Biró, P., Manlove, D. F., and McBride, I. (2014) *The Hospitals/Residents Problem with Couples: complexity and integer programming models.* Lecture Notes in Computer Science, 8504 . pp. 10-21. ISSN 0302-9743

The Hospitals / Residents Problem with Couples: Complexity and Integer Programming Models[⋆]

Péter Biró[1,⋆⋆], David F. Manlove[2,⋆⋆⋆], and Iain McBride[2,†,‡]

[1] Institute of Economics, Research Centre for Economic and Regional Studies,
Hungarian Academy of Sciences, H-1112, Budaorsi ut 45, Budapest, Hungary
[2] School of Computing Science, Sir Alwyn Williams Building, University of Glasgow,
Glasgow G12 8QQ, UK

Abstract. The Hospitals / Residents problem with Couples (HRC) is a generalisation of the classical Hospitals / Residents problem (HR) that is important in practical applications because it models the case where couples submit joint preference lists over pairs of (typically geographically close) hospitals. In this paper we give a new NP-completeness result for the problem of deciding whether a stable matching exists, in highly restricted instances of HRC, and also an inapproximability bound for finding a matching with the minimum number of blocking pairs in equally restricted instances of HRC. Further, we present a full description of the first Integer Programming model for finding a maximum cardinality stable matching in an instance of HRC and we describe empirical results when this model applied to randomly generated instances of HRC.

1 Introduction

The Hospitals / Residents Problem. The *Hospitals / Residents problem* (HR) is a many-to-one allocation problem. An instance of HR consists of two groups of agents – one containing *hospitals* and one containing *residents*. Every hospital expresses a linear preference over some subset of the residents, its *preference list*. The residents in a hospital's preference list are its *acceptable* partners. Further, every hospital has a *capacity*, c_j, the maximum number of posts it has available to match with residents. Every resident expresses a linear preference over some subset of the hospitals, his *acceptable* hospitals.

The preferences expressed in this fashion are reciprocal: if a resident r_i is acceptable to a hospital h_j, then h_j is also acceptable to r_i, and vice versa. A many-to-one *matching* between residents and hospitals is sought, which is a

---

[⋆] A preliminary version of this paper appeared in the Proceedings of OR 2013.

[⋆⋆] Supported by the Hungarian Academy of Sciences under its Momentum Programme (LD-004/2010) and also by OTKA grant no. K108673.

[⋆⋆⋆] Supported by Engineering and Physical Sciences Research Council grant EP/K010042/1.

[†] Supported by a SICSA Prize PhD Studentship.

[‡] Email `i.mcbride.1@research.gla.ac.uk`.

set of acceptable resident-hospital pairs such that each resident appears in at most one pair and each hospital h_j at most c_j pairs. If a resident r_i appears in some pair of M, r_i is said to be *assigned* in M and *unassigned* otherwise. Any hospital assigned fewer residents than its capacity in some matching M is *under-subscribed* in M.

A matching is *stable* if it admits no *blocking pair*. Following the definition in [10], a blocking pair consists of a mutually acceptable resident-hospital pair (r, h) such that both of the following hold: (i) either r is unassigned, or r prefers h to his assigned hospital; (ii) either h is under-subscribed in the matching, or h prefers r to at least one of its assigned residents. Were such a pair to exist, they could form a pairing outside of the matching, undermining its integrity [20].

It is known that every instance of HR admits at least one stable matching and such a matching may be found in time linear in the size of the instance [10]. Also, for an arbitrary HR instance I, any resident that is assigned in one stable matching in I is assigned in all stable matchings in I, moreover any hospital that is under-subscribed in some stable matching in I is assigned exactly the same set of residents in every stable matching in I [11, 20, 21].

HR can be viewed as an abstract model of the matching process involved in a centralised matching scheme such as the National Resident Matching Program (NRMP) [18] through which graduating medical students are assigned to hospital posts in the USA. A similar process was used until recently to match medical graduates to Foundation Programme places in Scotland, called the Scottish Foundation Allocation Scheme (SFAS) [13]. Analogous allocation schemes having a similar underlying problem model exist around the world, both in the medical sphere, e.g. in Canada [9], Japan [14], and beyond, e.g. in higher education allocation in Hungary [5].

The Hospitals / Residents Problem with Couples. Centralised matching schemes such as the NRMP and the SFAS have had to evolve to accommodate couples who wish to be allocated to (geographically) compatible hospitals. The capability to take account of the joint preferences of couples has been in place in the NRMP context since 1983 and since 2009 in the case of SFAS. In schemes where the agents may be involved in couples, the underlying allocation problem can modelled by the so-called *Hospitals / Residents problem with Couples* (HRC).

As in the case of HR, an instance of HRC consists of a set of *hospitals* H and a set of *residents* R. The residents in R are partitioned into two sets, S and S'. The set S consists of *single* residents and the set S' consists of those residents involved in *couples*. There is a set $C = \{(r_i, r_j) : r_i, r_j \in S'\}$ of *couples* such that each resident in S' belongs to exactly one pair in C.

