Approximation algorithms for hard variants of the stable marriage and hospitals/residents problems[*,†]

Robert W. Irving and David F. Manlove

Department of Computing Science, University of Glasgow, Glasgow G12 8QQ, UK
Email: `rwi@dcs.gla.ac.uk`, `davidm@dcs.gla.ac.uk`.

Abstract

When ties and incomplete preference lists are permitted in the Stable Marriage and Hospitals/Residents problems, stable matchings can have different sizes. The problem of finding a maximum cardinality stable matching in this context is known to be NP-hard, even under very severe restrictions on the number, size and position of ties. In this paper, we describe polynomial-time 5/3-approximation algorithms for variants of these problems in which ties are on one side only and at the end of the preference lists. The particular variant is motivated by important applications in large scale centralised matching schemes.

1 Introduction

Background

An instance I of the Hospitals/Residents problem with Ties (HRT) comprises a set $R = \{r_1, \ldots, r_{n_1}\}$ of *residents* and a set $H = \{h_1, \ldots, h_{n_2}\}$ of *hospitals*. Each resident $r_i \in R$ has an *acceptable* set of hospitals that she ranks in order of preference, with ties (consisting of two or more hospitals of equal preference) being permitted, forming r_i's *preference list*. Meanwhile each hospital $h_j \in H$ ranks in order of preference those residents who find h_j acceptable, again with ties being permitted, yielding h_j's preference list. Finally, h_j has an associated *capacity*, denoted by $c_j \in \mathbb{Z}^+$, indicating the number of *posts* that h_j has. If a resident r_i and a hospital h_j appear on each other's preference list then (r_i, h_j) is called an *acceptable pair*. If h_j precedes h_k on r_i's list then r_i is said to *prefer* h_j to h_k (with *prefer* being defined analogously in the case of a hospital).

An *assignment* M is a set of acceptable pairs. If $(r_i, h_j) \in M$, we say that r_i is *assigned* to h_j, and h_j is *assigned* r_i. For any $q \in R \cup H$, we denote by $M(q)$ the set of assignees of q in M. If $r_i \in R$ and $M(r_i) = \emptyset$, we say that r_i is *unassigned*, otherwise r_i is *assigned*. Similarly, a hospital $h_j \in H$ is *under-subscribed*, *full*, or *over-subscribed* according as $|M(h_j)|$ is less than, equal to, or greater than c_j, respectively.

A *matching* M is an assignment such that $|M(r_i)| \leq 1$ for each $r_i \in R$ and $|M(h_j)| \leq c_j$ for each $h_j \in H$ (i.e., each resident is assigned to at most one hospital, and no hospital is over-subscribed). For convenience, given a resident $r_i \in R$ such that $M(r_i) \neq \emptyset$, the notation $M(r_i)$ is also used to refer to the single member of the set $M(r_i)$.

An acceptable pair (r_i, h_j) is a *blocking pair* for a matching M, or *blocks* M, if r_i is either unassigned in M or prefers h_j to $M(r_i)$, and simultaneously h_j is either under-subscribed in M or prefers r_i to at least one member of $M(h_j)$. A matching for which

[*] A preliminary version of this paper appeared in the Proceedings of COCOON 2007 [14].

[†] This work was supported by EPSRC grant EP/E011993/1.

there is no blocking pair is said to be *stable*. (This notion of stability is also referred to as *weak* stability in contexts where other forms of stability are discussed [11, 15, 16].)

HRT is an extension of the classical Hospitals/Residents problem (HR) (or College Admissions problem) introduced by Gale and Shapley [4]. In an instance of HR, all preference lists are strictly ordered. The classical Stable Marriage problem (SM) can be viewed as a restriction of HR in which each hospital has capacity one, $n_1 = n_2 = n$ and the preference lists are *complete* (i.e., the set of acceptable pairs is precisely $R \times H$). In such a case, the residents and hospitals are more commonly referred to as the *men* and *women* respectively.

Gale and Shapley proved that, for every instance of SM, there is at least one stable matching, and they described an $O(n^2)$ time algorithm to find such a matching; this has come to be known as the *Gale-Shapley algorithm*. This algorithm is easily extended to the case in which the numbers of men and women differ and preference lists are incomplete (SMI – Stable Marriage with Incomplete lists, which is a special case of HR in which each hospital has capacity one), and it has complexity $O(L)$ in this case, where L is the sum of the lengths of all of the preference lists [6]. In the case of SMI, not all men and women need be assigned in a stable matching.

The Gale-Shapley algorithm may be applied from either the men's side or the women's side, and in general these two applications will produce different stable matchings. When applied from the men's side, the *man-optimal* stable matching is found; in this case, every man has the best assignee that he can have in any stable matching, and every woman the worst. When the algorithm is applied from the women's side, the *woman-optimal* stable matching results, with analogous properties. Exceptionally, the man-optimal and woman-optimal stable matchings may coincide, in which case this is the unique stable matching, but in general there may be other stable matchings – possibly exponentially many [13] – between these two extremes. However, for a given instance of SMI, all stable matchings have the same size and assign exactly the same sets of men and women [5, 22].

