



**UNIVERSITY**  
*of*  
**GLASGOW**

Irving, R.W and Manlove, D. F and Scott, S. (2003) Strong stability in the Hospitals / Residents problem. In, Alt, H. and Habib, M., Eds. *Proceedings of STACS 2003: the 20th International Symposium on Theoretical Aspects of Computer Science, 27 February - 1 March, 2003* Lecture Notes in Computer Science Vol 2607, pages 439-450, Berlin, Germany.

<http://eprints.gla.ac.uk/archive/00001064/>

# Strong Stability in the Hospitals/Residents Problem

Robert W. Irving, David F. Manlove\* and Sandy Scott

Department of Computing Science, University of Glasgow, Glasgow G12 8QQ, UK.  
*Email:* {rwi,davidm,sas}@dcs.gla.ac.uk. *Fax:* +44 141 330 4913.

**Abstract.** We study a version of the well-known Hospitals/Residents problem in which participants' preferences may involve ties or other forms of indifference. In this context, we investigate the concept of *strong stability*, arguing that this may be the most appropriate and desirable form of stability in many practical situations. When the indifference is in the form of ties, we describe an  $O(a^2)$  algorithm to find a strongly stable matching, if one exists, where  $a$  is the number of mutually acceptable resident-hospital pairs. We also show a lower bound in this case in terms of the complexity of determining whether a bipartite graph contains a perfect matching. By way of contrast, we prove that it becomes NP-complete to determine whether a strongly stable matching exists if the preferences are allowed to be arbitrary partial orders.

**Keywords:** stable matching problem; strong stability; hospitals/residents problem; polynomial-time algorithm; lower bound; NP-completeness.

## 1 Introduction

The Hospitals/Residents problem [3] is a many-to-one extension of the classical Stable Marriage problem (SM), so-called because of its widespread application to matching schemes that match graduating medical students (residents) to hospital posts. In particular the National Resident Matching Program (NRMP) in the USA [14], the Canadian Resident Matching Service [1], and the Scottish PRHO Allocations (SPA) matching scheme [6] all make use of algorithms that solve variants of this problem.

An instance of the classical Hospitals/Residents problem (HR) involves two sets, a set  $R$  of *residents* and a set  $H$  of *hospitals*. Each resident in  $R$  seeks to be *assigned* to exactly one hospital, and each hospital  $h \in H$  has a specified number  $p_h$  of posts, referred to as its *quota*. Each resident ranks a subset of  $H$  in strict order of preference, and each hospital ranks, again in strict order, those residents who have ranked it. These are the *preference lists* for the instance. Note that preference lists are *consistent* in the sense that a resident  $r$  appears on a hospital  $h$ 's list if and only if  $h$  appears on  $r$ 's list. Consistency of preference lists will be assumed throughout. A resident-hospital pair  $(r, h)$  are *mutually acceptable*

---

\* Supported by award NUF-NAL-02 from the Nuffield Foundation and grant GR/R84597/01 from the Engineering and Physical Sciences Research Council.

if they are each on the other's preference list, and we denote by  $A$  (a subset of  $R \times H$ ) the set of mutually acceptable pairs, with  $|A| = a$ .

A *matching*  $M$  is a subset of  $A$  such that  $|\{h : (r, h) \in M\}| \leq 1$  for all  $r$  and  $|\{r : (r, h) \in M\}| \leq p_h$  for all  $h$ . For a matching  $M$ , we denote by  $M(r)$  the hospital to which  $r$  is assigned in  $M$  (this is null if  $r$  is unassigned), and by  $M(h)$  the set of residents assigned to  $h$  in  $M$ . For a pair  $(r, h) \in A$ , we define  $h \prec_r M(r)$  to mean that either  $r$  is unassigned in  $M$ , or  $r$  prefers  $h$  to  $M(r)$ . Likewise, we define  $r \prec_h M(h)$  to mean that either  $|M(h)| < p_h$ , or  $h$  prefers  $r$  to at least one of the members of  $M(h)$ .

A matching  $M$  in an instance of HR is *stable* if there is no pair  $(r, h) \in A \setminus M$ , such that  $h \prec_r M(r)$  and  $r \prec_h M(h)$ . If such a pair  $(r, h)$  exists it is said to be a *blocking pair* for the matching  $M$ , or to *block*  $M$ . The existence of a blocking pair potentially undermines the matching, since both members of the pair could improve their situation by becoming matched to each other.

An instance of HR can be solved by an extension of the Gale/Shapley (GS) algorithm for SM [2, 3]. SM can be viewed as a restriction of HR in which each hospital has quota 1 (and the residents and hospitals are re-named men and women). As with SM, a stable matching exists for every instance of HR, and all stable matchings for a given instance have the same size [3]. The extension of the GS algorithm finds one such matching in  $O(a)$  time [3].

Recent pressure from student bodies associated with the NRMP has ensured that the extended version of the GS algorithm that is employed by the scheme is now *resident-oriented*, meaning that it produces the *resident-optimal* stable matching for a given instance of HR [13]. This is the unique stable matching  $M_0$  in which every resident assigned in  $M_0$  is assigned the best hospital that he/she could obtain in any stable matching, and any resident unassigned in  $M_0$  is unassigned in any stable matching.

