ \text{then (the unique) 1-arc } ww' \text{ is sm-equivalent to } vv'.\end{gathered} \tag{1}$$

To find a top sm-class C, introduce an auxiliary digraph on the vertices of G_i, such that if uu' is a 1-arc and $u'v$ is a 2-arc, then we introduce an arc uv of the auxiliary graph. It is well known that by depth first search, we can find a source strong component of the auxiliary graph in linear time. If it contains vertices $u_1, u_2, \ldots, u_k$ then it determines a top sm-class $C = \{u_1u_1', u_2u_2', \ldots, u_ku_k'\}$ formed by 1-arcs. Note that it is possible here that $u_i = u_j'$ for different i and j.

5th priority step If for 1-arcs $u_iu_i', u_ju_j' \in C$ there are 2-arcs vu_i and vu_j with $vu_i \not<_v vu_j$ then forbid vu_i.

To justify this step, assume that $vu_i \in M$ for some stable matching M of G_i. As vu_i does not dominate vu_j, vu_j has to be dominated at u_j by $u_ju_j' \in M$. As u_iu_i' and u_ju_j' are sm-equivalent, this means that u_ju_j' also belongs to M, a contradiction. So vu_i does not belong to any stable matching and after forbidding it, the set of stable matchings does not change. Note that after we take a 5th priority step, new 2-arcs may be created so we might continue with a 4th priority step.

6th priority step Forbid all edges of C in $(G_i, F_i, \mathcal{O}_i)$.

To justify this kind of step, we check two cases. Case 1 is that C is not a matching, that is, $u_i = u_j'$ for some $i \neq j$. As a subset of a matching is a matching, no matching (hence no stable matching) can contain C. So by sm-equivalence, C is disjoint from any stable matching of G_i, and forbidding C is not changing the set of stable matchings.

Case 2 is that C is a matching. Each u_i is adjacent to at least two free edges: the incoming and the outgoing 1-arcs. So each u_i receives at least one free 2-arc. This free 2-arc must come from some u_j' by property (1). Let C' denote the set of free 2-arcs of the form $u_j'u_i$. As we have seen, each u_i receives at least one arc of C', hence $|C'| \geq k$. As we cannot execute any more 5th priority steps in $(G_i, F_i, \mathcal{O}_i)$, from each u_j' there is at most one arc of C' leaving, implying $|C'| \leq k$. This means that $|C'| = k$ and each u_i receives exactly one arc of C' and each u_i' sends exactly one arc of C'. As sets $\{u_1, u_2, \ldots, u_k\}$ and $\{u'_1, u'_2, \ldots, u'_k\}$ are disjoint, this means that set C' forms a perfect matching on vertices $u_1, u'_1, u_2, u'_2, \ldots, u_k, u'_k$.

Let M be a stable matcing of $(G_i, F_i, \mathcal{O}_i)$. If M is disjoint from C then M is stable in $(G_{i+1}, F_{i+1}, \mathcal{O}_{i+1})$ as well. Otherwise, by sm-equivalence, M contains all edges of C and disjoint from C'. We claim that $M' := M \setminus C \cup C'$ is another stable matching of $(G_i, F_i, \mathcal{O}_i)$ and hence it is a stable matching of $(G_{i+1}, F_{i+1}, \mathcal{O}_{i+1})$, as well.

Indeed: M' is a matching, as C and C' cover the same set of vertices. Each edge $u_i u'_i$ is dominated at u'_i by M' by Theorem 2.1. Each forbidden 2-arc of type $u'_j u_i$ is dominated at u'_j by the 5th priority step. For the remaining edges, if some edge e does not have a vertex u_i then e is dominated the same way in M' as in M. Otherwise, if u_i is a vertex of e then e is neither a first nor a second choice of u_i as we have already checked these edges. This means that the free 2-arc pointing to u_i is dominating e, so C' and thus M' also dominates e at u_i.

Clearly, this 6th priority step corresponds to the so called rotation elimination of Irving's algorithm [4], where $C \cup C'$ is the generalization of a rotation.

If the algorithm does not stop after some 2nd priority step with the conclusion that no stable matching exists then it keeps on forbidding and deleting edges. Sooner or later it cannot do this any more, so no further step can be made. Pick a vertex v of the actual G_i. As no 3rd priority step is possible, there is a free edge adjacent to v. So v sends a free 1-arc, and it also receives a free 1-arc. Again by the 3rd priority step, these arcs are different, hence there is a 2-arc pointing to v. This implies that a 5th or a 6th priority step can be executed, a contradiction. So the algorithm always terminates before a 3rd priority step either by concluding that no stable matching exists or by constructing a stable matching.

To convince ourselves about the polynomial time complexity of the algorithm let us calculate the cost of deleting or forbidding an edge. Clearly, the most time consuming is the 6th priority deletion step. For this we check every edge for the 1st and 2nd priority steps in $O(m)$ time (where m is the number of edges of G_0), and we check all vertices in $O(n)$ time for the 3rd priority step. (n is the number of vertices of G_0.) To check the possible 4th priority steps takes $O(m)$ time, and finding top sm-class C is a depth first search, that can be done in $O(n+m)$ time. Checking the 5th priority steps takes $O(m)$ time, and after this we can forbid C. So forbidding or deleting an edge takes altogether $O(n+m)$ time. We can delete or forbid at most $2m$ times altogether, so the total complexity of our algorithm is $O(m(n+m))$. (Note that this is a pretty rough estimate. Probably, by streamlining the algorithm, one can get a much better estimate.)

References

[1] Vânia M. F. Dias, Guilherme D. da Fonseca, Celina M. H. de Figueiredo, and Jayme L. Szwarcfiter, The stable marriage prob-

lem with restricted pairs, *Theoret. Comput. Sci.* (2003) **306**(1-3) 391–405

[2] Tamás Fleiner, Robert W. Irving, and David F. Manlove, Efficient algorithms for generalised stable marriage and roommates problems, *Theoret. Comput. Sci.* (2007) **381**(1-3) 162–176 and DCS Tech Report, TR-2005-207, `http://www.dcs.gla.ac.uk/publications/` (2005)

[3] D. Gale and L.S. Shapley, College admissions and stability of marriage, *Amer. Math. Monthly* (1962) **69**(1) 9–15

[4] Robert W. Irving, An efficient algorithm for the "stable roommates" problem, *J. Algorithms* (1985) **6**(4) 577-595

[5] Robert W. Irving and David F. Manlove, The stable roommates problem with ties, *J. Algorithms* (2002) **43**(1) 85–105

[6] Eytan Ronn, NP-complete stable matching problems, *J. Algorithms* (1990) **11**(2) 285–304

[7] Sandy Scott, The study of stable marriage problems with ties, 2005, PhD dissertation, University of Glasgow, Department of Computing Science.

[8] Jimmy J. M. Tan, A necessary and sufficient condition for the existence of a complete stable matching, *J. Algorithms* (1991) **12**(1) 154–178