Socially stable matchings in the Hospitals/Residents problem

Georgios Askalidis[1], Nicole Immorlica[1,2], Augustine Kwanashie[3],
David F. Manlove[3] and Emmanouil Pountourakis[1]

[1]Dept. of Electrical Engineering and Computer Science, Northwestern University
[2]Microsoft Research New England
[3]School of Computing Science, University of Glasgow

Abstract. In the Hospitals/Residents (HR) problem, agents are partitioned into hospitals and residents. Each agent wishes to be matched to an agent (or agents) in the other set and has a strict preference over these potential matches. A matching is stable if there are no blocking pairs, i.e., no pair of agents that prefer each other to their assigned matches. Such a situation is undesirable as it could lead to a deviation in which the blocking pair form a private arrangement outside the matching. This however assumes that the blocking pair have social ties or communication channels to facilitate the deviation. Relaxing the stability definition to take account of the potential lack of social ties between agents can yield larger stable matchings.

In this paper, we define the Hospitals/Residents problem under Social Stability (HRSS) which takes into account social ties between agents by introducing a *social network graph* to the HR problem. Edges in the social network graph correspond to resident-hospital pairs in the HR instance that know one another. Pairs that do not have corresponding edges in the social network graph can belong to a matching M but they can never block M. Relative to a relaxed stability definition for HRSS, called *social stability*, we show that socially stable matchings can have different sizes and the problem of finding a maximum socially stable matching is NP-hard, though approximable within 3/2. Furthermore we give polynomial time algorithms for special cases of the problem.

1 Introduction

Matching problems generally involve the assignment of a set (or sets) of agents to one another. Agents may be required to list other agents they find acceptable in order of preference, either explicitly or implicitly through a list of desirable characteristics. Agents may also be subject to capacity constraints, indicating the maximum number of assignments they are allowed to be involved in.

An example of such a matching problem that has received much attention in literature is the *Hospitals/Residents problem (HR)* [9,10,18,15]. An HR instance consists of a set of *residents* seeking to be matched to a set of *hospitals*. Each resident ranks a subset of the hospitals in strict order of preference, and vice versa. Further, each resident forms an *acceptable pair* with every hospital on his preference list. Finally, each hospital has a *capacity*, indicating the maximum number of residents that it can be assigned. A *matching* is a set assignments among acceptable pairs such that no resident is assigned to more than one hospital, and no hospital exceeds its capacity. An acceptable pair forms a *blocking pair* with respect to a matching, or *blocks* a matching, if both agents would rather be assigned to each other than remain with their assignees (if any) in the matching. A matching is *stable* if it admits no blocking pair. HR has a wide range of applications including traditional markets like the assignment of graduating medical students (residents) to hospitals [13,17] and students to high schools [1,2], and online markets like oDesk (an online labour market), AirBnB (an online short-term housing rental market), and Match.com/OkCupid/etc. (online dating markets). In applications such as these, it has been convincingly argued that stability is a desirable property of a matching [17].

Although the concept of stability is important in many applications of matching problems, there are classes of matching problems (such as the Stable Roommates problem) for which an instance is not guaranteed to admit a stable matching [9]. Moreover, enforcing the stability requirement tends to reduce the size of matchings discovered [6]. This is an issue particularly in the case of applications where it is desirable to find the largest possible matching. Also, it is generally assumed that a resident-hospital pair that blocks a matching in theory will also block the matching in practice. However this assumption is not always true in some real-life applications, as resident-hospital pairs are more likely to form blocking pairs in practice if social ties exist between them. These factors have motivated studies into alternative, weaker stability definitions that still aim to prevent a given matching from being subverted in practice while increasing the number of agents involved in the matchings.