In this paper we consider generalisations of HR in which preferences involve some form of indifference. This is highly relevant for practical matching schemes — for example, a popular hospital may be unable or unwilling to produce a strict ranking over all of its many applicants.

The most natural form of indifference involves ties. A set  $R'$  of  $k$  residents forms a *tie* of length  $k$  in the preference list of hospital  $h$  if  $h$  does not prefer  $r_i$  to  $r_j$  for any  $r_i, r_j \in R'$  (i.e.  $h$  is *indifferent* between  $r_i$  and  $r_j$ ), while for any other resident  $r$  who is acceptable to  $h$ , either  $h$  prefers  $r$  to all of the residents in  $R'$ , or  $h$  prefers all of the residents in  $R'$  to  $r$ . A tie on a resident's list is defined similarly. For convenience in what follows, we consider an untied entry in a preference list as a tie of length 1.

We denote by HRT the version of the problem in which preference lists can include arbitrary ties, and by HRP the version in which each preference 'list' can be an arbitrary partial order. This latter version allows for more complex forms of indifference, generalising the case of lists with ties. For a given practical application, a variety of external factors could contribute to a given preference structure, yielding a more complex form of indifference represented by an arbitrary partial order. Given an instance  $I$  of HRT or HRP, a *derived* instance of

HR is any instance of HR obtained from  $I$  by resolving the indifference (breaking all of the ties or extending each partial order to a total order).

These extensions of the original problem force a re-evaluation of the concept of a blocking pair. We could view a pair  $(r, h)$  to be a blocking pair if, by coming together (a) both parties would be better off, or (b) neither party would be worse off, or (c) one party would be better off and the other no worse off. These three possibilities give rise to the notions of weak stability, super-stability, and strong stability, respectively, first considered by Irving [5] in the context of the Stable Marriage problem. We now formally define these three forms of stability.

A matching  $M$  in an instance of HRT or HRP is *weakly stable* if there is no pair  $(r, h) \in A \setminus M$ , such that  $h \prec_r M(r)$  and  $r \prec_h M(h)$ .

A weakly stable matching exists for every instance of HRP, and can be found by forming a derived instance of HR, and applying the extended GS algorithm. It turns out that, in contrast to HR, weakly stable matchings for an instance of HRT may have different sizes, and it is notable that the problem of finding the largest weakly stable matching, and various other problems involving weak stability, are NP-hard [9, 12].

To define super-stability and strong stability we need to extend our notation. For a given matching  $M$  and pair  $(r, h) \in A$ , we define  $h \triangleleft_r M(r)$  to mean that  $r$  is unassigned in  $M$ , or that  $r$  prefers  $h$  to  $M(r)$ , or is indifferent between them. Likewise,  $r \triangleleft_h M(h)$  means that  $|M(h)| < p_h$ , or  $h$  prefers  $r$  to at least one member of  $M(h)$ , or is indifferent between  $r$  and at least one member of  $M(h)$ .

A matching  $M$  is *super-stable* if there is no pair  $(r, h) \in A \setminus M$ , such that  $h \triangleleft_r M(r)$  and  $r \triangleleft_h M(h)$ .

By contrast with weak stability, it is trivial to show that there are instances of HRT and HRP for which no super-stable matching exists. However, there is an  $O(a)$  algorithm to determine whether an instance of HRT admits a super-stable matching, and to find one if it does [7]. With some straightforward modifications, this algorithm is also applicable in the more general context of HRP.

A matching  $M$  is *strongly stable* if there is no pair  $(r, h) \in A \setminus M$ , such that either (i)  $h \prec_r M(r)$  and  $r \triangleleft_h M(h)$ ; or (ii)  $h \triangleleft_r M(r)$  and  $r \prec_h M(h)$ . If  $(r, h) \in M$  for some strongly stable matching  $M$  we say that  $(r, h)$  is a *strongly stable pair*.

Again, it is easy to construct an instance of HRT that does not admit a strongly stable matching (see [5] for further details). Clearly, as is implied by the terminology, a super-stable matching is strongly stable, and a strongly stable matching is weakly stable.

There is a sense in which strong stability can be viewed as the most appropriate criterion for a practical matching scheme when there is indifference in the preference lists, and that in cases where a strongly stable matching exists, it should be chosen instead of a matching that is merely weakly stable. Consider a weakly stable matching  $M$  for an instance of HRT or HRP, and suppose that  $h \prec_r M(r)$  while  $h$  is indifferent between  $r$  and at least one member  $r'$  of  $M(h)$ . Such a pair  $(r, h)$  would not constitute a blocking pair for weak stability (if  $|M(h)| = p_h$ ). However,  $r$  might have such an overriding preference for  $h$  over

$M(r)$  that he is prepared to engage in persuasion, even bribery, in the hope that  $h$  will reject  $r'$  and accept  $r$  instead. Hospital  $h$ , being indifferent between  $r$  and  $r'$  may yield to such persuasion, and, of course, a similar situation could arise with the roles reversed. However, the matching cannot be potentially undermined in this way if it is strongly stable. On the other hand, insisting on super-stability seems unnecessarily restrictive (and is less likely to be attainable).

Hence, strong stability avoids the possibility of a matching being undermined by persuasion or bribery, and is therefore a desirable property in cases where it can be achieved.