Each single resident $r_i \in S$ expresses a linear preference order over his acceptable hospitals. Each pair of residents $(r_i, r_j) \in C$ expresses a joint linear preference order over a subset A of $H \times H$ where $(h_p, h_q) \in A$ represents the joint assignment of r_i to h_p and r_j to h_q. The hospital pairs in A represent those joint assignments that are *acceptable* to (r_i, r_j), all other joint assignments being *unacceptable* to (r_i, r_j).

Each hospital $h_j \in H$ expresses a linear preference order over those residents who find h_j acceptable, either as a single resident or as part of a couple. As in the HR case, each hospital $h_j \in H$ has a *capacity*, c_j.

A many-to-one *matching* between residents and hospitals is sought, which is defined as for HR with the additional restriction that each couple (r_i, r_j) is either jointly unassigned, meaning that both r_i and r_j are unassigned, or jointly assigned to some pair (h_k, h_l) that (r_i, r_j) find acceptable.As in HR, we seek a *stable* matching, which guarantees that no resident and hospital, and no couple and pair of hospitals, have an incentive to deviate from their assignments and become assigned to each other.

Roth [20] considered stability in the HRC context although did not define the concept explicitly. Whilst Gusfield and Irving [12] defined stability in HRC, their definition neglected to deal with the case that both members of a couple may wish to be assigned to the same hospital. Manlove and McDermid [16] extended their definition to deal with this possibility (however both definitions are equivalent in the case that no pair of the form (h_p, h_p) appears in any couple's preference list). We adopt Manlove and McDermid's stability definition in this paper, and now define it formally as follows.

Definition 1 ([16]) *A matching M is* stable *if none of the following holds:*

1. *The matching is blocked by a hospital h_j and a single resident r_i, as in the classical HR problem.*
2. *The matching is blocked by a couple (r_i, r_j) and a hospital h_k such that* either
   - *(a) (r_i, r_j) prefers $(h_k, M(r_j))$ to $(M(r_i), M(r_j))$, and h_k is either under-subscribed in M or prefers r_i to some member of $M(h_k)\backslash\{r_j\}$* or
   - *(b) (r_i, r_j) prefers $(M(r_i), h_k)$ to $(M(r_i), M(r_j))$, and h_k is either under-subscribed in M or prefers r_j to some member of $M(h_k)\backslash\{r_i\}$*
3. *The matching is blocked by a couple (r_i, r_j) and (not necessarily distinct) hospitals $h_k \neq M(r_i)$, $h_l \neq M(r_j)$; that is, (r_i, r_j) prefers the joint assignment (h_k, h_l) to $(M(r_i), M(r_j))$, and* either
   - *(a) $h_k \neq h_l$, and h_k (respectively h_l) is either under-subscribed in M or prefers r_i (respectively r_j) to at least one of its assigned residents in M;* or
   - *(b) $h_k = h_l$, and h_k has at least two free posts in M, i.e., $c_k - |M(h_k)| \geq 2$;* or
   - *(c) $h_k = h_l$, and h_k has one free post in M, i.e., $c_k - |M(h_k)| = 1$, and h_k prefers at least one of r_i, r_j to some member of $M(h_k)$;* or
   - *(d) $h_k = h_l$, h_k is full in M, h_k prefers r_i to some $r_s \in M(h_k)$, and h_k prefers r_j to some $r_t \in M(h_k)\backslash\{r_s\}$.*

Existing Algorithmic Results for HRC. In contrast with HR, an instance of HRC need not admit a stable matching [20]. Also an instance of HRC may admit stable matchings of differing sizes [2]. Further, the problem of deciding whether a stable matching exists in an instance of HRC is NP-complete, even in the restricted case where there are no single residents and all of the hospitals have only one available post [17, 19].

In many practical applications of HRC the residents' preference lists are short. Let (α, β)-HRC denote the restriction of HRC in which each single resident's preference list contains at most α hospitals, each couple's preference list contains at most α pairs of hospitals and each hospital's preference list contains at most β residents. (α, β)-HRC is hard even for small values of α and β: Manlove and McDermid [16] showed that $(3, 6)$-HRC is NP-complete.

Since the existence of an efficient algorithm for finding a stable matching, or reporting that none exists, in an instance of HRC is unlikely, in practical applications such as SFAS and NRMP, stable matchings are found by applying heuristics [3,6,22]. However, neither the SFAS heuristic, nor the NRMP heuristic guarantee to terminate and output a stable matching, even in instances where a stable matching does exist. Hence, a method which guarantees to find a maximum cardinality stable matching in an arbitrary instance of HRC, where one exists, might be of considerable interest. For further results on HRC the reader is referred to [7] and [15].