Arcaute and Vassilvitskii [3] described the Hospitals/Residents problem in the context of assigning job applicants to company positions. They observed that applicants are more likely to be employed by a company if they are recommended by their friends who are already employees of that company. In their model, an applicant-company pair (a, c) may block a matching M if (a, c) blocks it in the traditional sense (as described in the analogous HR context) and *a is friends with* another applicant a' assigned to c in M. Thus their problem incorporates both the traditional HR problem and additionally an underlying social network, represented as an undirected graph consisting of applicants as nodes and edges between nodes where the corresponding applicants have some social ties (e.g., are friends). Matchings that admit no blocking pair in this context are called *locally stable* due to the addition of the informational constraint on blocking pairs. Cheng and McDermid [8] investigated the problem (which they called HR+SN) further and established various algorithmic properties and complexity results. They showed that locally stable matchings can be of different sizes and the problem of finding a maximum locally stable matching is NP-hard. They identified special cases where the problem is polynomially solvable and gave upper and lower bounds on the approximability of the problem.

While the HR+SN model is quite natural in the job market, it makes an assumption that the employed applicant a' will always be willing to make a recommendation. This however may not be the case as a recommendation may in practice lead to a' being rejected by his assigned company. Ultimately this may lead to a reassignment for a' to a worse company or indeed a' may end up unmatched. While it is true that a scenario may arise where these social ties between applicants may lead to a blocking pair of a matching, it is arguably equally likely that social ties between an applicant and the company itself will exist. That is, an applicant need not know another applicant who was employed by the company in order to block a matching; it is enough for him to know *any* employee in the company (for example the Head of Human Resources). Such a model could also be natural in many applications both within and beyond the job market context.

Additionally, many matching markets are cleared by a centralised clearinghouse. While more traditional markets require agents to explicitly list potential matches, many online markets ask agents to list desirable characteristics and then use software to infer the preference lists of the agents. In these markets, communication between agents is facilitated by the centralised clearinghouse. Some agent pairs in the market may have social ties outside the clearinghouse. Often these social ties are due to past interactions within the marketplace and so the clearinghouse is aware of them. These pairs can communicate outside the clearinghouse and might block proposed matchings. Most pairs, however, only become aware of each other when the clearinghouse proposes them as a match. Thus even if they prefer each other to their assigned matches, they will not be able to discover each other and deviate from the matching.

Based on these ideas, we present a variant of HR called the *Hospitals / Residents problem under Social Stability (HRSS)*. In this model, which we describe in the context of assigning graduating medical residents to hospital positions, we assume that a resident-hospital pair will only form a blocking pair in practice if there exists some social relationship between them. Two agents that have such a social relationship are called an *acquainted pair*, and this is represented by an edge in a *social network graph*. We call a pair of agents that do not have such a social relationship an *unacquainted pair*. Such a pair may be part of a matching M (given that M is typically constructed by a trusted third party, i.e., a centralised clearinghouse) but cannot form a blocking pair with respect to M. As a consequence, although a resident-hospital pair may form a blocking pair in the classical sense, if they are an unacquainted pair, they will not form a blocking pair in the HRSS context. A matching that admits no blocking pair in this new context is said to be *socially stable*. We denote the one-to-one restriction of HRSS as the *Stable Marriage problem with Incomplete lists under Social Stability (SMISS)*.

Hoefer and Wagner [11,12] studied a problem that generalises both HR+SN and HRSS. In their model, the social network graph involves all agents and need not be bipartite. A pair *locally blocks* a given matching M if (i) it blocks in the classical sense, and (ii) the agents involved are at most l edges apart in the social network graph augmented by M. This scenario can be viewed as a generalisation of the HR+SN ($l = 2$) and HRSS ($l = 1$) models. They studied the convergence time for better-response dynamics that converge to locally stable matchings, and also established a lower bound for the approximabiliy of the problem of finding a maximum locally stable matching (for the case that $l \leq 2$).

Locally stable matchings have also been investigated in the context of the Stable Roommates problem (a non-bipartite generalisation of the Stable Marriage problem) in [7]. Here, the *Stable Roommates problem with Free edges (SRF)* as introduced was motivated by the observation that, in kidney exchange matching schemes, donors and recipients do not always have full information about others and are more likely to have information only on others in the same transplant centre as them. The problem is defined by the traditional Stable Roommates problem together with a set of *free edges*. These correspond to pairs of agents in different transplant centres that do not share preference information; such pairs may be involved in stable matchings, but cannot block any matching. It is shown in [7] that the problem of determining whether a stable matching exists, given an SRF instance, is NP-complete.