Two variants of the Gale-Shapley algorithm have also been formulated for the HR context [6, Section 1.6], with the complexity also being $O(L)$ in each case, producing the *resident-optimal* and *hospital-optimal* stable matchings, with analogous optimality properties to those of the man-optimal and woman-optimal stable matchings in the SM case, respectively. For a given instance of HR, all stable matchings have the same size, assign exactly the same set of residents, and fill the same number of posts at each hospital [5, 22].

The situation for HRT is dramatically different. Again, at least one stable matching exists for every instance, and can be found in $O(L)$ time by breaking all ties in an arbitrary way to give an instance of HR, and applying the Gale-Shapley algorithm to that instance. However, the ways in which ties are broken can significantly affect the outcome. In particular, not all stable matchings need be of the same size, and in the most extreme case, there may be two stable matchings M and M' with $|M| = 2|M'|$ [20]. Furthermore, the problem of finding a stable matching of maximum cardinality for an instance of HRT – problem MAX-HRT – is NP-hard [20]. This hardness result holds even if each hospital has capacity one – in other words, for the so-called Stable Marriage problem with Ties and Incomplete lists (SMTI), thus establishing NP-hardness of the problem MAX-SMTI. Furthermore, the result holds even under severe restrictions on the ties: for example, if the ties are on one side only, each list contains at most one tie, and that tie, if present, is at the end of the list [20].

Practical applications

The practical importance of stable matching problems arises from their application in the assignment of applicants to positions in various job markets. The Hospitals/Residents problem gained its name because of its widespread application in the medical employment domain [1, 12, 21, 22, 24].

Variants of the extended Gale-Shapley algorithm for HR are routinely used in a number of countries, including the United States [21], Canada [1] and Scotland [24], to allocate graduating medical students to hospital posts, and in a variety of other countries and contexts. In large scale matching schemes of this kind, participants, particularly large popular hospitals, may not be able to provide a genuine strict preference order over what may be a very large number of applicants, so that HRT is a more appropriate model than HR. If artificial tie-breaking is carried out, either by the participant, because a strictly ordered list is required by the matching scheme, or by the administrators of the scheme, prior to running an algorithm that requires strict preferences, then the size of the resulting stable matching is likely to be affected. Breaking ties in different ways will typically yield stable matchings of different sizes; what would be ideal would be to find a way of breaking the ties that maximises the size of the resulting stable matching, but the NP-hardness of this problem makes this an objective that is unlikely to be feasible.

One restricted case of HRT that has arisen in practice, for example in the Scottish Foundation Allocation Scheme (SFAS) [12, 24], is where each resident's preference list is strictly-ordered, and each hospital's list may contain one tie, of arbitrary length, at the end of the list. The correspondingly restricted version of SMTI, where each man's list is strict and each woman's list may contain one tie at the end, is, of course, the special case of the restricted HRT problem in which all hospital capacities are equal to one. We refer to these restricted versions of SMTI and HRT as *Special SMTI/HRT* (SSMTI/SHRT), and use the terms MAX-SSMTI and MAX-SHRT for the problems of finding maximum cardinality stable matchings in these cases, which remain NP-hard problems [20].

Figure 1 shows an example of SSMTI in which there are two stable matchings, $M_1 = \{(m_1, w_2), (m_2, w_1)\}$ of size 2 and $M_2 = \{(m_1, w_1)\}$ of size 1. A tie in the preference lists is indicated by parentheses.

Men's preferences	Women's preferences
m_1 : w_1 w_2	w_1 : (m_1 m_2)
m_2 : w_1	w_2 : m_1

Figure 1: An instance of SSMTI with stable matchings of sizes 1 and 2.

Related results

It is straightforward to establish that there can be at most a factor of two difference between the sizes of a maximum cardinality stable matching and a minimum one for an instance of SMTI or HRT [20], and as a consequence, breaking ties arbitrarily and applying the appropriate version of the Gale-Shapley algorithm gives 2-approximation algorithms for MAX-SMTI and MAX-HRT. A number of improved approximation algorithms for versions of MAX-SMTI have recently been proposed.

For the general case, Iwama et al. [17] gave an algorithm with a performance guarantee of $2 - (c \log n)/n$, for the case of n men and n women, where c is a positive constant. This was later improved to $2 - c'/\sqrt{n}$ [18], where c' is a positive constant such that

$c' \leq 1/4\sqrt{6}$. Very recently, Iwama et al. [19] gave the first approximation algorithm for the general case with a constant performance guarantee better than 2, namely 15/8. From the inapproximability point of view, Halldórsson et al. [7] showed the problem to be APX-complete, and Halldórsson et al. [8] gave a lower bound of 21/19 on any polynomial-time approximation algorithm (assuming $P \neq NP$). This lower bound applies even to MAX-SSMTI.

As far as special cases are concerned, Halldórsson et al. [8] gave a $(2/(1 + T^{-2})$-approximation algorithm for the case where all ties are on one side, and are of length at most T – so, for example, this gives a bound of 8/5 when all ties are of length 2. If ties are on both sides and restricted to be of length 2, a bound of 13/7 is shown in [8]. Halldórsson et al. [9] also described a randomised algorithm with an expected guarantee of 10/7 for the same special case.