Contribution of this Work. In this paper, we present in Section 2 a new NP-completeness result for the problem of deciding whether there exists a stable matching in an instance of $(2, 2)$-HRC where there are no single residents and all hospitals have capacity 1. This is the most restricted case of HRC currently known for which NP-completeness holds. A natural way to try to cope with this complexity is to approximate a matching that is 'as stable as possible', i.e., admits the minimum number of blocking pairs [1]. Let MIN-BP-HRC denote the problem of finding a matching with the minimum number of blocking pairs, given an instance of HRC, and let (α, β)-MIN-BP-HRC denote the restriction to instances of (α, β)-HRC. We prove that $(2, 2)$-MIN-BP-HRC is not approximable within $n_1^{1-\varepsilon}$, where n_1 is the number of residents in a given instance, for any $\varepsilon > 0$, unless $P = NP$. Further in Section 3 we present a description of the first Integer Programming (IP) model for finding a maximum cardinality stable matching or reporting that none exists in an arbitrary instance of HRC. Then in Section 4 we present elements of an empirical study of this model as applied to randomly generated instances.

2 Complexity Results

In this section we present hardness results for finding and approximating stable matchings in instances of HRC. For space reasons all of the proofs are omitted but appear in full in [8], a technical report by the same authors. We begin by establishing NP-completness for the problem of deciding whether a stable matching exists in a highly restricted instance of HRC. Our proof involves a reduction from (2,2)-E3-SAT, the problem of deciding, given a Boolean formula B in CNF over a set of variables V, whether B is satisfiable, where B has the following properties: (i) each clause contains exactly 3 literals and (ii) for each $v_i \in V$, each of literals v_i and $\bar{v}_i$ appears exactly twice in B. Berman et al. [4] have shown that (2,2)-E3-SAT is NP-complete.

Theorem 1. *Given an instance of* $(2, 2)$-HRC*, the problem of deciding whether there exists a stable matching is NP-complete. The result holds even if there are no single residents and each hospital has capacity 1.*

We now turn to MIN-BP-HRC. Clearly Theorem 1 implies that this problem is NP-hard. By chaining together instances of $(2, 2)$-HRC constructed in the proof of Theorem 1, we arrive at a gap-introducing reduction which establishes a strong inapproximability result for MIN-BP-HRC under the same restrictions as in Theorem 1.

Theorem 2. $(2, 2)$-MIN-BP-HRC *is not approximable within* $n_1^{1-\varepsilon}$*, where* n_1 *is the number of residents in a given instance, for any* $\varepsilon > 0$*, unless* $P = NP$*, even if there are no single residents and each hospital has capacity 1.*

3 An IP Formulation for HRC

In this section we describe an IP model which finds a maximum cardinality stable matching in an arbitrary instance of HRC, or reports that no stable matching exists. The variables and constraints required to construct the model are shown below; a detailed proof of the correctness of the model is omitted due to space restrictions, but is presented in full in [8].

Let I be an instance of HRC with residents $R = \{r_1, r_2, \ldots, r_{n_1}\}$ and hospitals $H = \{h_1, h_2, \ldots, h_{n_2}\}$. Without loss of generality, suppose residents $r_1, r_2 \ldots r_{2c}$ are in couples. Again, without loss of generality, suppose that the couples are (r_{2i-1}, r_{2i}) $(1 \leq i \leq c)$. Suppose that the joint preference list of a couple $c_i = (r_{2i-1}, r_{2i})$ is:

$$c_i \ : \ (h_{\alpha_1}, h_{\beta_1}), (h_{\alpha_2}, h_{\beta_2}) \ldots (h_{\alpha_l}, h_{\beta_l}).$$

From this list we create the following projected preference lists for r_{2i-1} and r_{2i}:

$$r_{2i-1} \ : \ h_{\alpha_1}, h_{\alpha_2} \ldots h_{\alpha_l} \qquad\qquad r_{2i} \ : \ h_{\beta_1}, h_{\beta_2} \ldots h_{\beta_l}.$$

Let $l(c_i)$ denote the length of the preference list of c_i, and let $l(r_{2i-1})$ and $l(r_{2i})$ denote the lengths of the projected preference lists of r_{2i-1} and r_{2i} respectively. Then $l(r_{2i-1}) = l(r_{2i}) = l(c_i)$. A given hospital h_j may appear more than once in the projected preference list of a resident in a couple $c_i = (r_{2i-1}, r_{2i})$.

Let the single residents be $r_{2c+1}, r_{2c+2} \ldots r_{n_1}$, where each single resident r_i, has a preference list of length $l(r_i)$ consisting of individual hospitals $h_j \in H$. Each hospital $h_j \in H$ has a preference list of individual residents $r_i \in R$ of length $l(h_j)$. Further, each hospital $h_j \in H$ has capacity $c_j \geq 1$, the number of residents with which it may match.