In this paper, we present some algorithmic results for the HRSS model described above. In Section 2, we present some preliminary definitions and observations. In Section 3, we consider the approximability of MAX HRSS, the problem of finding a maximum socially stable matching in an HRSS instance. We give a 3/2-approximation algorithm for the problem, and also show that it is not approximable within $21/19 - \varepsilon$, for any $\varepsilon > 0$, unless P=NP, and not approximable within $3/2 - \varepsilon$, for any $\varepsilon > 0$, assuming the Unique Games Conjecture. In Section 4 we present polynomial-time algorithms for two special cases of MAX HRSS where (i) the number of unacquainted pairs is constant, and (ii) the number of acquainted pairs is constant. Finally some open problems are given in Section 5. All proofs for this paper are omitted for space reasons but can be found in [5].

2 Preliminary definitions and results

An instance I of the Hospitals/Residents problem (HR), as defined in [9], contains a set $R = \{r_1, r_2, ..., r_{n_1}\}$ of residents, a set $H = \{h_1, h_2, ..., h_{n_2}\}$ of hospitals. Each resident $r_i \in R$ ranks a subset of H in strict order of preference; each hospital $h_j \in H$ ranks a subset of R, consisting of those residents who ranked h_j, in strict order of preference. Each hospital h_j also has a capacity $c_j \in \mathbb{Z}^+$ indicating the maximum number of residents that can be assigned to it. A pair (r_i, h_j) is called an *acceptable pair* if h_j appears in r_i's preference list. We denote by $\mathcal{A}$ the set of all acceptable pairs. A *matching* M is a set of acceptable pairs such that each resident is assigned to at most one hospital and the number of residents assigned to each hospital does not exceed its capacity. If r_i is matched in M, we denote the hospital assigned to resident r_i in M by $M(r_i)$. We denote the set of residents assigned to hospital h_j in M as $M(h_j)$. A resident r_i is *unmatched* in M if no pair in M contains r_i. A hospital h_j is *undersubscribed* in M if $|M(h_j)| < c_j$. A pair (r_i, h_j) is said to *block* a matching M, or form a

men's preferences	women's preferences
m_1: $\underline{w_1}$	w_1: m_2 $\underline{m_1}$
m_2: w_1 $\underline{w_2}$	w_2: $\underline{m_2}$

Fig. 1. SMISS instance (I, G)

blocking pair with respect to M, in the classical sense, if (i) r_i is unmatched in M or prefers h_j to $M(r_i)$ and (ii) h_j is undersubscribed in M or prefers r_i to some resident in $M(h_j)$. A matching that admits no blocking pair is *stable*.

We define an instance (I, G) of the *Hospitals/Residents Problem under Social Stability (HRSS)* as consisting of an HR instance I (as defined above) and a bipartite graph $G = (R \cup H, A)$, where $A \subseteq \mathcal{A}$. A pair (r_i, h_j) belongs to A if and only if r_i has social ties with h_j. We call (r_i, h_j) an *acquainted pair*. We also define the set of *unacquainted* pairs (which cannot block any matching) to be $U = \mathcal{A} \backslash A$. A pair (r_i, h_j) *socially blocks* a matching M, or forms a *social blocking pair* with respect to M, if (r_i, h_j) blocks M in the classical sense in the underlying HR instance I and $(r_i, h_j) \in A$. A matching M is said to be *socially stable* if there exists no social blocking pair with respect to M. If we restrict the hospitals' capacities to 1, we obtain the *Stable Marriage problem with Incomplete lists under Social Stability (SMISS)*.

Clearly every instance of HRSS admits a socially stable matching. This is because the underlying HR instance is bound to admit a stable matching [9] which is also socially stable. However socially stable matchings could be larger than stable matchings. Consider the SMISS instance (I, G) shown in Figure 1, where the acquainted pairs in the social network graph are underlined in the preference lists. Matchings $M_1 = \{(m_1, w_1), (m_2, w_2)\}$ and $M_2 = \{(m_2, w_1)\}$ are both socially stable in (I, G) and M_2 is the unique stable matching. Thus an instance of SMISS (and hence HRSS) can admit a socially stable matching that is twice the size of a stable matching. Clearly the instance shown in Figure 1 can be replicated to give an arbitrarily large SMISS instance with a socially stable matching that is twice the size of a stable matching. This, and applications where we seek to match as many agents as possible, motivates MAX HRSS.

