Kwanashie, A., and Manlove, D. F. (2014) *An integer programming approach to the Hospitals/Residents problem with ties*. In: Operations Research Proceedings 2013: Selected Papers of the International Conference on Operations Research, OR2013. Series: Operations Research Proceedings . Springer International Publishing, pp. 263-269. ISBN 9783319070001

An Integer Programming approach to the Hospitals/Residents problem with Ties

Augustine Kwanashie and David F. Manlove*

School of Computing Science, University of Glasgow, UK
Email: a.kwanashie.1@research.gla.ac.uk, David.Manlove@glasgow.ac.uk

Abstract

The classical Hospitals/Residents problem (HR) models the assignment of junior doctors to hospitals based on their preferences over one another. In an instance of this problem, a stable matching M is sought which ensures that no blocking pair can exist in which a resident r and hospital h can improve relative to M by becoming assigned to each other. Such a situation is undesirable as it could naturally lead to r and h forming a private arrangement outside of the matching.

The original HR model assumes that preference lists are strictly ordered. However in practice, this may be an unreasonable assumption: an agent may find two or more agents equally acceptable, giving rise to *ties* in its preference list. We thus obtain the Hospitals/Residents problem with Ties (HRT). In such an instance, stable matchings may have different sizes and MAX HRT, the problem of finding a maximum cardinality stable matching, is NP-hard.

In this paper we describe an Integer Programming (IP) model for MAX HRT. We also provide some details on the implementation of the model. Finally we present results obtained from an empirical evaluation of the IP model based on real-world and randomly generated problem instances.

1 Introduction

The Hospital Residents Problem (HR) has applications in a number of centralised matching schemes which seek to match graduating medical students (residents) to hospital positions. Examples of such schemes include the National Resident Matching Program (NRMP) in the US [10] and the Scottish Foundation Allocation Scheme (SFAS), which ran until recently in Scotland. The challenges presented by these and other applications have motivated research in the area of algorithms for matching problems.

Formally an instance I of HR involves a set $R = \{r_1, r_2, ..., r_{n_1}\}$ of *residents* and $H = \{h_1, h_2, ..., h_{n_2}\}$ of *hospitals*. Each resident $r_i \in R$ ranks a subset of H in strict order of preference with each hospital $h_j \in H$ ranking 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 and r_i on h_j's preference list. 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. A resident r_i is *unmatched* in M if no acceptable pair in M contains r_i. We denote the hospital assigned to resident r_i in M as $M(r_i)$ (if r_i is unmatched in M then

*Supported by Engineering and Physical Sciences Research Council grant EP/K010042/1.

$M(r_i)$ is undefined) and the set of residents assigned to hospital h_j in M as $M(h_j)$. A hospital h_j is *under-subscribed* in M if $|M(h_j)| < c_j$. An acceptable pair (r_i, h_j) can *block* a matching M or forms a *blocking pair* with respect to M if r_i is either unmatched or prefers h_j to $M(r_i)$ and h_j is either under-subscribed or prefers r_i to at least one resident in $M(h_j)$. A matching M is said to be *stable* if there exists no blocking pair with respect to M.

We consider a generalisation of HR which occurs when the preference lists of the residents and hospitals are allowed to contain *ties*, thus forming the Hospital/Residents Problem with Ties (HRT). In an HRT instance a resident (hospital respectively) is indifferent between all hospitals (residents respectively) in the same tie on its preference list. In this context various definitions of stability exists. We consider *weak stability* [2] in which a pair (r_i, h_j) can *block* a matching M if r_i is either unmatched or strictly prefers h_j to $M(r_i)$ and h_j is either under-subscribed or strictly prefers r_i to at least one resident in $M(h_j)$. A matching M is said to be *weakly stable* if there exists no blocking pairs with respect to M. Henceforth we will refer to a weakly stable matching as simply a stable matching.