'Cloning' hospitals

It is known that, by identifying residents with men, and 'cloning' each hospital into c women, where c is the hospital's capacity, an instance I of HR may be transformed in linear time to an instance I^* of SMI such that there is a bijective function between the set of stable matchings in I and those in I^* [6, p.38] (see also [23, pp.131-132]). A formal proof of this result is given in [2, Lemma 3.1] for the Stable Multiple Activities problem, a many-to-many non-bipartite generalisation of HR. It is not difficult to extend this proof to the HRT case, thus giving a correspondence with SMTI. To be more precise, it can be shown that an instance I of HRT may be transformed in linear time to an instance I^* of SMTI with the property that there is a function, surjective in this case, between the set of stable matchings in I and those in I^*, under which matching cardinality is preserved.

An immediate consequence of this correspondence between HRT and SMTI is that, given an approximation algorithm A for MAX-SMTI with performance guarantee δ, for some constant $\delta > 1$, we may obtain (except in certain cases) an approximation algorithm for MAX-HRT with the same performance guarantee, simply by applying A to I^* and mapping the obtained stable matching S^* in I^* to a stable matching S in I such that $|S| = |S^*|$. The special cases that cause an exception to this occur when A depends on certain properties of the preference lists which are not preserved under 'cloning': an example of such a property is the length of the ties in the residents' lists, which are in general inflated under such a transformation. This implies that the approximation algorithms for MAX-SMTI mentioned in the previous subsection, except for those with performance guarantees 13/7 and 10/7 [8, 9], also yield approximation algorithms for corresponding versions of MAX-HRT with the same performance guarantees.

The contribution of this paper

In this paper, we present a polynomial-time 5/3-approximation algorithm for MAX-SHRT, thus improving on the 15/8 bound for the general MAX-HRT case [19]. Since MAX-SSMTI is a special case of MAX-SHRT, the algorithm also applies to the former problem, and we point out some simplifications that apply in the one-to-one case. We also show that this guarantee is the best that can be proved for the algorithm, either for MAX-SSMTI or for MAX-SHRT, by providing a family of instances of the former problem for which this bound is realised.

In view of the observations regarding 'cloning' hospitals in the previous subsection, it would have been sufficient to present only the MAX-SSMTI version of our approximation algorithm. However neither the extension of the algorithm to the MAX-SHRT case, nor the derivation of its performance guarantee, are significantly more difficult than for the

MAX-SSMTI case. We therefore choose to focus on MAX-SHRT in view of the fact that the above 'cloning' transformation has the undesirable property that the number of women in I^* is greater than the number of hospitals in I by a factor of C/n_2, where C is the sum of the hospital capacities in I and n_2 is the number of hospitals in I. Moreover, if the minimum hospital capacity in I is c, then the total length of the preference lists in I^* is greater than the total length of those in I by a factor of $\Omega(c)$. For these reasons, in practical applications such as SFAS and its counterparts in the US and Canada [24, 21, 1], direct algorithms for finding stable matchings in HRT are routinely employed without resorting to 'cloning' together with the use of algorithms for SMTI. Moreover, a variant of our algorithm for MAX-SHRT has already been implemented and is utilised by SFAS. It is thus with algorithmic efficiency in mind that we present a direct approach for MAX-SHRT.

We finally remark that, in a preliminary version of this paper [14], an approximation algorithm for MAX-SSMTI was presented with a claimed performance guarantee of 8/5. That bound turned out to be incorrect. This paper not only establishes the correct bound of 5/3 for the approximation algorithm in the MAX-SSMTI case, but also extends the algorithm and the derivation of the same performance guarantee to the MAX-SHRT case.

2 The approximation algorithm for MAX-SHRT

Algorithm SHRT-APPROX Phase 1

Let I be an instance of SHRT; recall that in I, each resident's preference list is strict, and each hospital's preference list is strict except for a tie (of length ≥ 1) at the end. Henceforth we assume the notation and terminology as defined for an instance of HRT in the previous section.

The first phase of the algorithm is a variant of the hospital-oriented Gale-Shapley algorithm for the standard Hospitals/Residents problem [6, Section 1.6.2]. During this phase, zero or more deletions are made from the preference lists – by the deletion of the pair (r_i, h_j), we mean the removal of h_j from r_i's list and the removal of r_i from that of h_j. Initially, every resident is free and every hospital is assigned the empty set of residents. Throughout, a_j denotes the number of residents currently assigned to hospital h_j. As long as there is a hospital h_j for which $a_j < c_j$ and which has more than a_j untied residents on its current list, the resident r_i in position $a_j + 1$ is (at least temporarily) assigned to h_j, and a_j is incremented. If r_i was already assigned, say to h_k then that assignment is broken, and a_k is decremented. Then all pairs (r_i, h_l) such that r_i prefers h_j to h_l are deleted. This phase of the algorithm is summarised in Figure 2.