There is also a strong relationship between HRSS and the HR+SN problem described in [8]. We have shown (following an idea of Király) in [5] that an instance (I, G) of HRSS can be transformed in polynomial time to an instance (I', G') of HR+SN such that a socially stable matching M in (I, G) is locally stable in (I', G') and a complete locally stable matching (one in which all the residents are matched) in (I', G') is a complete socially stable matching in (I, G).

3 Approximating MAX HRSS

We begin this section by noting that MAX HRSS is NP-hard even in a very restricted setting. Let MAX SMISS denote the restriction of MAX HRSS in which all hospitals have capacity 1.

Theorem 1 *MAX SMISS is NP-hard even if each list is of length at most 3.*

In order to deal with this hardness, polynomial-time approximation algorithms can be developed for MAX HRSS. In this section we present a 3/2-approximation algorithm for MAX HRSS. We show this is tight assuming the Unique Games Conjecture (UGC), and also show a $21/19 - \varepsilon$ lower bound assuming $P \neq NP$. The lower bounds hold even for MAX SMISS. We start by giving the inapproximability result assuming P≠NP.

Theorem 2 *MAX SMISS is not approximable within* $21/19 - \varepsilon$*, for any* $\varepsilon > 0$*, unless P=NP.*

We can obtain a better lower bound of $3/2 - \varepsilon$, for any $\varepsilon > 0$, if we strengthen our assumption from $P \neq NP$ to the truth of the UGC.

Theorem 3 *Assuming the UGC, MAX SMISS cannot be approximated within* $3/2 - \varepsilon$*, for any* $\varepsilon > 0$.

For the upper bound for MAX HRSS, we observe that a technique known as *cloning* has been described in literature [10,18], which may be used to convert an HR instance I into an instance I' of the Stable Marriage problem with Incomplete lists in polynomial time, such that there is a one-to-one correspondence between the set of stable matchings in I and I'. A similar technique can be used to convert an HRSS instance to an SMISS instance in polynomial time.

Theorem 4 *Given an instance* (I, G) *of HRSS, we may construct in* $O(n_1 + c_{max}m)$ *time an instance* (I', G') *of SMISS such that a socially stable matching* M *in* (I, G) *can be transformed in* $O(c_{max}m)$ *time to a socially stable matching* M' *in* (I', G') *with* $|M'| = |M|$ *and conversely, where* n_1 *is the number of residents,* c_{max} *is the maximum hospital capacity and* m *is the number of acceptable resident-hospital pairs in* I*.*

Due to Theorem 4, an approximation algorithm α for MAX SMISS with performance guarantee c (for some constant $c > 0$) can be used to obtain an approximation for MAX HRSS with the same performance guarantee. This can be done by cloning the HRSS instance (I, G) to form an SMISS instance (I', G'), and applying α to (I', G') to obtain a matching M'. This matching can then be transformed to a matching M in (I, G) such that $|M| = |M'|$. Our first upper bound for MAX HRSS is an immediate consequence of the fact that any stable matching is at least half the size of a maximum socially stable matching.

Proposition 5 *MAX HRSS is approximable within a factor of 2.*

We now present a 3/2-approximation algorithm for MAX SMISS. The algorithm relies on the principles outlined in the 3/2-approximation algorithms for the general case of MAX HRT, the problem of finding a maximum cardinality stable matching given an instance of the Hospitals / Residents problem with Ties, as presented by Király [14] and McDermid [16]. Given an instance (I, G) of SMISS, the algorithm works by running a modified version of the extended Gale-Shapley algorithm [9] where unmatched men are given a chance to propose again by promoting them on all the preference lists on which they appear.

Let A and U denote the sets of acquainted and unacquainted pairs in (I, G) respectively.