Let J be the following IP formulation of I. In J, for each i $(1 \leq i \leq n_1)$ and p $(1 \leq p \leq l(r_i))$, define a variable $x_{i,p}$ such that

$$x_{i,p} = \begin{cases} 1 \text{ if } r_i \text{ is assigned to their } p^{th} \text{ choice hospital} \\ 0 \text{ otherwise.} \end{cases}$$

For $p = l(r_i)+1$ define a variable $x_{i,p}$ whose intuitive meaning is that resident r_i is unassigned. Therefore we also have

$$x_{i,l(r_i)+1} = \begin{cases} 1 \text{ if } r_i \text{ is unassigned} \\ 0 \text{ otherwise.} \end{cases}$$

Let $X = \{x_{i,p} : 1 \leq i \leq n_1 \wedge 1 \leq p \leq l(r_i) + 1\}$. As part of the model, for all $x_{i,p} \in X$, we enforce $x_{i,p} \in \{0, 1\}$. Let $pref(r_i, p)$ denote the hospital at position p of a single resident r_i's preference list or on the projected preference list of coupled resident where $1 \leq i \leq n_1$ and $1 \leq p \leq l(r_i)$. Let $pref((r_{2i}, r_{2i-1}), p)$ denote the hospital pair at position p on the joint preference list of (r_{2i-1}, r_{2i}).

For an acceptable resident-hospital pair (r_i, h_j), let $rank(h_j, r_i) = q$ denote the rank which hospital h_j assigns resident r_i where $1 \leq j \leq n_2$, $1 \leq i \leq n_1$ and $1 \leq q \leq l(h_j)$. Thus, $rank(h_j, r_i)$ is equal to the number of residents that h_j prefers to r_i plus one.

Further, for i $(1 \leq i \leq n_1)$, j $(1 \leq j \leq n_2)$, p $(1 \leq p \leq l(r_i))$ and q $(1 \leq q \leq l(h_j))$ let the set $R(h_j, q)$ contain resident integer pairs (r_i, p) such that $rank(h_j, r_i) = q$ and $pref(r_i, p) = h_j$. Hence:

$$R(h_j, q) = \{(r_i, p) \in R \times \mathbb{Z} : rank(h_j, r_i) = q \wedge 1 \leq p \leq l(r_i) \wedge pref(r_i, p) = h_j\}.$$

Intuitively, the set $R(h_j, q)$ contains the resident-position pairs (r_i, p) such that r_i is assigned a rank of q $(1 \leq q \leq l(h_j))$ by h_j and h_j is in position p $(1 \leq p \leq l(r_i))$ on r_i's preference list.

Let $A = \{\alpha_{j,q} : 1 \leq j \leq n_2 \wedge 1 \leq q \leq l(h_j)\}$ and further, for all $\alpha_{j,q} \in A$, we enforce $\alpha_{j,q} \in \{0, 1\}$. Similarly, Let $B = \{\beta_{j,q} : 1 \leq j \leq n_2 \wedge 1 \leq q \leq l(h_j)\}$ and again, for all $\beta_{j,q} \in B$, we enforce $\beta_{j,q} \in \{0, 1\}$. The intuitive meaning of the variables $\alpha_{j,q}$ and $\beta_{j,q}$ will be given later.

We now introduce the constraints that belong to the model. The text in bold before a constraint definition below shows the part of Definition 1 with which the constraint corresponds. Hence, a constraint preceded by '**Stability 1**' is intended to prevent blocking pairs described by part 1 of Definition 1.

As each resident $r_i \in R$ is either assigned to a single hospital or is unassigned, we introduce the following constraint for all i $(1 \leq i \leq n_1)$:

$$\sum_{p=1}^{l(r_i)+1} x_{i,p} = 1. \tag{1}$$

Since a hospital h_j may be assigned at most c_j residents, $x_{i,p} = 1$ where $pref(r_i, p) = h_j$ for at most c_j residents. We thus obtain the following constraint for all j $(1 \leq j \leq n_2)$:

$$\sum_{i=1}^{n_1} \sum_{p=1}^{l(r_i)} \{x_{i,p} \in X : pref(r_i, p) = h_j\} \leq c_j. \tag{2}$$

For each couple (r_{2i-1}, r_{2i}), if resident r_{2i-1} is assigned to the hospital in position p in their projected preference list then r_{2i} must also be assigned to the hospital in position p in their projected preference list. We thus obtain the following constraint for all $1 \leq i \leq c$ and $1 \leq p \leq l(r_{2i-1}) + 1$:

$$x_{2i-1,p} = x_{2i,p}. \tag{3}$$

Stability 1 - In a stable matching M in I, if a single resident $r_i \in R$ has a worse partner than some hospital $h_j \in H$ where $pref(r_i, p) = h_j$ and $rank(h_j, r_i) = q$ then h_j must be fully subscribed with better partners than r_i. Therefore, either $\sum_{p'=p+1}^{l(r_i)+1} x_{i,p'} = 0$ or h_j is fully subscribed with better partners than r_i and $\sum_{q'=1}^{q-1} \{x_{i',p''} \in X : (r_{i',p''}) \in R(h_j, q')\} = c_j$. Thus, for each i ($2c + 1 \leq i \leq n_1$) and p ($1 \leq p \leq l(r_i)$) we obtain the following constraint where $pref(r_i, p) = h_j$ and $rank(h_j, r_i) = q$:

$$c_j \sum_{p'=p+1}^{l(r_i)+1} x_{i,p'} \leq \sum_{q'=1}^{q-1} \{x_{i',p''} \in X : (r_{i',p''}) \in R(h_j, q')\}. \tag{4}$$