We now use a_j to denote the number of residents assigned to hospital h_j on termination of Phase 1 of the algorithm, and we let b_j be the total number of untied residents remaining in the preference list of hospital h_j. Then it follows from the loop condition that $a_j = \min(b_j, c_j)$. A resident r_i can be assigned to at most one hospital, so that if r_i is one of the first a_j residents on the list of hospital h_j, she cannot be one of the first a_k residents on the list of h_k for any $h_k \neq h_j$. Hence the number of assigned residents is $x = \sum_{j=1}^{n_2} a_j$. We denote this set of assigned residents by X, and call its members the *X-residents*. The remaining residents form the set $\overline{X}$, and are the *$\overline{X}$-residents*. At each point during the algorithm's execution, the current preference lists are referred to as the *reduced* lists.

Lemma 2.1. *On termination of Phase 1 of Algorithm SHRT-APPROX,*
(i) no deleted pair can belong to a stable matching;
(ii) for each hospital h_j, the first a_j residents on h_j's reduced list are assigned in every stable matching, i.e., each X-resident is assigned in every stable matching.

```
assign each resident to be free;
for (each hospital $h_j \in H$)
    $a_j = 0$;
while (there exists a hospital $h_j$ such that $a_j < c_j$
        and $h_j$ has $> a_j$ untied residents on its list)
{
    $r_i$ = resident in position $a_j + 1$ on $h_j$'s list;
    assign $r_i$ to $h_j$;
    $a_j$++;
    for (each successor $h_k$ of $h_j$ on $r_i$'s list)
    {
        if ($r_i$ is assigned to $h_k$)
        {
            break the assignment;
            $a_k$--;
        }
        delete the pair $(r_i, h_k)$ from the preference lists;
    }
}
```

Figure 2: Phase 1 of Algorithm SHRT-APPROX

Proof. (i) Suppose that (r_i, h_j) is a deleted pair that belongs to a stable matching M, and that (r_i, h_j) was the first such pair to be deleted in an execution of Phase 1 of the algorithm. This must have happened because r_i became assigned to a hospital h_l whom she prefers to h_j. Hospital h_l either is under-subscribed in M or prefers r_i to at least one member of the set $M(h_l)$. For suppose h_l is full in M and prefers all its assignees in M to r_i. Then there exists a pair (r_k, h_l) in M, already deleted, such that h_l prefers r_k to r_i, contradicting the fact that (r_i, h_j) was the first pair of M to be deleted. Hence (r_i, h_l) blocks M, a contradiction.
(ii) Suppose that r_i is one of the first a_j residents at the head of hospital h_j's reduced list, and that M is a stable matching in which r_i is unassigned. Then, by part (i), h_j is either under-subscribed in M or prefers r_i to at least one member of $M(h_j)$, so that (r_i, h_j) blocks M, a contradiction. □

A refinement of Phase 1 is possible. If, at any stage, the total number of residents remaining on the preference list of a hospital h_j is at most c_j, then we may assign all of these residents to h_j, including any that are in h_j's tie. However, this refinement has no bearing on what happens in the worst case, so for simplicity we choose to omit it.

We call the preference lists that remain after Phase 1 the *Phase-1 lists*, and if resident r_i and hospital h_j are in each other's Phase-1 lists, we say that (r_i, h_j) is a *Phase 1 acceptable pair.*

Algorithm SHRT-APPROX Phase 2

We define the *Phase 2 capacity* c'_j of a hospital h_j to be $c'_j = c_j - a_j$. In this phase of the algorithm, we seek to increase the number of residents who are guaranteed to be assigned. To this end, we find a maximum cardinality many-to-one matching, or *degree-constrained subgraph* [3], K of residents to hospitals, where a pair (r_i, h_j) can be in K if and only if (i) r_i is an $\overline{X}$-resident, (ii) (r_i, h_j) is a Phase 1 acceptable pair, and (iii) r_i

is in the tie in h_j's list. Indeed, it is sufficient to consider only those hospitals H' such that $c'_j > 0$. Then, for any resident $r_i \in \overline{X}$ and for any hospital $h_j \in H'$, it is immediate that h_j's tie must be non-empty. The cardinality of the subgraph K is maximised with respect to the upper degree-constraining function $d : R \cup H' \longrightarrow \mathbb{N}$, where $d(r_i) = 1$ for each $r_i \in R$ and $d(h_j) = c'_j$ for each $h_j \in H'$. That is, hospital h_j can be assigned up to c'_j residents, but of course each resident to at most one hospital. Having found such a degree-constrained subgraph K, we break the tie in each hospital h_j's Phase-1 list by promoting those residents r_i, if any, who are assigned to h_j in K, in arbitrary order, ahead of the rest of that tie (leaving the rest of the tie intact). This produces an instance I' of SHRT that is a refinement of the original instance I – or more properly, a refinement of the variant of I that results following the deletions of Phase 1; clearly any matching that is stable for I' is also stable for I, but not necessarily vice-versa. Phase 2 of the algorithm is summarised in Figure 3. (Note that, in the special case of MAX-SSMTI, since all capacities are 1, the maximum cardinality degree-constrained subgraph is just a standard maximum cardinality matching.)

```
$V = \overline{X} \cup H'$;
$E = \{(r_i, h_j) \in \overline{X} \times H' :$ $(r_i, h_j)$ is a Phase 1 acceptable pair and $r_i$ is in $h_j$'s tie$\}$;
construct the bipartite graph $G = (V, E)$;
$K$ = maximum cardinality degree-constrained subgraph in $\langle G, d \rangle$ ;
for (each pair $(r_i, h_j) \in K$)
    promote $r_i$ ahead of the tie in $h_j$'s list;
reactivate the loop of Phase 1;
```