In this paper we present an  $O(a^2)$  algorithm for finding a strongly stable matching, if one exists, given an instance of HRT, thus solving an open problem described in [7]. Our algorithm is resident-oriented in that it finds a strongly stable matching with similar optimality properties to those of the resident-optimal stable matching in HR, as mentioned above. This algorithm is a non-trivial extension of the strong stability algorithms for the stable marriage problem due to Manlove [10] and Irving [5]. We also prove that the complexity of any algorithm for HRT under strong stability has the same lower bound as applies to the problem of determining if a bipartite graph has a perfect matching. By contrast, we show that the problem of deciding whether a given instance of HRP admits a strongly stable matching is NP-complete.

The remainder of this paper is structured as follows. In Section 2 we present the polynomial-time algorithm that finds a strongly stable matching in an instance of HRT, when one exists. In Section 3 we establish the complexity of the algorithm to be  $O(a^2)$ . The lower bound for the problem of finding a strongly stable matching in an instance of HRT is given in Section 4, whilst Section 5 contains the NP-completeness result for HRP under strong stability. Finally, Section 6 presents our conclusions and some open problems.

## 2 An algorithm for strong stability in HRT

In this section we describe our algorithm for finding a strongly stable matching, if one exists, given an instance of HRT, and prove its correctness. Before doing so, we present some definitions relating to the algorithm.

A hospital  $h$  such that  $|\{r : (r, h) \in M\}| = p_h$  is said to be *full* in the matching  $M$ . During the execution of the algorithm residents become *provisionally assigned* to hospitals, and it is possible for a hospital to be provisionally assigned a number of residents that exceeds its quota. At any stage, a hospital is said to be *over-subscribed*, *under-subscribed* or *fully-subscribed* according as it is provisionally assigned a number of residents greater than, less than, or equal to, its quota. We describe a hospital as *replete* if at any time during the execution of the algorithm it has been over-subscribed or fully subscribed.

The algorithm proceeds by deleting from the preference lists pairs that cannot be strongly stable. By the *deletion* of a pair  $(r, h)$ , we mean the removal of  $r$  and  $h$  from each other's lists, and, if  $r$  is provisionally assigned to  $h$ , the breaking of this provisional assignment. By the *head* and *tail* of a preference list at a given

point we mean the first and last ties respectively on that list (recalling that a tie can be of length 1). We say that a resident  $r$  is *dominated* in a hospital  $h$ 's list if  $h$  prefers to  $r$  at least  $p_h$  residents who are provisionally assigned to it.

A resident  $r$  who is provisionally assigned to a hospital  $h$  is said to be *bound* to  $h$  if  $h$  is not over-subscribed or  $r$  is not in  $h$ 's tail (or both). The *provisional assignment graph*  $G$  has a vertex for each resident and each hospital, with  $(r, h)$  forming an edge if resident  $r$  is provisionally assigned to hospital  $h$ . A *feasible matching* in a provisional assignment graph is a matching  $M$  such that, if  $r$  is bound to one or more hospitals, then  $r$  is matched with one of these hospitals in  $M$ , and subject to this restriction,  $M$  has maximum possible cardinality.

A *reduced assignment graph*  $G_R$  is formed from a provisional assignment graph as follows. For each resident  $r$ , for any hospital  $h$  such that  $r$  is bound to  $h$ , we delete the edge  $(r, h)$  from the graph, and we reduce the quota of  $h$  by one; furthermore, we remove *all* other edges incident to  $r$ . Each isolated resident vertex is then removed from the graph. Finally, if the quota of any hospital  $h$  is reduced to 0, or  $h$  becomes an isolated vertex, then  $h$  is removed from the graph. For each surviving  $h$  we denote by  $p'_h$  the revised quota.

Given a set  $Z$  of residents in  $G_R$ , define  $\mathcal{N}(Z)$ , the *neighbourhood* of  $Z$ , to be the set of hospital vertices adjacent in  $G_R$  to a resident vertex in  $Z$ . The *deficiency* of  $Z$  is defined by  $\delta(Z) = |Z| - \sum_{h \in \mathcal{N}(Z)} p'_h$ . It is not hard to show that, if  $Z_1$  and  $Z_2$  are maximally deficient, then so also is  $Z_1 \cap Z_2$ , so there is a unique minimal set with maximum deficiency. This is the *critical set*.

The algorithm, displayed in Figure 1, begins by assigning each resident to be free (i.e. not assigned to any hospital), and each hospital  $h$  to be non-replete. The iterative stage of the algorithm involves each free resident in turn being provisionally assigned to the hospital(s) at the head of his list. If by gaining a new provisional assignee a hospital  $h$  becomes fully or over-subscribed then it is set to be replete, and each pair  $(r, h)$ , such that  $r$  is dominated in  $h$ 's list, is deleted. This continues until every resident is provisionally assigned to one or more hospitals or has an empty list. We then find the reduced assignment graph  $G_R$  and the critical set  $Z$  of residents. As we will see later, no hospital in  $\mathcal{N}(Z)$  can be assigned a resident from those in its tail in any strongly stable matching, so all such pairs are deleted. The iterative step is then reactivated, and this entire process continues until  $Z$  is empty, which must happen eventually, since if  $Z$  is found to be non-empty, then at least one pair is subsequently deleted from the preference lists.