Every instance of the HRT problem admits at least one stable matching. This can be obtained by breaking the ties in both sets of preference lists in an arbitrary manner, thus giving rise to a HR instance which can then be solved using the Gale-Shapley algorithm for HR [1]. The resulting stable matching is then stable in the original HR instance. However, in general, the order in which the ties are broken yields stable matchings of varying sizes [7] and the problem of finding a maximum weakly stable matching given an HRT instance (MAX HRT) is known to be NP-hard [7]. Various approximation algorithms for MAX HRT can be found in the literature [8, 5] with the best current algorithm achieving a performance guarantee of 3/2.

Due to the NP-hardness of MAX HRT and the need to maximize the cardinality of stable matchings in practical applications, *Integer Programming* (IP) can be used to solve MAX HRT instances to optimality. This paper presents a new IP model for MAX HRT (Section 2). in Section 3 we provide some details on the implementation of the model. Finally Section 4 summarises some of the results obtained by evaluating the model against real-world and randomly generated problem instances. Proofs and more detailed empirical results can be found in [6].

2 An IP model for MAX HRT

In this section we describe an IP model for MAX HRT which is a non-trivial extension of an earlier IP model for a 1-1 restriction of MAX HRT due to Podhradskỳ [9]. Let I be an instance of HRT consisting of a set $R = \{r_1, r_2, ..., r_{n_1}\}$ of residents and $H = \{h_1, h_2, ..., h_{n_2}\}$ of hospitals. We denote the binary variable $x_{i,j}$ $(1 \le i \le n_1, 1 \le j \le n_2)$ to represent an acceptable pair in I formed by resident r_i and hospital h_j. Variable $x_{i,j}$ will indicate whether r_i is matched to h_j in a solution or not: if $x_{i,j} = 1$ in a given solution J then r_i is matched to h_j in M (the matching obtained from J), else r_i is not matched to h_j in M. We define $rank(r_i, h_j)$, the rank of h_j on r_i's preference list to be $k + 1$ where k is the number of hospitals that r_i strictly prefers to h_j. An analogous definition for $rank(h_j, r_i)$ holds. Obviously for HRT instances agents in the same tie have the same rank. We define $rank(r_i, h_j) = rank(h_j, r_i) = \infty$ for an unacceptable pair (r_i, h_j). With respect to a pair (r_i, h_j), we define the set $T_{i,j} = \{r_p \in R : rank(h_j, r_p) \le rank(h_j, r_i)\}$ and $S_{i,j} = \{h_q \in H : rank(r_i, h_q) \le rank(r_i, h_j)\}$. We also define the set $P(r_i)$ to be the set of hospitals that r_i finds acceptable and $P(h_j)$ to be the set of residents that h_j finds acceptable. Figure 1 shows the resulting model. Constraint 1 ensures that each resident

$$\max \sum_{i=1}^{n_1} \sum_{h_j \in P(r_i)} x_{i,j}$$

subject to

$$1. \quad \sum_{h_j \in P(r_i)} x_{i,j} \leq 1 \qquad (1 \leq i \leq n_1)$$

$$2. \quad \sum_{r_i \in P(h_j)} x_{i,j} \leq c_j \qquad (1 \leq j \leq n_2)$$

$$3. \quad c_j \left(1 - \sum_{h_q \in S_{i,j}} x_{i,q}\right) - \sum_{r_p \in T_{i,j}} x_{p,j} \leq 0 \quad (1 \leq i \leq n_1, h_j \in P(r_i))$$

$$x_{i,j} \in \{0, 1\}$$

Figure 1: **model1**: A HRT IP model

is matched to at most one hospital and Constraint 2 ensures that each hospital does not exceed its capacity. Finally Constraint 3 ensures that the matching is stable by ruling out the existence of any blocking pair.

Theorem 1. *Given a HRT instance I modeled as an IP using model1, a feasible solution to model1 produces a weakly stable matching in I. Conversely a weakly stable matching in I corresponds to a feasible solution to model1.*

3 Implementing the model

In this section we describe some techniques used to reduce the size of the HRT model generated and improve the performance of the IP solver. Techniques were described in [3] for removing acceptable pairs that cannot be part of any stable matching from HRT instances with ties on one side of the preference lists only. The *hospitals-offer* and *residents-apply* algorithms described identify pairs that cannot be involved in any stable matching, nor form a blocking pair with respect to any stable matching, and remove them from the instance. This produces a reduced HRT instance that would yield fewer variables and constraints when modelled as an IP thus speeding up the optimisation process. The original instance and the reduced instance have the same set of stable matchings.