Figure 3: Phase 2 of Algorithm SHRT-APPROX

Lemma 2.2. *On termination of Phase 2 of Algorithm SHRT-APPROX, for each hospital h_j, every untied resident who is among the first c_j residents on h_j's list is assigned in every stable matching for I'.*

Proof. The proof is completely analogous to that of Lemma 2.1(ii). □

Note that, while Lemma 2.2 can be expected, in many cases, to give a stronger lower bound on the size of a stable matching than is given by Lemma 2.1, this need not be the case. It is possible that, after Phase 1, each hospital with a positive Phase 2 capacity has only X-residents in its tie, and that, as a consequence, K is the empty set. However, to take account of the general case where K is non-empty, we now partition $\overline{X}$ to reflect this fact. More precisely, we define Y to be the set of residents who are assigned in K, and define $Z = \overline{X} \setminus Y$, so that X, Y and Z constitute a partition of the set of residents. The residents in sets Y and Z are referred to as Y-residents and Z-residents respectively. We denote $|X|$, $|Y|$ and $|Z|$ by x, y and z respectively. Let P denote the set of *hospital posts* in I, say

$$P = \{p_{j,k} : 1 \leq j \leq n_2 \wedge 1 \leq k \leq c_j\}.$$

(Of course, in reality, the hospital posts are indistinguishable, but it is convenient, for our purposes, to label them.)

The next lemma gives a precise consequence of the fact that the degree-constrained subgraph K found in Phase 2 has maximum possible cardinality.

Lemma 2.3. *Let M be a maximum cardinality stable matching for I. Then $|M| \leq 2x + y$.*

Proof. Define a subset Q of P as follows:

$$Q = \{p_{j,k} : 1 \leq j \leq n_2 \wedge 1 \leq k \leq a_j\}.$$

Then as observed in the paragraph before Lemma 2.1, $|Q| = \sum_{j=1}^{n_2} a_j = x = |X|$. (In fact we can construct a perfect matching A_1^* of X to Q as follows: if r_i is the kth-best assignee of h_j upon termination of Phase 1, add $(r_i, p_{j,k})$ to A_1^*.)

Define $\overline{Q} = P \backslash Q$. During Phase 2, a maximum matching K of $\overline{X}$ to H' is found, with respect to the capacities $c'_j = c_j - a_j$ for each hospital $h_j \in H'$. This gives rise to a matching K^* of $\overline{X}$ to $\overline{Q}$ which can be seen as follows: with respect to an instance I^* of HR obtained from I by arbitrarily breaking ties, if r_i is the kth-best assignee of h_j in K, add (r_i, p_{j,a_j+k}) to K^*. Then $|K| = |K^*| = y$, and K^* is a maximum matching of $\overline{X}$ to $\overline{Q}$.

Now define M^* from M as follows: if r_i is the kth-best assignee of h_j in M (with respect to the same tie-break instance I^*), add $(r_i, p_{j,k})$ to M^*. Then $|M^*| = |M|$. Moreover for each pair $(r_i, p_{j,k})$ in M^*, at least one of the following holds: (a) $r_i \in X$; (b) $p_{j,k} \in Q$; or (c) $r_i \in \overline{X}$ and $p_{j,k} \in \overline{Q}$. The number of pairs satisfying (a) is x, the number satisfying (b) is also x, and the number satisfying (c) is at most y, since y is the size of a maximum matching of $\overline{X}$ to $\overline{Q}$. □

Algorithm SHRT-APPROX Phase 3

Phase 3 of the algorithm involves completely breaking the remaining ties and then applying to the resulting instance of HR the standard Gale-Shapley algorithm (in fact, for consistency with Phases 1 and 2, the extended version of that algorithm that deletes redundant entries from the preference lists – see [6]). The algorithm may be applied from either the residents' or the hospitals' side; as is well known, the size of the resulting matching will be the same in each case. We choose to apply the algorithm in Phase 3 from the residents' side, as is typically done by practical matching schemes that incorporate a variant of the Gale-Shapley algorithm.

Tie-breaking is carried out according to just one restriction, namely, for each tie, the Z-residents are given priority over the X-residents and Y-residents. In other words, each tie is resolved by listing the Z-residents that it contains, in arbitrary order, followed by the X-residents and Y-residents that it contains, again in arbitrary order. (Intuitively, because we know that X-residents and Y-residents will be assigned no matter what, we want to maximise the chances of Z-residents being assigned.) It is immediate that the algorithm produces a matching that is stable for the original instance of HRT. For an instance I of HRT, we denote by I'' an instance of HR obtained by application of Phases 1 and 2 of Algorithm SHRT-APPROX, followed by tie-breaking according to this rule. Again it is immediate that a matching that is stable for I'' is also stable for I. Phase 3 of the algorithm is summarised in Figure 4.