Let  $M$  be any feasible matching in the final provisional assignment graph  $G$ . Then  $M$  is a strongly stable matching unless either (a) some replete hospital  $h$  is not full in  $M$ , or (b) some non-replete hospital  $h$  has a number of assignees in  $M$  less than its degree in  $G$ ; in cases (a) and (b) no strongly stable matching exists.

The correctness of Algorithm HRT-strong, and an optimality property of any strongly stable matching that it finds, are established by means of three key lemmas. Here we give sketch proofs of the three lemmas. The full proofs, together with some auxiliary lemmas, appear in [8].

```

assign each resident to be free;
assign each hospital to be non-replete;
repeat {
    while some resident  $r$  is free and has a non-empty list
        for each hospital  $h$  at the head of  $r$ 's list {
            provisionally assign  $r$  to  $h$ ;
            if  $h$  is fully-subscribed or over-subscribed {
                set  $h$  to be replete;
                for each resident  $r'$  dominated on  $h$ 's list
                    delete the pair  $(r', h)$ ; } }
        form the reduced assignment graph;
        find the critical set  $Z$  of residents;
        for each hospital  $h \in \mathcal{N}(Z)$ 
            for each resident  $r$  in the tail of  $h$ 's list
                delete the pair  $(r, h)$ ;
    } until  $Z == \emptyset$ ;
let  $G$  be the final provisional assignment graph;
let  $M$  be a feasible matching in  $G$ ;
if (some replete hospital is not full in  $M$ ) or
(some non-replete hospital has fewer assignees in  $M$  than its degree in  $G$ )
    no strongly stable matching exists;
else
    output the strongly stable matching specified by  $M$ ;

```

**Fig. 1.** Algorithm HRT-strong

**Lemma 1.** *A matching output by Algorithm HRT-strong is strongly stable.*

*Proof (Sketch).* We suppose that a pair  $(r, h)$  blocks a matching  $M$  output by Algorithm HRT-strong, and show that  $(r, h)$  cannot have been deleted. Hence  $r$  must be bound to both  $h$  and  $M(r)$  in  $G$ , the final provisional assignment graph. But we also show that, if any resident is bound to two hospitals in  $G$  then the algorithm reports that no strongly stable matching exists, a contradiction.  $\square$

**Lemma 2.** *No strongly stable pair is ever deleted during an execution of Algorithm HRT-strong.*

*Proof (Sketch).* We let  $(r, h)$  be the first strongly stable pair deleted during some execution of the algorithm, and let  $M$  be a strongly stable matching containing  $(r, h)$ . We consider both points in the algorithm at which  $(r, h)$  could be deleted. If  $(r, h)$  is deleted because  $r$  becomes dominated in  $h$ 's list, we show that one of the residents provisionally assigned to  $h$  at this point must form a blocking pair for  $M$  with  $h$ . If, on the other hand,  $(r, h)$  is deleted because  $r$  is in  $h$ 's tail at a point when  $h$  is provisionally assigned a resident from the critical set  $Z$ , then we show that there is some resident  $r' \in Z$  and some hospital  $h' \in \mathcal{N}(Z)$ , with  $r'$  provisionally assigned to  $h'$  at this point, such that  $r'$  is not assigned in  $M$  to a hospital from the head of his current list, and  $h'$  is assigned in  $M$  at least one resident from the tail of its current list. It follows that  $(r', h')$  blocks  $M$ .  $\square$

**Lemma 3.** *Let  $M$  be a feasible matching in the final provisional assignment graph  $G$ . If (a) some non-replete hospital  $h$  has fewer assignees in  $M$  than provisional assignees in  $G$ , or (b) some replete hospital  $h$  is not full in  $M$ , then no strongly stable matching exists.*

*Proof (Sketch).* We suppose that condition (a) or (b) is satisfied, so that some hospital  $h$  must satisfy  $|M(h)| < \min(d_G(h), p_h)$ , where  $d_G(h)$  denotes the degree of vertex  $h$  in  $G$  (i.e. the number of residents provisionally assigned to  $h$ ). We also suppose that there is a strongly stable matching,  $M'$ , for the instance. We show that  $|M'| \leq |M|$ , so that some hospital  $h'$  must also satisfy  $|M(h')| < \min(d_G(h'), p_{h'})$ . Clearly  $h'$  then forms a blocking pair for  $M'$  with one of the residents provisionally assigned to it in  $G$ .  $\square$

Lemmas 1 and 3 prove the correctness of Algorithm HRT-strong. Further, Lemma 2 shows that there is an optimality property for each assigned resident in any strongly stable matching output by the algorithm. To be precise, we have proved:

**Theorem 1.** *For a given instance of HRT, Algorithm HRT-strong determines whether or not a strongly stable matching exists. If such a matching does exist, all possible executions of the algorithm find one in which every assigned resident is assigned as favourable a hospital as in any strongly stable matching, and any unassigned resident is unassigned in every strongly stable matching.*

### 3 Implementation and analysis of Algorithm HRT-strong

For the implementation and analysis of Algorithm HRT-strong, we require to describe the efficient construction of maximum cardinality matchings and critical sets in a context somewhat more general than that of simple bipartite graphs.