Consider a woman w_j in (I, G). We denote an *unacquainted man* m_i *on* w_j*'s preference list* as one where $(m_i, w_j) \in U$. Similarly we denote an *acquainted man* m_i *on* w_j*'s preference list* as one where $(m_i, w_j) \in A$. For a man m_i, we denote $next(m_i)$ as the next woman on m_i's list succeeding the last woman to whom he proposed to or the first woman on m_i's list if he has been newly promoted or is proposing for the first time. During the execution of the algorithm if a man runs out of women to propose to on his list for the first time, he is *promoted*, thus allowing him to propose to the remaining women on his list beginning from the first. A man can only be promoted once during the execution of the algorithm. If a promoted man still remains unmatched after proposing to all the women on his preference list, he is removed from the instance and will not be part of the final matching.

In the classical Gale-Shapley algorithm [9], a woman w_j prefers a man m_i to another m_k if $rank(w_j, m_i) < rank(w_j, m_k)$. We define a modified version of the extended Gale-Shapley algorithm [10], *mod-EXGS*, where a woman does not accept or reject proposals from men solely on the basis of their positions on her preference list, but also on the basis of their status as to whether they are acquainted or unacquainted men on her list and whether they have been promoted. Given two men m_i and m_k on a woman w_j's preference list, we define the relations $\triangleleft_{w_j}$, $\triangleleft'_{w_j}$ and $\prec_{w_j}$ as follows:

Definition 6 *Let m_i and m_k be any two men on a woman w_j's list. Then*
1. $m_i \triangleleft_{w_j} m_k$ if either
(i) $(m_i, w_j) \in U$, $(m_k, w_j) \in U$, m_i is promoted and m_k is unpromoted or
(ii) $(m_i, w_j) \in A$, $(m_k, w_j) \in U$ and m_k is unpromoted.
2. $m_i \triangleleft'_{w_j} m_k$ if $m_i \ntriangleleft_{w_j} m_k$, $m_k \ntriangleleft_{w_j} m_i$ and w_j prefers m_i to m_k in the classical sense.
We define $\prec_{w_j} = \triangleleft_{w_j} \cup \triangleleft'_{w_j}$.

The relation $\prec_{w_j}$ will be used to determine whether a proposal from a man is accepted or rejected by w_j.

The main algorithm *approx-SMISS* (as shown in Algorithm 1) starts by calling *mod-EXGS* (as shown in Algorithm 2) where a *proposal sequence* is started by allowing each man to propose to women beginning from the first woman on his preference list. If a man m_i proposes to a woman w_j on his list and w_j is matched and $m_i \prec_{w_j} M(w_j)$, then w_j is unmatched from her partner m_k, and m_k will be allowed to continue proposing to other women on his list. w_j is then assigned to m_i. On the other hand, if $M(w_j) \prec_{w_j} m_i$ then w_j rejects m_i's proposal. Also if w_j is unmatched when m_i proposes, she is assigned to m_i. Irrespective of whether the proposal from m_i is accepted or rejected, if $(m_i, w_j) \in A$ then all pairs (m_k, w_j) such that $rank(w_j, m_k) > rank(w_j, m_i)$ are deleted from the instance. However if $(m_i, w_j) \in U$ no such deletions take place. This proposal sequence continues until every man is either matched or has exhausted his preference list.

Algorithm 1 approx-SMISS

```
initial matching $M = \emptyset$;
**while** some unmatched man with a non-empty preference list exists **do**
    call mod-EXGS;
    **for** all $m_i$ such that $m_i$ is unmatched and promoted **do**
        remove $m_i$ from instance;
    **end for**
    **for** all $m_i$ such that $m_i$ is unmatched, unpromoted and has a non-empty preference list **do**
        promote $m_i$;
    **end for**
**end while**
return the resulting matching $M$;
```

Algorithm 2 mod-EXGS

```
**while** some man $m_i$ is unmatched and still has a woman left on his list **do**
    $w_j = next(m_i)$;
    **if** $w_j$ is matched in $M$ and $m_i \prec_{w_j} M(w_j)$ **then**
        $M = M \setminus \{(M(w_j), w_j)\}$;
    **end if**
    **if** $w_j$ is unmatched in $M$ **then**
        $M = M \cup \{(m_i, w_j)\}$;
    **end if**
    **if** $(m_i, w_j) \in A$ **then**
        **for** each $m_k$ such that $(m_k, w_j) \in \mathcal{A}$ and $rank(w_j, m_k) > rank(w_j, m_i)$ **do**
            delete $(m_k, w_j)$ from instance;
        **end for**
    **end if**
**end while**
```