A number of steps were taken to improve the optimisation performance of the models. These include placing a lower bound on the objective function and providing an initial solution to the CPLEX solver. Both can be obtained by executing any of the approximation algorithms [3] on the HRT instance (the 3/2-approximation algorithm for HRT with ties on one side only due to Király [4] was chosen).

4 Empirical evaluations

An empirical evaluation of the IP model was carried out. Large numbers of random instances of HRT were generated by varying certain parameters relating to the construction of the instance and passed on to the CPLEX IP solver. Data from past SFAS matching runs were also modelled and solved. This section discusses the methodology used and some of the results obtained. Experiments were carried out on a Linux machine with 8 Intel(R) Xeon(R) CPUs at 2.5GHz and 32GB RAM.

Although the theoretical model has been proven to be correct, it is still important to verify the correctness of the implementation. The system was tested to ensure a high degree of confidence in the results obtained. The correctness of the pre-processing steps and the IP solution were evaluated by generating multiple instances (100,000) of various sizes (with up to 400 residents) and testing the stability and size of the resulting matching against both the original and the trimmed problem instance. For all the instances tested, the solver produced optimal stable matchings.

Random HRT problem instances were generated. The instances consist of n_1 residents, n_2 hospitals and C posts where n_1, n_2 and C can be varied. The hospital posts were randomly distributed amongst the hospitals. Other properties of the generated instance that can be varied include the lengths of residents' preference lists as well as a measure of the density t_d of ties present in the preference lists. The tie density t_d ($0 \leq t_d \leq 1$) of the preference lists is the probability that some agent is tied to the agent next to it in a given preference list. At $t_d = 1$ each preference lists would be contained a single tie while at $t_d = 0$ no tie would exist in the preference lists of all agents thus reducing the problem to an HR instance. We define the size of the instance as the number of residents n_1 present.

Since ties cause the size of stable matchings to vary, an obvious question to investigate is how the variation in tie density affects the runtime of the IP model and the size of the maximum stable matchings found. These values were measured for multiple instances of MAX HRT while varying the tie density t_d of hospitals' preference lists. This was done for increasing sizes ($n_1 = 200, 250, 300$) of the problem instance with the residents' preference list being kept strictly ordered at 5 hospitals each. A total of 10,000 instances were randomly generated for each tie density value (starting at $t_d = 0\%$ to $t_d = 100\%$ with an interval of 5%) and instance size. For each instance $C = n_1$ and $n_2 = \lfloor 0.07 \times nR \rfloor$.

To avoid extreme outliers skewing the mean measures, we define what we regard as a reasonable solution time (300 seconds) and abandon search if the solver exceeds this cut-off time. In most cases this cut-off was not exceeded: in [6] we show the percentage of instances that were solved before the cut-off was exceeded for the values of n_1 and t_d considered (the lowest of which was 97.76%).

From Figures 2 and 3 we see that the mean and median runtime remain significantly low for instances with $t_d < 60\%$ but then gradually increase until they reach their peaks (in the region of $80\% - 90\%$) before falling as the tie density approaches 100%. From a theoretical perspective, it is known that the problem is polynomially solvable when the tie density is at both 0% and 100% and it is easy to see how the IP solver will find these cases trivial. As the tie density increases the number of stable matchings that the instance is likely to admit also increases, explaining the observed increase in the runtime. The *hospitals-offer* and *residents-apply* algorithms used to trim the instance also play their part in this trend with limited trimming done for higher tie densities.

We also looked to answer the question of how scalable the IP model is by increasing n_1 and measuring the mean and median time taken to solve multiple random instances. The tie densities of the hospitals' preference lists were set to 0.85 on all instances. The instance size n_1 was increased by 50 starting at $n_1 = 100$. A total of 100 instances for each instance size was generated. The number of hospitals n_2 in each instance was set to $\lfloor 0.07 \times n_1 \rfloor$. No cut-off was set for this experiment. Figure 4 shows how the mean computational time increases with n_1. We assume the increasingly sharp difference between the mean and median is due to the presence of outliers due to exceptionally difficult instances.