Consider a *capacitated* bipartite graph  $G = (V, E)$ , with bipartition  $V = R \cup H$ , in which each vertex  $h \in H$  has a positive integer *capacity*  $c_h$ . In this context, a *matching* is a subset  $M$  of  $E$  such that  $|\{h : \{r, h\} \in M\}| \leq 1$  for all  $r \in R$ , and  $|\{r : \{r, h\} \in M\}| \leq c_h$  for all  $h \in H$ .

For any vertex  $x$ , a vertex joined to  $x$  by an edge of  $M$  is called a *mate* of  $x$ . A vertex  $r \in R$  with no mate, or a vertex  $h \in H$  with fewer than  $c_h$  mates, is said to be *exposed*. An *alternating path* in  $G$  relative to  $M$  is any simple path in which edges are alternately in, and not in,  $M$ . An *augmenting path* is an alternating path both of whose end points are exposed. It is immediate that an augmenting path is of odd length, with one end point in  $R$  and the other in  $H$ .

The following lemmas may be established by straightforward extension of the corresponding results for one-to-one bipartite matching.

**Lemma 4.** *Let  $P$  be the set of edges on an augmenting path relative to a matching  $M$  in a capacitated bipartite graph  $G$ . Then  $M' = M \oplus P$  is a matching of cardinality  $|M| + 1$  in  $G$ .*

**Lemma 5.** *A matching  $M$  in a capacitated bipartite graph has maximum cardinality if and only if there is no augmenting path relative to  $M$  in  $G$ .*



The process of replacing  $M$  by  $M' = M \oplus P$  is called *augmenting*  $M$  along path  $P$ .

With these lemmas, we can extend to the context of capacitated bipartite graphs the classical augmenting path algorithm for a maximum cardinality matching. The algorithm starts with an arbitrary matching – say the empty matching – and repeatedly augments the matching until there is no augmenting path. The search for an augmenting path relative to  $M$  is organised as a restricted breadth-first search in which only edges of  $M$  are followed from vertices in  $H$  and only edges not in  $M$  are followed from vertices in  $R$ , to ensure alternation. The number of iterations is  $O(\min(|R|, \sum c_h))$ , and each search can be completed in  $O(|R| + |H| + |E|)$  time.

During the breadth-first search, we record the parent in the BFS spanning tree of each vertex. This enables us to accomplish the augmentation in time  $O(|R| + |H| + |E|)$ , observing that, for each vertex  $h \in H$ , the set of mates can be updated in constant time by representing the set as, say, a doubly linked list, and storing a pointer into this list from any child node in the BFS spanning tree.

Hence, overall, the augmenting path algorithm in a capacitated bipartite graph can be implemented to run in  $O((\min(|R|, \sum c_h))(|R| + |H| + |E|))$  time.

The following lemma (whose proof appears in [8]) points the way to finding the critical set.

**Lemma 6.** *Given a maximum cardinality matching  $M$  in the capacitated bipartite graph  $G_R$ , the critical set  $Z$  consists of the set  $U$  of unmatched residents together with the set  $U'$  of residents reachable from a vertex in  $U$  via an alternating path.*

During each iteration of the repeat-until loop of Algorithm HRT-strong we need to form the reduced assignment graph, which takes  $O(a)$  time, then search for a maximum cardinality matching in the bipartite graph  $G_R$ . This allows us to use Lemma 6 to find the critical set. The key to the analysis of Algorithm HRT-strong, as with Algorithm STRONG in [5], is bounding the total amount of work done in finding the maximum cardinality matchings.

It is clear that work done other than in finding the maximum cardinality matchings and critical sets is bounded by a constant times the number of deleted pairs, and so is  $O(a)$ .

Suppose that Algorithm HRT-strong finds a maximum cardinality matching  $M_i$  in the reduced assignment graph  $G_R$  at the  $i$ th iteration. Suppose also that, during the  $i$ th iteration,  $x_i$  pairs are deleted because they involve residents in the critical set  $Z$ , or residents tied with them in the list of a hospital in  $\mathcal{N}(Z)$ . Suppose further that in the  $(i + 1)$ th iteration,  $y_i$  pairs are deleted during the proposal sequence. Note that any edge in  $G_R$  at the  $i$ th iteration which is not one of these  $x_i + y_i$  deleted pairs must be in  $G_R$  at the  $(i + 1)$ th iteration, since a resident can only become bound to a hospital when he becomes provisionally assigned to it. In particular at least  $|M_i| - x_i - y_i$  pairs of  $M_i$  remain in  $G_R$  at the  $(i + 1)$ th iteration. Hence, in that iteration, we can start from these pairs and find a maximum cardinality matching in  $O(\min(na, (x_i + y_i + z_i)a))$  time,

where  $n$  is the number of residents and  $z_i$  is the number of edges in  $G_R$  at the  $(i + 1)$ th iteration which were not in  $G_R$  at the  $i$ th iteration.