After each proposal sequence (where control is returned to the *approx-SMISS* algorithm), if a promoted man still remains unmatched after proposing to all the women on his preference list, he is removed from the instance. Also if a previously unpromoted man exhausts his preference lists and is still unmatched, he is promoted and a new proposal sequence initiated (by calling *mod-EXGS*). The algorithm terminates when each man either (i) is assigned a partner, (ii) has no woman on his preference list or (iii) has been promoted and has proposed to all the women on his preference list for a second time.

Lemma 7 *If algorithm* approx-SMISS *is executed on an SMISS instance* (I, G)*, it terminates with a socially stable matching* M *in* (I, G)*.*

The execution of the *mod-EXGS* algorithm takes $O(m)$ time where $m = |\mathcal{A}|$ is the number of acceptable pairs. These executions can be performed at most $2n_1$ times, where n_1 is the number of men, as a man is given at most two chances to propose to the women on his list. Thus the overall time complexity of the algorithm is $O(n_1 m)$. The above results, together with Theorem 4, lead

men's preferences	women's preferences
m_1: $\underline{w_1}$ $\underline{w_3}$	w_1: m_2 $\underline{m_1}$
m_2: w_1 $\underline{w_2}$	w_2: $\underline{m_2}$ m_3
m_3: w_2	w_3: $\underline{m_1}$

Fig. 2. $|M_{opt}| = (3/2).|M|$

us to state the following theorem concerning the performance guarantee of the approximation algorithm for MAX HRSS.

Theorem 8 *MAX HRSS is approximable within a factor of 3/2.*

The SMISS instance shown in Figure 2 (where the acquainted pairs in the social network graph are underlined in the preference lists) shows that the 3/2 bound for the algorithm is tight. Here $M_{opt} = \{(m_1, w_3), (m_2, w_1), (m_3, w_2)\}$ is the unique maximum socially stable matching. Also the approximation algorithm outputs $M = \{(m_1, w_1), (m_2, w_2)\}$ irrespective of the order in which proposals are made. Clearly this instance can be replicated to obtain an arbitrarily large SMISS instance for which the performance guarantee is tight.

We remark that a similar 3/2-approximation algorithm for MAX HRSS was presented independently by Askalidis et al. in [4].

4 Some special cases of HRSS

Given the hardness results obtained for the problem of finding a maximum socially stable matching in a general HRSS instance, the need arises to investigate special cases of the problem that are tractable. This section describes some polynomial-time solvability results for two special cases of HRSS.

Before presenting the two main results of this section, we first note that we have given an $O(n^{3/2} \log n)$ algorithm for finding a maximum socially stable matching, given an instance of MAX SMISS where each man is allowed to have at most two women on his preference list and n is the total number of men and women involved. This algorithm, presented in [5], is omitted for space reasons.

4.1 HRSS with a constant number of unacquainted pairs

It is easy to see that in the special case where the set U of unacquainted pairs is exactly the set $\mathcal{A}$ of acceptable pairs in the underlying HR instance, then the set A of acquainted pairs satisfies $A = \emptyset$ and every matching found is a socially stable matching. Also if the instance contains no unacquainted pairs (i.e., $A = \mathcal{A}$ and $U = \emptyset$), then only stable matchings in the classical sense are socially stable. In both these cases, a maximum socially stable matching can be generated in polynomial time. The case may however arise where the number of unacquainted pairs is constant. In this case, we show that it is also possible to generate a maximum socially stable matching in polynomial time.

Let (I, G) be an instance of HRSS and let $S \subseteq \mathcal{A}$ be a subset of the acceptable pairs in I. We denote $I \backslash S$ as the instance of HR obtained from I by deleting the pairs in S from the preference lists in I. The following proposition plays a key role in establishing the correctness of the algorithm.