Stability 2(a) - In a stable matching M in I, if a couple $c_i = (r_{2i-1}, r_{2i})$ prefers hospital pair (h_{j_1}, h_{j_2}) (which is at position p_1 on c_i's preference list) to $(M(r_{2i-1}), M(r_{2i}))$ (which is at position p_2) then it must not be the case that, if $h_{j_2} = M(r_{2i})$ then h_{j_1} is under-subscribed or prefers r_{2i-1} to one of its partners in M. In the special case in which $pref(r_{2i-1}, p_1) = pref(r_{2i}, p_1) = h_{j_1}$ it must not be the case that, if $h_{j_1} = h_{j_2} = M(r_{2i})$ then h_{j_1} is under-subscribed or prefers r_{2i-1} to one of its partners in M other than r_{2i}.

Thus, for the general case, we obtain the following constraint for all i ($1 \leq i \leq c$) and p_1, p_2 ($1 \leq p_1 < p_2 \leq l(r_{2i-1})$) such that $pref(r_{2i}, p_1) = pref(r_{2i}, p_2)$ and $rank(h_{j_1}, r_{2i-1}) = q$:

$$c_{j_1} x_{2i,p_2} \leq \sum_{q'=1}^{q-1} \{x_{i',p''} \in X : (r_{i',p''}) \in R(h_{j_1}, q')\}. \tag{5}$$

However, for the special case in which $pref(r_{2i-1}, p_1) = pref(r_{2i}, p_1) = h_{j_1}$ we obtain the following constraint for all i ($1 \leq i \leq c$) and p_1, p_2 where ($1 \leq p_1 < p_2 \leq l(r_{2i-1})$) such that $pref(r_{2i}, p_1) = pref(r_{2i}, p_2)$ and $rank(h_{j_1}, r_{2i-1}) = q$:

$$(c_{j_1} - 1) x_{2i,p_2} \leq \sum_{q'=1}^{q-1} \{x_{i',p''} \in X : q' \neq rank(h_{j_1}, r_{2i}) \wedge (r_{i',p''}) \in R(h_{j_1}, q')\}. \tag{6}$$

Stability 2(b) - A similar constraint is required for the odd members of each couple. Thus, for the general case, we obtain the following constraint for all i ($1 \leq i \leq c$) and p_1, p_2 where ($1 \leq p_1 < p_2 \leq l(r_{2i})$) such that $pref(r_{2i-1}, p_1) = pref(r_{2i-1}, p_2)$ and $rank(h_{j_2}, r_{2i}) = q$:

$$c_{j_2} x_{2i-1,p_2} \in X \leq \sum_{q'=1}^{q-1} \{x_{i',p''} : (r_{i',p''}) \in R(h_{j_2}, q')\}. \tag{7}$$

Again, for the special case in which $pref(r_{2i-1}, p_1) = pref(r_{2i}, p_1) = h_{j_2}$ we obtain the following constraint for all i $(1 \leq i \leq c)$ and p_1, p_2 where $(1 \leq p_1 < p_2 \leq l(r_{2i}))$ such that $pref(r_{2i-1}, p_1) = pref(r_{2i-1}, p_2)$ and $rank(h_{j_2}, r_{2i}) = q$:

$$(c_{j_1} - 1)x_{2i-1,p_2} \leq \sum_{q'=1}^{q-1} \{x_{i',p''} \in X : q' \neq rank(h_{j_2}, r_{2i-1}) \wedge (r_{i',p''}) \in R(h_{j_2}, q')\}. \tag{8}$$

For all j $(1 \leq j \leq n_2)$ and q $(1 \leq q \leq l(h_j))$ define a new constraint such that:

$$\alpha_{j,q} \geq 1 - \frac{\sum_{q'=1}^{q-1} \{x_{i',p''} \in X : (r_{i',p''}) \in R(h_j, q')\}}{c_j}. \tag{9}$$

Thus, if h_j is full with assignees better than rank q then $\alpha_{j,q}$ may take the value 0 or 1. However, if h_j is not full with assignees better than rank q then $\alpha_{j,q} = 1$.