Let  $s$  denote the number of iterations carried out, let  $S = \{1, 2, \dots, s\}$ , and let  $S' = S \setminus \{s\}$ . Let  $T \subseteq S'$  denote those indices  $i$  such that  $\min(na, (x_i + y_i + z_i)a) = na$ , and let  $t = |T|$ . Then the algorithm has time complexity  $O(\min(n, p)a + tna + a \sum_{i \in S' \setminus T} (x_i + y_i + z_i))$ , where  $p$  is the total number of posts, and the first term is for the first iteration. But  $\sum_{i \in S'} (x_i + y_i) \leq a$  and  $\sum_{i \in S'} z_i \leq a$  (since each of these summations is bounded by the total number of deletions and proposals, respectively), and since  $x_i + y_i + z_i \geq n$  for each  $i \in T$ , it follows that

$$tn + \sum_{i \in S' \setminus T} (x_i + y_i + z_i) \leq \sum_{i \in S'} (x_i + y_i + z_i) \leq 2a. \quad \text{Thus} \quad \sum_{i \in S' \setminus T} (x_i + y_i + z_i) \leq 2a - tn.$$

Hence the overall complexity of Algorithm HRT-strong is  $O(\min(n, p)a + tna + a(2a - tn)) = O(a^2)$ .

#### 4 A lower bound for finding a strongly stable matching

To establish the lower bound of this section, we let STRONGLY STABLE MATCHING IN HRT be the problem of deciding whether a given instance of HRT admits a strongly stable matching.

Let  $n$  denote the number of participants in an instance of HRT. We show that, for any function  $f$  on  $n$ , where  $f(n) = \Omega(n^2)$ , the existence of an  $O(f(n))$  algorithm for STRONGLY STABLE MATCHING IN HRT would imply the existence of an  $O(f(n))$  algorithm for PERFECT MATCHING IN BIPARTITE GRAPHS (the problem of deciding whether a given bipartite graph admits a perfect matching).

The result is established by the following simple reduction from PERFECT MATCHING IN BIPARTITE GRAPHS to STRONGLY STABLE MATCHING IN HRT.

Let  $G = (V, E)$  be a bipartite graph with bipartition  $V = R \cup H$ . Let  $R = \{r_1, \dots, r_n\}$  and  $H = \{h_1, \dots, h_n\}$ , and, without loss of generality, assume that  $G$  contains no isolated vertices. Also, for each  $i$  ( $1 \leq i \leq n$ ), let  $P_i$  denote the set of vertices in  $H$  adjacent to  $r_i$ .

We form an instance  $I$  of HRT as follows. Let  $p_i = 1$  for all  $i$ . Form a preference list for each participant in  $I$  as follows:

$$r_i : (P_i) (H \setminus P_i) \qquad h_i : (R) \qquad (1 \leq i \leq n)$$

In a given participant's preference list ( $S$ ) denotes all members of the set  $S$  listed as a tie in the position where the symbol occurs.

It is straightforward to verify that  $G$  admits a perfect matching if and only if  $I$  admits a strongly stable matching. Clearly the reduction may be carried out in  $O(n^2)$  time. Hence, for any function  $f$  on  $n$ , where  $f(n) = \Omega(n^2)$ , an  $O(f(n))$  algorithm for STRONGLY STABLE MATCHING IN HRT would solve PERFECT MATCHING IN BIPARTITE GRAPHS in  $O(f(n))$  time. The current best algorithm for PERFECT MATCHING IN BIPARTITE GRAPHS has complexity  $O(\sqrt{nm})$  [4], where  $m$  is the number of edges in  $G$ .

## 5 NP-completeness of strong stability in HRP

In this section we establish NP-completeness of STRONGLY STABLE MATCHING IN SMP, which is the problem of deciding whether a given instance of SMP admits a strongly stable matching. Here, SMP denotes the variant of SM in which each person's preferences over the members of the opposite sex are represented as an arbitrary partial order (hereafter this preference structure is referred to as a *preference poset*). Clearly SMP is a special case of HRP in which  $A = R \times H$  and each hospital has quota 1. It therefore follows immediately that the problem of deciding whether a given instance of HRP admits a strongly stable matching is also NP-complete.

To prove our result we give a reduction from the following problem:

*Name:* RESTRICTED SAT.

*Instance:* Boolean formula  $B$  in CNF, where each variable  $v$  occurs in exactly two clauses of  $B$  as literal  $v$ , and in exactly two clauses of  $B$  as literal  $\bar{v}$ .

*Question:* Is  $B$  satisfiable?

RESTRICTED SAT is NP-complete (see [8] for further details). We now state and prove the main result of this section.

**Theorem 2.** STRONGLY STABLE MATCHING IN SMP is NP-complete.

*Proof.* Clearly STRONGLY STABLE MATCHING IN SMP is in NP. To show NP-hardness, we give a polynomial reduction from RESTRICTED SAT, which is NP-complete as mentioned above. Let  $B$  be a Boolean formula in CNF, given as an instance of this, in which  $X = \{x_1, x_2, \dots, x_n\}$  is the set of variables and  $C = \{c_1, c_2, \dots, c_m\}$  is the set of clauses. For each  $i$  ( $1 \leq i \leq n$ ) and for each  $r$  ( $1 \leq r \leq 2$ ), let  $c(x_i^r)$  (respectively  $c(\bar{x}_i^r)$ ) denote the clause corresponding to the  $r$ th occurrence of literal  $x_i$  (respectively  $\bar{x}_i$ ).