Proposition 9 *Let (I, G) be an instance of HRSS. Let M be a socially stable matching in (I, G). Then there exists a set of unacquainted pairs $U' \subseteq U$ such that M is stable in $I' = I \backslash U'$. Conversely suppose that M is a stable matching in $I' = I \backslash U'$ for some $U' \subseteq U$. Then M is socially stable in (I, G).*

By considering all subsets $U' \subseteq U$, forming I', finding a stable matching in each such I' and keeping a record of the maximum stable matching found, we obtain a maximum socially stable matching in (I, G). This discussion leads to the following theorem.

Theorem 10 *Given an instance (I, G) of HRSS where the set U of unacquainted pairs is of constant size, a maximum socially stable matching can be generated in $O(m)$ time, where $m = |\mathcal{A}|$ is the number of acceptable pairs.*

4.2 HRSS with a constant number of acquainted pairs

We now consider the restriction of HRSS in which the set A of acquainted pairs is of constant size k. Given an instance (I, G) of this problem we show that a maximum socially stable matching can be found in polynomial time. Let $A = \{e_1, e_2, ..., e_k\}$ where e_i represents an acquainted pair (r_{s_i}, h_{t_i}) $(1 \leq i \leq k)$. A tree T of depth k is constructed with all nodes at depth i labelled e_{i+1} $(i \geq 0)$. There are left and right branches below e_i. Each branch corresponds to a condition placed on r_{s_i} or h_{t_i} with respect to a matching M. The left branch below e_i (i.e., a resident condition branch) corresponds to the condition that r_{s_i} is matched in M and prefers his partner to h_{t_i}. The right branch below e_i (i.e., a hospital condition branch) corresponds to the condition that h_{t_i} is fully subscribed in M and has a partner no worse than r_{s_i}. Satisfying at least one of these conditions ensures that M admits no blocking pair involving (r_{s_i}, h_{t_i}). The tree is constructed in this manner with the nodes at depth $k-1$, labelled e_k, branching in the same way to dummy leaf nodes e_{k+1} (not representing acquainted pairs).

A path P from the root node e_1 to a leaf node e_{k+1} will visit all pairs in A exactly once. Every left branch in P gives a resident condition and every right branch gives a hospital condition. Let R' and H' be the set of residents and hospitals involved in resident and hospital conditions in P respectively. Given a matching M, enforcing all the conditions along P can be achieved by first deleting all pairs from the instance I that could potentially violate these conditions. So if some resident condition along P states that a resident r_{s_i} must be matched in M to a hospital he prefers to h_{t_i} then r_{s_i}'s preference list is truncated starting with h_{t_i}. If some hospital condition states that a hospital h_{t_i} must be fully subscribed in M and must not be matched to a resident worse than r_{s_i} then h_{t_i}'s preference list is truncated starting from the resident immediately following r_{s_i}. After performing these truncations based on the conditions along P, a new HR instance I' is obtained.

Proposition 11 *If M is a matching in I' that is computed at the leaf node of a path P and all residents in R' are matched in M and all hospitals in H' are fully subscribed in M then M is a socially stable matching in (I, G).*

By finding a maximum weight matching in a suitable weighted graph (see [5] for further details), we may in polynomial time find a largest matching M satisfying the constraints of Proposition 11 or report that no such matching exists. In the latter case P is ruled as infeasible and another path is considered, otherwise P is called feasible.

There are 2^k paths from the root node to leaf nodes in the tree T. The following proposition is important to our result.

Proposition 12 *There must exist at least one feasible path in T.*

To generate a maximum socially stable matching M in an instance (I, G) of HRSS, all 2^k paths through T from the root node to leaf nodes are considered with a record kept of the largest matching M (satisfying the constraints of Proposition 11) computed at the leaf node of each feasible path. M is then the desired matching as the following proposition shows

Proposition 13 *If M is a matching obtained from the process described above, M is a maximum socially stable matching in (I, G).*

The above proposition leads to the following main result of this subsection.