For all j $(1 \leq j \leq n_2)$ and q $(1 \leq q \leq l(h_j))$ define a new constraint such that:

$$\beta_{j,q} \geq 1 - \frac{\sum_{q'=1}^{q-1} \{x_{i',p''} \in X : (r_{i',p''}) \in R(h_j, q')\}}{(c_j - 1)}. \tag{10}$$

Thus, if h_j has $c_j - 1$ or more assignees better than rank q then $\beta_{j,q}$ may take the value 0 or 1. However, if h_j has less than $c_j - 1$ assignees better than rank q then $\beta_{j,q} = 1$.

Stability 3(a) - In a stable matching M in I, if a couple $c_i = (r_{2i-1}, r_{2i})$ is assigned to a worse pair than hospital pair (h_{j_1}, h_{j_2}) (where $h_{j_1} \neq h_{j_2}$) it must be the case that for some $t \in \{1, 2\}$, h_{j_t} is full and prefers its worst assignee to r_{2i-2+t}.

Thus we obtain the following constraint for all i $(1 \leq i \leq c)$ and p $(1 \leq p \leq l(r_{2i-1}))$ where $h_{j_1} = pref(r_{2i-1}, p)$, $h_{j_2} = pref(r_{2i}, p)$, $h_{j_1} \neq h_{j_2}$, $rank(h_{j_1}, r_{2i-1}) = q_1$ and $rank(h_{j_2}, r_{2i}) = q_2$:

$$\sum_{p'=p+1}^{l(r_{2i-1})+1} x_{2i-1,p'} + \alpha_{j_1,q_1} + \alpha_{j_2,q_2} \leq 2. \tag{11}$$

Stability 3(b) - In a stable matching M in I, if a couple $c_i = (r_{2i-1}, r_{2i})$ is assigned to a worse pair than (h_j, h_j) where $M(r_{2i-1}) \neq h_j$ and $M(r_{2i}) \neq h_j$ then h_j must not have two or more free posts available.

Stability 3(c) - In a stable matching M in I, if a couple $c_i = (r_{2i-1}, r_{2i})$ is assigned to a worse pair than (h_j, h_j) where $M(r_{2i-1}) \neq h_j$ and $M(r_{2i}) \neq h_j$ then h_j must not prefer at least one of r_{2i-1} or r_{2i} to some assignee of h_j in M while having a single free post.

Both of the preceding stability definitions may be modeled by a single constraint. Thus, we obtain the following constraint for i $(1 \leq i \leq c)$ and p $(1 \leq p \leq$

$l(r_{2i-1}))$ such that $pref(r_{2i-1}, p) = pref(r_{2i}, p)$ and $h_j = pref(r_{2i-1}, p)$ where $q = \min\{rank(h_j, r_{2i}),\ rank(h_j, r_{2i-1})\}$:

$$c_j \sum_{p'=p+1}^{l(r_{2i-1})+1} x_{2i-1,p'} - \frac{\sum\limits_{q'=1}^{q-1} \{x_{i',p''} \in X : (r_{i',p''}) \in R(h_j, q')\}}{(c_j - 1)}$$

$$\leq \sum_{q'=1}^{l(h_j)} \{x_{i',p''} \in X : (r_{i'}, p'') \in R(h_j, q')\}. \tag{12}$$

Stability 3(d) - In a stable matching M in I, if a couple $c_i = (r_{2i-1}, r_{2i})$ is jointly assigned to a worse pair than (h_j, h_j) where $M(r_{2i-1}) \neq h_j$ and $M(r_{2i}) \neq h_j$ then h_j must not be fully subscribed and also have two assigned partners r_x and r_y (where $x \neq y$) such that h_j strictly prefers r_{2i-1} to r_x and also prefers r_{2i} to r_y.

For each (h_j, h_j) acceptable to (r_{2i-1}, r_{2i}), let r_{min} be the better of r_{2i-1} and r_{2i} according to hospital h_j with $rank(h_j, r_{min}) = q_{min}$. Analogously, let r_{max} be the worse of r_{2i} and r_{2i-1} according to hospital h_j with $rank(h_j, r_{max}) = q_{max}$. Thus we obtain the following constraint for i $(1 \leq i \leq c)$ and p $(1 \leq p \leq l(r_{2i-1}))$ such that $pref(r_{2i-1}, p) = pref(r_{2i}, p) = h_j$.

$$\sum_{p'=p+1}^{l(r_{2i-1})+1} x_{2i-1,p'} + \alpha_{j,q_{max}} + \beta_{j,q_{min}} \leq 2. \tag{13}$$

Objective Function - A maximum cardinalilty matching M in I is a stable matching in which the largest number of residents is assigned amongst all of the stable matchings admitted by I. To maximise the size of the stable matching found we apply the following objective function:

$$\max \sum_{i=1}^{n_1} \sum_{p=1}^{l(r_i)} x_{i,p}. \tag{14}$$

Given an instance I of HRC, the above IP model J constructed from I satisfies the property that I admits a stable matching if and only if J admits a feasible solution, the full details of the proof are shown in [8]. The model has $O(m)$ binary-valued variables and $O(m + cL^2)$ constraints where m is the total length of the single residents' preference lists and the coupled residents' projected preference lists, c is number of couples and L is the maximum length of a couple's preference list. The space complexity of the model is $O(m(m + cL^2))$ and the model can be built in $O(m(m + cL^2))$ time in the worst case for an arbitrary instance.