The performance guarantee

Let A be a matching produced by application of Algorithm SHRT-APPROX, and let M be a maximum cardinality stable matching for the original instance I of HRT. As previously established, all of the X-residents and Y-residents are assigned in A. Let us call the residents, necessarily Z-residents, who are assigned in M but not in A, the *extra residents* (for M relative to A). Also, a hospital h_j that has at least one extra resident assigned to it in M will be called a *key hospital.*

Lemma 2.4. *(i) Each key hospital is full in A.*
(ii) A key hospital prefers, according to its preference list in I, all of its A-assignees who are X-residents, if any, to any extra resident.

```
for (each hospital $h_j$)
    break the tie (if any) in $h_j$'s list, placing the $Z$-residents ahead of the others;
// Now apply the standard Gale-Shapley algorithm from the residents' side
assign each resident to be free;
for (each hospital $h_j$)
    $a_j = 0$;
while (some resident $r_i$ is free and $r_i$ has a non-empty list)
{
    $h_j$ = first hospital on $r_i$'s list;
    assign $r_i$ to $h_j$;
    $a_j$++;
    if ($a_j \geq c_j$)
    {
        $r_k$ = resident ranked in position $c_j$ among those assigned to $h_j$;
        for (each successor $r_l$ of $r_k$ on $h_j$'s list)
        {
            if ($r_l$ is assigned to $h_j$)
            {
                break the assignment;
                $a_j$--;
            }
            delete the pair $(r_l, h_j)$ from the preference lists;
        }
    }
}
return the set $A$ of assigned pairs;
```

Figure 4: Phase 3 of Algorithm SHRT-APPROX

Proof. Let h_j be a key hospital, and let r_i be an extra resident (necessarily a Z-resident) who is assigned to h_j in M. Recall that M is stable for the original instance I, while A is stable for the refined instance I'' (of HR), and therefore also for the instances I' and I (of SHRT).
(i) By definition, r_i is not assigned in A, so that if h_j is not full in A then the pair (r_i, h_j) blocks A.
(ii) Let r_k be an X-resident who is assigned in A to h_j. We note that the pair (r_k, h_j) could not have been deleted by Phases 1 or 2 of the algorithm, since it is in A, and nor could the pair (r_i, h_j), since r_i is in Z and only pairs (r, h) with $r \in X \cup Y$ are so deleted.

If h_j prefers r_i to r_k then the pair (r_i, h_j) blocks A in I, a contradiction. If r_i and r_k are tied in h_j's list then, as argued above, that tie was still intact at the start of Phase 3, and when it was broken to form I'', r_i, being a Z-resident, must have preceded r_k in the resulting strict preference list. Hence, since r_i is unassigned in A, the pair (r_i, h_j) blocks A in I'', a contradiction. □

Let M be a maximum stable matching in I. Fix an arbitrary tie-break instance I^* of I. Form a matching M^* as follows: if $(r_i, h_j) \in M$ and r_i is the kth-best assignee of h_j in M according to the preferences in I^*, add $(r_i, p_{j,k})$ to M^*. Then $|M| = |M^*|$. Form A^* from A in an analogous way (with respect to the same tie-break instance I^*). Then $|A| = |A^*|$.

If $(r_i, p_{j,k}) \in M^*$ and r_i is an extra resident, define $p_{j,k}$ to be an *extra hospital post.* By Lemma 2.4(i), each extra hospital post is assigned in A^*. Partition the extra residents into three sets, namely T, U and V according as the post they are assigned in M^* is assigned in A^* to an X-resident, a Y-resident or a Z-resident, respectively. Let $|T| = t$, $|U| = u$ and $|V| = v$.

Our next lemma gives us certain inequalities involving the sizes of matchings M and A that will enable us to establish the claimed performance guarantee for Algorithm SHRT-APPROX.

Lemma 2.5. *(i)* $|M| \leq |A| + t + u + v$.
(ii) $|A| \geq x + y + v$.
(iii) $|A| \geq 2t + u + v$.