Theorem 14 *Given an instance (I, G) of HRSS where the set A of acquainted pairs satisfies $|A| = k$ for some constant k, a maximum socially stable matching can be generated in $O(c_{max} m \sqrt{n_1 + C})$ time where n_1 is the number of residents, m is the number of acceptable pairs, c_{max} is the largest capacity of any hospital and C is the total capacity of all the hospitals in the problem instance.*

Following the results in Theorems 10 and 14, we conclude this section with the theorem below showing the existence of FPT algorithms for MAX HRSS under two different parameterisations.

Theorem 15 *MAX HRSS is in FPT with parameter k, where either $k = |A|$ or $k = |U|$, and A and U are the sets of acquainted and unacquainted pairs respectively.*

5 Open problems

The study of the Hospitals / Residents problem under Social Stability is still at an early stage, and some interesting open problems remain. Firstly it is worth considering the scenario where ties exist in the preference lists of agents. Also it could be argued that information about undersubscribed hospitals would be in the public domain, and hence an undersubscribed hospital may form an acquainted pair with all the residents on its preference list. It would be interesting to investigate algorithmic aspects of this variant of HRSS.

Acknowledgments

The fourth author is supported by grant EP/K010042/1 from the Engineering and Physical Sciences Research Council. We would like to thank Martin Hoefer for making the third and fourth authors aware of [4]; Zoltán Király for observing Theorem 15 and for other valuable comments; and Rob Irving and an anonymous referee for further valuable suggestions concerning this paper.

References

1. A. Abdulkadiroğlu, P.A. Pathak, and A.E. Roth. The Boston public school match. *American Economic Review*, 95(2):368–371, 2005.
2. A. Abdulkadiroğlu, P.A. Pathak, and A.E. Roth. The New York City high school match. *American Economic Review*, 95(2):364–367, 2005.
3. E. Arcaute and S. Vassilvitskii. Social networks and stable matchings in the job market. In *Proc. WINE '09*, *LNCS* vol. 5929, pp. 220-231, 2009.
4. G. Askalidis, N. Immorlica, and E. Pountourakis. Socially stable matchings. CoRR Technical Report 1302.3309. Available from `http://arxiv.org/abs/1302.3309`.
5. G. Askalidis, N. Immorlica, A. Kwanashie, D.F. Manlove and E. Pountourakis. Socially stable matchings in the Hospitals / Residents problem. CoRR Technical Report 1303.2041. Available from `http://arxiv.org/abs/1303.2041`.
6. P. Biró, D.F. Manlove, and S. Mittal. Size versus stability in the marriage problem. *Theoretical Computer Science*, 411:1828–1841, 2010.
7. K. Cechlárová and T. Fleiner. Stable roommates with free edges. Technical Report 2009-01, Egerváry Research Group on Combinatorial Optimization, Budapest, 2009.
8. C. Cheng and E. McDermid. Maximum locally stable matchings. In *Proc. Match-UP '12*, pp. 51–62, 2012.
9. D. Gale and L.S. Shapley. College admissions and the stability of marriage. *American Mathematical Monthly*, 69:9–15, 1962.
10. D. Gusfield and R.W. Irving. *The Stable Marriage Problem: Structure and Algorithms.* MIT Press, 1989.
11. M. Hoefer. Local matching dynamics in social networks. *Information and Computation*, 222:20–35, 2013.
12. M. Hoefer and L. Wagner. Locally stable marriage with strict preferences. In *Proc. ICALP 2013*, *LNCS*, to appear, 2013.
13. R.W. Irving. Matching medical students to pairs of hospitals: a new variation on a well-known theme. In *Proc. ESA '98*, *LNCS* vol. 1461, pp. 381–392, 1998.
14. Z. Király. Linear time local approximation algorithm for maximum stable marriage. In *Proc. MATCH-UP '12*, pp. 99-110, 2012.
15. D.F. Manlove. *Algorithmics of Matching Under Preferences.* World Scientific, 2013.
16. E. McDermid. A 3/2 approximation algorithm for general stable marriage. In *Proc. ICALP '09*, *LNCS* vol. 5555, pp. 689–700, 2009.
17. 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.
18. A.E. Roth and M.A.O. Sotomayor. *Two-sided matching: a study in game-theoretic modeling and analysis*, Cambridge University Press, 1990.