Another question worth asking is whether the IP model can handle instance sizes found in real-world applications. In [3], various approximation algorithms and heuristics were implemented and tested on real datasets from the SFAS matching scheme for 2006, 2007 and 2008 where the residents' preferences are strictly ordered with ties existing on the hospitals' preference lists. With the IP model, it is now possible to trim the instances using the techniques mentioned in Section 3, generate an optimal solution and compare the results obtained with those reported in [3]. Results from these tests showed that, while some algorithms did marginally better than others, all the algorithms developed generated relatively large stable matchings with respect to the optimal values. Table 5 shows this comparison. Let M' denote the largest stable matching found over all the algorithms tested in [3].

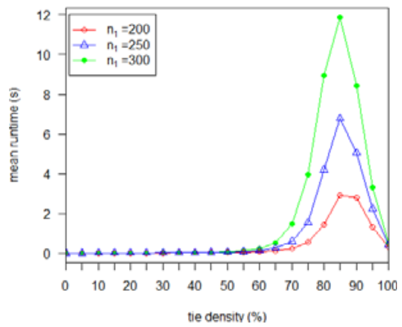

Figure 2: Mean runtime vs t_d

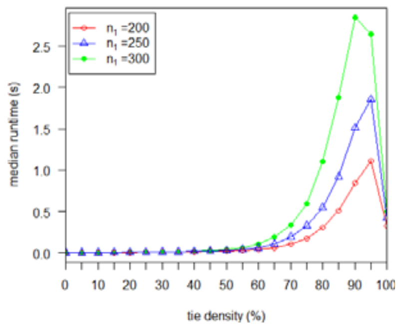

Figure 3: Median runtime vs t_d

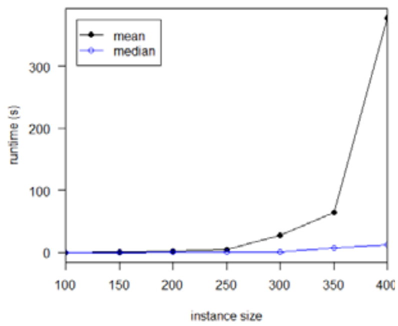

Figure 4: Mean and median runtime vs n_1

year	n_1	n_2	t_d	time (s)	$\|M\|$	$\|M'\|$ from [3]
2006	759	53	92%	92.96	758	754
2007	781	53	76%	21.78	746	744
2008	748	52	81%	75.50	709	705

Figure 5: SFAS IP Results

References

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

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

[3] R.W. Irving and D.F. Manlove. Finding large stable matchings. *ACM Journal of Experimental Algorithmics*, 14, 2009. Section 1, article 2, 30 pages.

[4] Z. Király. Better and simpler approximation algorithms for the stable marriage problem. In *Proceedings of ESA '08*, volume 5193 of *LNCS*, pages 623–634. Springer, 2008.

[5] Z. Király. Linear time local approximation algorithm for maximum stable marriage. In *Proceedings of MATCH-UP '12*, pages 99–110, 2012.

[6] A. Kwanashie and D.F. Manlove. An Integer Programming approach to the Hospitals/Residents problem with Ties. Technical Report 1308.4064, Computing Research Repository, Cornell University Library, 2013. Available from `http://arxiv.org/abs/1308.4064`

[7] 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.

[8] E. McDermid. A 3/2 approximation algorithm for general stable marriage. In *Proceedings of ICALP '09*, volume 5555 of *LNCS*, pages 689–700. Springer, 2009.

[9] A. Podhradský. *Stable marriage problem algorithms*, Master's thesis, Masaryk University, Faculty of Informatics, 2011. Available from `http://is.muni.cz/th/172646/fi_m`

[10] National Resident Matching Program website. Available at `http://www.nrmp.org`.