4 Empirical Results

We ran experiments on a Java implementation of the IP models as described in Section 3 applied to randomly-generated data. We present data showing (i) the

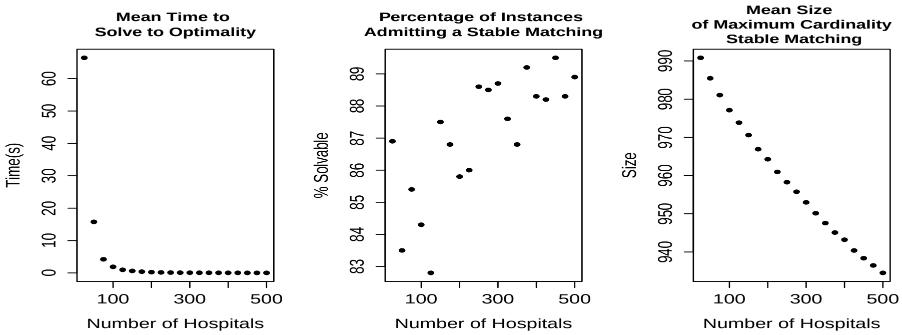

Fig. 1. Empirical Results in Experiment 1.

average time taken to find a maximum cardinality stable matching or report that no stable matching exists, and (ii) the average size of a maximum cardinality stable matching where a stable matching did exist. Instances were generated with a skewed preference list distribution on both sides, taking into account that in practice some residents and hospitals are more popular than others (on both sides, the most popular agent was approximately 3 times as popular as the least popular agent).

All experiments were carried out on a desktop PC with an Intel i5-2400 3.1Ghz processor, with 8Gb of memory running Windows 7. The IP solver used in all cases was CPLEX 12.4 and the model was implemented in Java using CPLEX Concert. We have also extended the model to cope with preference lists containing ties, and we are able to find a maximum cardinality stable matching in real data derived from the SFAS application (see [8] for further details).

Experiment 1. In our first experiment, we report on data obtained as we increased the number of hospitals in the instance while maintaining the same total number of residents, couples and posts. For various values of x ($25 \leq x \leq 500$) in increments of 25, 1000 randomly generated instances of size 1000 were created consisting of 1000 residents in total, x hospitals, 100 couples (and hence 800 single residents) and 1000 available posts which were unevenly distributed amongst the hospitals. The time taken to find a maximum cardinality stable matching or report that no stable matching existed in each instance is plotted in Figure 1 for all tested values of x. Figure 1 also shows charts displaying the percentage of instances encountered which admitted a stable matching and the mean size of a maximum cardinality stable solution for all tested values of x.

Figure 1 shows that the mean time taken to find a maximum cardinality stable matching tended to decrease as we increased the number of hospitals in the instances. We believe that this is due to the hospitals' preference lists becoming shorter, thereby reducing the model's complexity. The data in Figure 1 also shows that the percentage of HRC instances admitting a stable matching appeared to increase with the number of hospitals involved in the instance. We

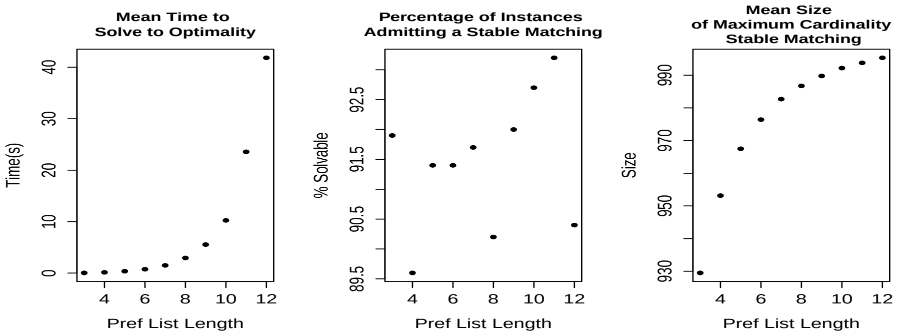

Fig. 2. Empirical Results in Experiment 2.

conjecture that this is because, as each hospital has a smaller number of posts, it is more likely to become full, and therefore less likely to be involved in a blocking pair due to being under-subscribed. Finally, the data shows that as the number of hospitals in the instances increased, the mean size of a maximum cardinality stable matching supported by the instances tended to decrease. This can be explained by the fact that, as the number of hospitals increases but the residents' preference list lengths and the total number of posts remain constant, the number of posts per hospital decreases. Hence the total number of posts among all hospitals on a resident's preference list decreases.