We now construct an instance  $I$  of SMP, as follows. Let  $U = X^1 \cup X^2 \cup \bar{X}^1 \cup \bar{X}^2 \cup Z$  be the set of men in  $I$ , and let  $W = Y^1 \cup Y^2 \cup \bar{Y}^1 \cup \bar{Y}^2 \cup C$  be the set of women in  $I$ , where

$$\begin{aligned} X^r &= \{x_i^r : 1 \leq i \leq n\} \quad (1 \leq r \leq 2), & Y^r &= \{y_i^r : 1 \leq i \leq n\} \quad (1 \leq r \leq 2), \\ \bar{X}^r &= \{\bar{x}_i^r : 1 \leq i \leq n\} \quad (1 \leq r \leq 2), & \bar{Y}^r &= \{\bar{y}_i^r : 1 \leq i \leq n\} \quad (1 \leq r \leq 2), \\ Z &= \{z_i : 1 \leq i \leq m\}, & C &= \{c_i : 1 \leq i \leq m\}. \end{aligned}$$

Clearly  $|U| = |W| = 4n + m$ . Now, for each person  $p$  in  $I$ , we formulate  $\prec_p^*$ , the preference poset of  $p$ . In order to define  $\prec_p^*$ , we will construct a relation  $\prec_p$ , where  $q \prec_p r$  implies that  $p$  prefers  $q$  to  $r$ . We then obtain the partial order  $\prec_p^*$  by taking the transitive closure of  $\prec_p$ . Note that  $p$  is indifferent between  $q$  and  $r$  if and only if  $q, r$  are incomparable in  $\prec_p^*$  (i.e. neither  $q \prec_p^* r$  nor  $r \prec_p^* q$  holds). For each person  $q$  we will also define a subset  $P(q)$  of members of the opposite sex; if  $r \in P(q)$  we say that  $r$  is *proper* for  $q$ .

- *Preference poset of  $x_i^r$  ( $1 \leq i \leq n, 1 \leq r \leq 2$ ):*  $\bar{y}_i^1 \prec_{x_i^r} c(x_i^r), \bar{y}_i^2 \prec_{x_i^r} c(x_i^r), y_i^r \prec_{x_i^r} p$ , for every  $p \in W \setminus P(x_i^r)$ , where  $P(x_i^r) = \{c(x_i^r), y_i^r, \bar{y}_i^1, \bar{y}_i^2\}$ .

- Preference poset of  $\bar{x}_i^r$  ( $1 \leq i \leq n$ ,  $1 \leq r \leq 2$ ):  $y_i^1 \prec_{\bar{x}_i^r} c(\bar{x}_i^r)$ ,  $y_i^2 \prec_{\bar{x}_i^r} c(\bar{x}_i^r)$ ,  $\bar{y}_i^r \prec_{\bar{x}_i^r} p$ , for every  $p \in W \setminus P(\bar{x}_i^r)$ , where  $P(\bar{x}_i^r) = \{c(\bar{x}_i^r), \bar{y}_i^r, y_i^1, y_i^2\}$ .
- Preference poset of  $z_i$  ( $1 \leq i \leq m$ ):  $y \prec_{z_i} p$ , for every  $y \in P(z_i)$  and for every  $p \in W \setminus P(z_i)$ , where  $P(z_i) = Y^1 \cup Y^2 \cup \bar{Y}^1 \cup \bar{Y}^2$ .
- Preference poset of  $y_i^r$  ( $1 \leq i \leq n$ ,  $1 \leq r \leq 2$ ):  $x_i^r \prec_{y_i^r} \bar{x}_i^1$ ,  $x_i^r \prec_{y_i^r} \bar{x}_i^2$ . Let  $P(y_i^r) = \{x_i^r, \bar{x}_i^1, \bar{x}_i^2\} \cup Z$ .
- Preference poset of  $\bar{y}_i^r$  ( $1 \leq i \leq n$ ,  $1 \leq r \leq 2$ ):  $\bar{x}_i^r \prec_{\bar{y}_i^r} x_i^1$ ,  $\bar{x}_i^r \prec_{\bar{y}_i^r} x_i^2$ . Let  $P(\bar{y}_i^r) = \{\bar{x}_i^r, x_i^1, x_i^2\} \cup Z$ .
- Preference poset of  $c_i$  ( $1 \leq i \leq m$ ):  $\prec_{c_i} = \emptyset$ . Let  $P(c_i)$  contain those members of  $X^1 \cup X^2 \cup \bar{X}^1 \cup \bar{X}^2$  corresponding to the literal-occurrences in clause  $c_i$ .

It is easy to verify that, for any two people  $q, r$  of the opposite sex,  $r$  is proper for  $q$  if and only if  $q$  is proper for  $r$ .

Now suppose that  $B$  admits a satisfying truth assignment  $f$ . We form a matching  $M$  in  $I$  as follows. For each clause  $c_i$  in  $B$  ( $1 \leq i \leq m$ ), pick any literal-occurrence  $x \in X^1 \cup X^2 \cup \bar{X}^1 \cup \bar{X}^2$  corresponding to a true literal in  $c_i$ , and add  $(x, c_i)$  to  $M$ . For any  $x_i^r$  left unmatched ( $1 \leq i \leq n$ ,  $1 \leq r \leq 2$ ), add  $(x_i^r, y_i^r)$  to  $M$ . Similarly, for any  $\bar{x}_i^r$  left unmatched ( $1 \leq i \leq n$ ,  $1 \leq r \leq 2$ ), add  $(\bar{x}_i^r, \bar{y}_i^r)$  to  $M$ . Finally, there remain  $m$  members of  $Y^1 \cup Y^2 \cup \bar{Y}^1 \cup \bar{Y}^2$  that are as yet unmatched. Add to  $M$  a perfect matching between these women and the men in  $Z$ . It is straightforward to verify that  $M$  is strongly stable in  $I$ .