Proof. (i) The number of extra residents is $t + u + v$, and, by definition, these are the only residents who are assigned in M but not in A.
(ii) Each extra hospital post $p_{j,k}$ that is assigned in M^* to a resident in V is assigned in A^* to a Z-resident. This gives us v of the Z-residents who are assigned in A, and together with all $x + y$ of the X-residents and Y-residents who, by Lemma 2.2, are assigned in A, we have a total of $x + y + v$ distinct residents who are assigned in A.
(iii) By Lemma 2.4(i), all $t + u + v$ extra hospital posts are assigned in A^*. Let r_i be a resident in T, and let h_j be the hospital that r_i is assigned to in M. By definition of T, h_j has an X-resident, say r_k, as an assignee in A. By Lemma 2.4(ii), h_j prefers r_k to r_i. Now the X-residents are assigned in every stable matching, so in particular are assigned in M. Suppose that $(r_k, p_{l,a}) \in M^*$. Then $p_{l,a}$ cannot be an extra hospital post, since the residents assigned to extra hospital posts in M^* are Z-residents. We claim that $p_{l,a}$ is assigned in A^*. This is true if $j = l$ by Lemma 2.4(i), hence suppose that $j \neq l$. It follows that r_k prefers h_l to h_j, for otherwise (r_k, h_j) would block M. Furthermore, $p_{l,a}$ must be assigned in A^*, for if not, the pair (r_k, h_l) would block A. Hence the claim is established, and gives a further t hospital posts (none of which are extra hospital posts as previously observed) that are guaranteed to be assigned in A^*, and together with the extra hospital posts, a total of $2t + u + v$ such hospital posts. □

We are now in a position to establish our main theorem.

Theorem 2.1. *For a given instance of SHRT, or of the special case SSMTI, let M be a maximum cardinality stable matching and let A be a stable matching returned by Algorithm SHRT-APPROX. Then $|M| \leq 5|A|/3$.*

Proof. From Lemma 2.3 and Lemma 2.5 part (ii) we have

$$|M| - |A| \leq x - v. \tag{1}$$

By Lemma 2.5 parts (ii) and (iii) we have

$$|A| \geq \max(x + y + v, 2t + u + v) \geq \frac{1}{2}((x + y + v) + (2t + u + v)) \geq x/2 + t + u + v,$$

the third inequality following from the fact that residents in U are assigned in M^* to a post that is assigned in A^* to a Y-resident, so that $u \leq y$. By Lemma 2.5 part (i), this implies

$$|A| \geq |M| - |A| + x/2. \tag{2}$$

Combining (1) and (2) gives

$$|A| \geq |M| - |A| + |M|/2 - |A|/2 + v/2,$$

and this, together with the fact that v is non-negative, implies the claimed result. □

Complexity of the algorithm

The worst-case complexity of Algorithm SHRT-APPROX is dominated by the maximum cardinality degree-constrained subgraph step in Phase 2. Using Gabow's algorithm [3], this can be achieved in $O(\sqrt{n_1 + C}L)$ time, where n_1 is the number of residents, C is the sum of the hospital capacities, and L is the sum of the lengths of the preference lists.

In the special case of SSMTI, Phase 2 is standard maximum cardinality bipartite matching. Using the Hopcroft-Karp algorithm [10], this can be achieved in $O(\sqrt{n}L)$ time, where n is the total number of men and women, and L is the sum of the lengths of the preference lists.

Tightness of the performance guarantee

This is the tightest bound that can be established for Algorithm SHRT-APPROX, even in the one-to-one case of SSMTI. Figure 5 shows an SSMTI instance where the ratio of $|M|$ to $|A|$ is 5/3. The matching $M = \{(m_1, w_4), (m_2, w_5), (m_3, w_1), (m_4, w_3), (m_5, w_2)\}$ is a maximum cardinality stable matching of size 5. When the algorithm is applied to this instance, there are no deletions in Phase 1. The pairs resulting from this phase are (m_1, w_1) and (m_2, w_2), so that $X = \{m_1, m_2\}$. In Phase 2, a feasible maximum cardinality degree-constrained subgraph (or maximum cardinality matching in this case) is $K = \{(m_3, w_3)\}$, and if this is chosen we have $Y = \{m_3\}$ and $Z = \{m_4, m_5\}$. Phase 3 could break the remaining ties in favour of m_2, in which case it can be verified that the algorithm would return the matching $A = \{(m_1, w_2),\ (m_2, w_4), (m_3, w_3)\}$, of size 3. By duplicating this pattern, we can obtain arbitrarily large instances realising the 5/3 ratio.

Men's preferences	Women's preferences
$m_1: \ w_4 \ w_2 \ w_5 \ w_1$	$w_1: \ m_1 \ m_3$
$m_2: \ w_4 \ w_5 \ w_2$	$w_2: \ m_2 \ m_1 \ m_5$
$m_3: \ w_3 \ w_1$	$w_3: \ (\ m_3 \ m_4\ )$
$m_4: \ w_3$	$w_4: \ (\ m_1 \ m_2\ )$
$m_5: \ w_2$	$w_5: \ (\ m_1 \ m_2\ )$

Figure 5: An instance of SSMTI with ratio 5/3

3 Summary and open problems

We have described a polynomial-time approximation algorithm with a performance guarantee of 5/3 for a maximum cardinality stable matching in an NP-hard variant of the Hospitals/Residents problem that is of significant practical interest, which has a slightly simplified version for the corresponding variant of the Stable Marriage problem. We have also shown that this performance guarantee is the best that can be proved for these algorithms.

The most obvious open question to pursue is whether this or a similar approach can yield useful performance guarantees for more general versions of HRT and SMTI, for example when there can be a single tie at the end of the lists on both sides, or when the lists on one side can contain arbitrary ties.