Experiment 2. In our second experiment, we report on data obtained as we increased the length of the individual preference lists for the residents in the instance while maintaining the same total number of residents, couples, hospitals and posts. For various values of x $(3 \leq x \leq 12)$ in increments of 1, 1000 randomly generated instances of size 1000 were created consisting of 1000 residents in total, 100 hospitals, 100 couples (and hence 800 single residents) and 1000 available posts which were unevenly distributed amongst the hospitals.

The time taken to find a maximum cardinality stable matching or report that no stable matching existed in each instance is plotted in Figure 2 for all tested values of x. Figure 2 also shows charts displaying the percentage of instances encountered admitting a stable matching and the mean size of a maximum cardinality stable solution for all tested values of x. Figure 2 shows that the mean time taken to find a maximum cardinality stable matching increased as we increased the length of the individual residents' preference lists in the instances. The data in Figure 2 also shows that the percentage of HRC instances admitting a stable matching did not appear to be correlated with the length of the individual residents' preference lists in the instances and further that as the length of the individual residents' preference lists in the instances increased, the mean size of a maximum cardinality stable matching supported by the instances also tended to increase. The first and third of these phenomena would seem to be explained by the fact that the underlying graph is simply becoming more dense.

Acknowledgement. We would like to thank the anonymous reviewers for their valuable comments.

References

1. Abraham, D.J., Biró, P., Manlove, D.F.: "Almost stable" matchings in the roommates problem. In: Erlebach, T., Persiano, G. (eds.) WAOA 2005, LNCS, vol. 3879, pp. 1–14. Springer, Heidelberg (2006)
2. Aldershof, B., Carducci, O.M.: Stable matching with couples. Discrete Appl. Math. 68, 203–207 (1996)
3. Ashlagi, I., Braverman, M., Hassidim, A.: Matching with couples revisited. Tech. Rep. arXiv: 1011.2121, Cornell University Library (2010)
4. Berman, P., Karpinski, M., Scott, A.D.: Approximation hardness of short symmetric instances of MAX-3SAT. Tech. Rep. 49, ECCC (2003)
5. Biró, P.: Student admissions in Hungary as Gale and Shapley envisaged. Tech. Rep. TR-2008-291, University of Glasgow, Dept. of Computing Science (2008)
6. Biró, P., Irving, R.W., Schlotter, I.: Stable matching with couples: an empirical study. ACM J. Exp. Algorithmics 16 (2011), section 1, article 2
7. Biró, P., Klijn, F.: Matching with couples: a multidisciplinary survey. International Game Theory Review 15(2) (2013), article number 1340008
8. Biró, P., Manlove, D.F., McBride, I.: The hospitals / residents problem with couples: Complexity and integer programming models. Tech. Rep. arXiv: 1308.4534, Cornell University Library (2013)
9. Canadian Resident Matching Service, `http://www.carms.ca`
10. Gale, D., Shapley, L.S.: College admissions and the stability of marriage. Amer. Math. Monthly 69, 9–15 (1962)
11. Gale, D., Sotomayor, M.: Some remarks on the stable matching problem. Discrete Appl. Math. 11, 223–232 (1985)
12. Gusfield, D., Irving, R.W.: The Stable Marriage Problem: Structure and Algorithms. MIT Press, Boston (1989)
13. Irving, R.W.: Matching medical students to pairs of hospitals: A new variation on a well-known theme. In: Bilardi, G., Italiano, G., Pietracaprina, A., Pucci, G. (eds.) ESA 1998, LNCS, vol. 1461, pp. 381–392. Springer, Heidelberg (1998)
14. Japan Resident Matching Program, `http://www.jrmp.jp`
15. Manlove, D.F.: Algorithmics of Matching Under Preferences. World Scientific, Singapore (2013)
16. McDermid, E.J., Manlove, D.F.: Keeping partners together: Algorithmic results for the hospitals / residents problem with couples. J. Comb. Optim. 19(3), 279–303 (2010)
17. Ng, C., Hirschberg, D.S.: Lower bounds for the stable marriage problem and its variants. SIAM J. Comput. 19, 71–77 (1990)
18. National Resident Matching Program, `http://www.nrmp.org`
19. Ronn, E.: NP-complete stable matching problems. J. Algorithms 11, 285–304 (1990)
20. Roth, A.E.: The evolution of the labor market for medical interns and residents: A case study in game theory. J. Political Economy 92(6), 991–1016 (1984)
21. Roth, A.E.: On the allocation of residents to rural hospitals: A general property of two-sided matching markets. Econometrica 54, 425–427 (1986)
22. Roth, A.E., Peranson, E.: The effects of the change in the NRMP matching algorithm. J. Amer. Medical Assoc. 278(9), 729–732 (1997)