Conversely suppose that  $I$  admits a strongly stable matching  $M$ . Then it is not difficult to see that  $(m, w) \in M$  implies that  $w$  is proper for  $m$  and vice versa. Also, for each  $i$  ( $1 \leq i \leq n$ ),  $c_i$  is matched in  $M$  to some man  $x \in X^1 \cup X^2 \cup \bar{X}^1 \cup \bar{X}^2$  corresponding to an occurrence of a literal in clause  $c_i$  of  $B$ . Suppose that  $x = x_i^r$  for some  $i$  ( $1 \leq i \leq n$ ) and  $r$  ( $1 \leq r \leq 2$ ) (the argument is similar if  $x = \bar{x}_i^r$ ). Then by the strong stability of  $M$ ,  $(\bar{x}_i^1, \bar{y}_i^1) \in M$  and  $(\bar{x}_i^2, \bar{y}_i^2) \in M$ . Thus we may form a truth assignment  $f$  for  $B$  as follows: if  $x = x_i^r$  then set variable  $x_i$  to have value  $T$ , otherwise if  $x = \bar{x}_i^r$  then set variable  $x_i$  to have value  $F$ . Any remaining variable whose truth value has not yet been assigned can be set to  $T$ . Clearly  $f$  is a satisfying truth assignment for  $B$ .  $\square$

## 6 Conclusion and open questions

In this paper we have described a polynomial-time algorithm for the problem of finding a strongly stable matching, if one exists, given an instance of HRT. By contrast we have shown that the corresponding existence question becomes NP-complete for HRP. However, much remains to be investigated, and the following questions are particularly noteworthy:

1. For a given instance  $I$  of HRT that admits a strongly stable matching  $M$ , it is possible that  $I$  can admit weakly stable matchings of sizes  $> |M|$  and  $< |M|$  (see [8] for an example). However all strongly stable matchings for a given instance of HRT have the same cardinality [8]. In this paper, the robustness

of a strongly stable matching (against situations of persuasion or bribery) has been our primary motivation for studying HRT under strong stability. Nevertheless, further consideration should be given to the relative sizes of weakly stable matchings compared to the size of strongly stable matchings, given an instance of HRT.

2. The current algorithm for strong stability in HRT is resident-oriented. However, for super-stability in HRT there are both resident-oriented and hospital-oriented algorithms [7]. The problem of describing a hospital-oriented algorithm for HRT under strong stability remains open.
3. For a given instance of the stable marriage problem with ties, it is known that the set of strongly stable matchings forms a distributive lattice, when the set is partitioned by a suitable equivalence relation [11]. It remains open to characterise any similar structure in the set of strongly stable matchings for a given instance of HRT.

## References

1. Canadian Resident Matching Service. How the matching algorithm works. Web document available at <http://www.carms.ca/matching/algorithm.htm>.
2. D. Gale and L.S. Shapley. College admissions and the stability of marriage. *American Mathematical Monthly*, 69:9–15, 1962.
3. D. Gusfield and R.W. Irving. *The Stable Marriage Problem: Structure and Algorithms*. MIT Press, 1989.
4. J.E. Hopcroft and R.M. Karp. A  $n^{5/2}$  algorithm for maximum matchings in bipartite graphs. *SIAM Journal on Computing*, 2:225–231, 1973.
5. R.W. Irving. Stable marriage and indifference. *Discrete Applied Mathematics*, 48:261–272, 1994.
6. R.W. Irving. Matching medical students to pairs of hospitals: a new variation on a well-known theme. In *Proceedings of ESA '98*, volume 1461 of *Lecture Notes in Computer Science*, pages 381–392. Springer-Verlag, 1998.
7. R.W. Irving, D.F. Manlove, and S. Scott. The Hospitals/Residents problem with Ties. In *Proceedings of SWAT 2000*, volume 1851 of *Lecture Notes in Computer Science*, pages 259–271. Springer-Verlag, 2000.
8. R.W. Irving, D.F. Manlove and S. Scott. Strong Stability in the Hospitals/Residents Problem. Technical Report TR-2002-123, University of Glasgow, Computing Science Department, 2002.
9. K. Iwama, D. Manlove, S. Miyazaki, and Y. Morita. Stable marriage with incomplete lists and ties. In *Proceedings of ICALP '99*, volume 1644 of *Lecture Notes in Computer Science*, pages 443–452. Springer-Verlag, 1999.
10. D.F. Manlove. Stable marriage with ties and unacceptable partners. Technical Report TR-1999-29, University of Glasgow, Computing Science Department, 1999.
11. D.F. Manlove. The structure of stable marriage with indifference. *Discrete Applied Mathematics*, 122:167–181, 2002.
12. D.F. Manlove, R.W. Irving, K. Iwama, S. Miyazaki, and Y. Morita. Hard variants of stable marriage. *Theoretical Computer Science*, 276:261–279, 2002.
13. M. Mukerjee. Medical mismatch. *Scientific American*, 276(6):40–41, 1997.
14. A.E. Roth. The evolution of the labor market for medical interns and residents: a case study in game theory. *Journal of Political Economy*, 92(6):991–1016, 1984.