Acknowledgement

The authors would like to thank Eric McDermid for suggesting the use of a reduction from HRT to SMTI involving 'cloning' hospitals.

References

[1] `http://www.carms.ca/jsp/main.jsp` (Canadian Resident Matching Service website).

[2] K. Cechlárová and T. Fleiner. On a generalization of the stable roommates problem. *ACM Transactions on Algorithms*, 1(1):143–156, 2005.

[3] H.N. Gabow. An efficient reduction technique for degree-constrained subgraph and bidirected network flow problems. In *Proceedings of STOC '83: the 15th Annual ACM Symposium on Theory of Computing*, pages 448–456. ACM, 1983.

[4] D. Gale and L.S. Shapley. College admissions and the stability of marriage. *American Mathematical Monthly*, 69:9–15, 1962.

[5] D. Gale and M. Sotomayor. Some remarks on the stable matching problem. *Discrete Applied Mathematics*, 11:223–232, 1985.

[6] D. Gusfield and R.W. Irving. *The Stable Marriage Problem: Structure and Algorithms.* MIT Press, 1989.

[7] M. Halldórsson, R.W. Irving, K. Iwama, D.F. Manlove, S. Miyazaki, Y. Morita, and S. Scott. Approximability results for stable marriage problems with ties. *Theoretical Computer Science*, 306(1-3):431–447, 2003.

[8] M. Halldórsson, K. Iwama, S. Miyazaki, and H. Yanagisawa. Improved approximation of the stable marriage problem. In *Proceedings of ESA 2003: the Eleventh European Symposium on Algorithms*, volume 2832 of *Lecture Notes in Computer Science*, pages 266–277. Springer, 2003.

[9] M.M. Halldórsson, K. Iwama, S. Miyazaki, and H. Yanagisawa. Randomized approximation of the stable marriage problem. *Theoretical Computer Science*, 325(3):439–465, 2004.

[10] 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.

[11] R.W. Irving. Stable marriage and indifference. *Discrete Applied Mathematics*, 48:261–272, 1994.

[12] R.W. Irving. Matching medical students to pairs of hospitals: a new variation on a well-known theme. In *Proceedings of ESA '98: the Sixth European Symposium on Algorithms*, volume 1461 of *Lecture Notes in Computer Science*, pages 381–392. Springer, 1998.

[13] R.W. Irving and P. Leather. The complexity of counting stable marriages. *SIAM Journal on Computing*, 15(3):655–667, 1986.

[14] R.W. Irving and D.F. Manlove. An 8/5 approximation algorithm for a hard variant of stable marriage. In *Proceedings of COCOON 2007, 13th Annual International Computing and Combinatorics Conference, Banff Canada*, volume 4598 of *Lecture Notes in Computer Science*, pages 548–558. Springer, 2007.

[15] R.W. Irving, D.F. Manlove, and S. Scott. The Hospitals/Residents problem with Ties. In *Proceedings of SWAT 2000: the 7th Scandinavian Workshop on Algorithm Theory*, volume 1851 of *Lecture Notes in Computer Science*, pages 259–271. Springer, 2000.

[16] R.W. Irving, D.F. Manlove, and S. Scott. Strong stability in the Hospitals/Residents problem. In *Proceedings of STACS 2003: the 20th Annual Symposium on Theoretical Aspects of Computer Science*, volume 2607 of *Lecture Notes in Computer Science*, pages 439–450. Springer, 2003.

[17] K. Iwama, S. Miyazaki, and K. Okamoto. A $\left(2 - c\frac{\log n}{n}\right)$-approximation algorithm for the stable marriage problem. In *Proceedings of SWAT 2004: the 9th Scandinavian Workshop on Algorithm Theory*, volume 3111 of *Lecture Notes in Computer Science*, pages 349–361. Springer, 2004.

[18] K. Iwama, S. Miyazaki, and N. Yamauchi. A $\left(2 - c\frac{1}{\sqrt{n}}\right)$–approximation algorithm for the stable marriage problem. In *Proceedings of ISAAC 2005: the Sixteenth International Symposium on Algorithms and Computation*, volume 3827 of *Lecture Notes in Computer Science*, pages 902–914. Springer, 2005.

[19] K. Iwama, S. Miyazaki, and N. Yamauchi. A 1.875–approximation algorithm for the stable marriage problem. In *Proceedings of SODA 2007: the Eighteenth ACM/SIAM Symposium on Discrete Algorithms*, pages 288–297, 2007.

[20] D.F. Manlove, R.W. Irving, K. Iwama, S. Miyazaki, and Y. Morita. Hard variants of stable marriage. *Theoretical Computer Science*, 276(1-2):261–279, 2002.

[21] `http://www.nrmp.org/about_nrmp/how.html` (National Resident Matching Program website).

[22] 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.

[23] A.E. Roth and M.A.O. Sotomayor. *Two-sided matching: a study in game-theoretic modeling and analysis*, volume 18 of *Econometric Society Monographs.* Cambridge University Press, 1990.

[24] `http://www.nes.scot.nhs.uk/sfas/` (Scottish Foundation Allocation Scheme website).