


# Stability Notions for Hospital Residents with Sizes

Haricharan Balasundaram 


Indian Institute of Technology Madras, Chennai, India

J. B. Krishnashree<sup>1 2</sup> 

Tata Institute of Fundamental Research Mumbai, India

Girija Limaye<sup>3</sup> 

Birla Institute of Technology and Science, Pilani, K K Birla Goa Campus, India

Meghana Nasre 

Indian Institute of Technology Madras, Chennai, India

---

## Abstract

The Hospital Residents problem with sizes (HRS) is a generalisation of the well-studied hospital residents (HR) problem. In the HRS problem, an agent  $a$  has a size  $s(a)$  and the agent occupies  $s(a)$  many positions of the hospital  $h$  when assigned to  $h$ . The notion of stability in this setting is suitably modified, and it is known that deciding whether an HRS instance admits a stable matching is NP-hard under severe restrictions. In this work, we explore a variation of stability, which we term occupancy-based stability. This notion was defined by McDermid and Manlove (J. of Comb. Opt. 2010) but remained unexplored to the best of our knowledge. In our work, we show that every HRS instance admits an occupancy-stable matching. We further show that computing a maximum-size occupancy-stable matching is NP-hard. We complement our hardness result by providing an approximation algorithm with a guarantee strictly better than 3 for the max-size occupancy-stable matching problem.

Given that the classical notion of stability adapted for HRS is not guaranteed to exist in general, we show a practical restriction under which a stable matching is guaranteed to exist. We present an efficient algorithm to output a stable matching in the restricted HRS instances. We also provide an alternate NP-hardness proof for the decision version of the stable matching problem for HRS which imposes a severe restriction on the number of neighbours of non-unit sized agents.

**2012 ACM Subject Classification** Theory of computation → Design and analysis of algorithms

**Keywords and phrases** Stable matchings, Hospital Residents with sizes, Approximation algorithms, NP-hardness

**Digital Object Identifier** 10.4230/LIPIcs.FSTTCS.2025.11

**Acknowledgements** We thank the anonymous reviewers for their comments that significantly improved the presentation of the paper.

## 1 Introduction

The Hospital Residents (HR) setting is a well-studied scenario in the literature where a set of agents, commonly called as a set of residents, needs to be assigned to hospitals. Each agent strictly orders the set of hospitals acceptable to the agent. This ordering is called the preference list. Similarly, each hospital also has a strict ordering of agents acceptable to the hospital. In addition, each hospital has a capacity that denotes the maximum number of agents that can be assigned to the hospital. The goal in the HR problem is to compute an

---

<sup>1</sup> Corresponding author.

<sup>2</sup> Part of this work was done during a semester-long project at IIT Madras, when the author was pursuing Integrated M.Sc., at PSG College of Technology, Coimbatore.

<sup>3</sup> Part of this work was done when the author was a faculty at FLAME University, Pune.



© Haricharan Balasundaram, J. B. Krishnashree, Girija Limaye, and Meghana Nasre; licensed under Creative Commons License CC-BY 4.0

45th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2025).

Editors: C. Aiswarya, Ruta Mehta, and Subhajit Roy; Article No. 11; pp. 11:1–11:18



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

assignment of agents to hospitals which respects the capacities and is *optimal* with respect to the preferences submitted. The HR problem models several important applications, namely, recruiting medical interns (residents) in hospitals [11, 6], admitting students to degree programs; for example, the Joint Seat Allocation Authority (JOSAA) that is used in undergraduate admissions in India [8, 2], and many similar scenarios [13, 1]. In most of these settings, the de facto standard of optimality with preferences is the notion of stability. Informally, an assignment is stable if no agent hospital pair wishes to deviate from the assignment. It is well known that every instance of the HR problem admits a stable assignment [6].

Since the HR setting is relevant in various practical scenarios, the problem has been extensively studied in the literature. Furthermore, several generalisations of the HR setting are investigated. In this paper, we consider a particular variant in which agents can have sizes, and we denote this setting as the HRS setting throughout the paper. An agent  $a$  in the HRS setting has a positive size  $s(a)$  associated with it. This agent can be thought of as a group of residents/students, all of whom have the same preference ordering and have to be assigned together in any assignment. Furthermore, assigning an agent  $a$  of size  $s(a)$  to a hospital  $h$  occupies  $s(a)$  many positions at  $h$ .

As mentioned in [5, 4], the HRS setting models multiple applications which include refugee settlement and accommodating children in day-cares. In the refugee settlement problem refugee families require multiple units of services (say, hospital beds) from each host locality. The refugee family can be represented as an agent with size proportional to the demand of the family, and the host locality with its service capacity can be represented by a hospital. A similar scenario arises in accommodating children in day-care centres. It can be realised where it is desirable to have siblings in the same day care or every child occupies a certain number of slots in the day care per day. The siblings (or the number of slots needed per child) can be represented by an agent of a given size, and day cares typically have a capacity representing how many children can be accommodated (or available slots). These correspond to hospitals in our setting.

In the presence of sizes for agents, the notion of stability is suitably adapted to take into account agent sizes (see Definition 1 and Definition 2). It is well known that an HRS instance may fail to admit a stable matching. Furthermore, deciding whether an instance of the HRS setting admits a stable matching is NP-hard [9]. Thus, the natural extension of the notion of stability does not offer a practical solution. In the work of McDermid and Manlove [9], which investigated the HRS problem, the authors had defined an alternate notion of stability, which we call occupancy-stable (see Definition 3 and Definition 4). We show that occupancy-stable matching always exists and can be computed efficiently.

As discussed in [9], the rationale behind such an extended definition is as follows: in order to accommodate a better-preferred agent, the hospital may have to displace a set of lower-preferred agents. However, this displacement may create more capacity at the hospital than what is needed for the better-preferred agent. The hospital may prefer being more occupied over getting better-preferred agents. The alternate notion of stability ensures that this requirement is fulfilled. This notion was defined in the concluding remarks in [9] but has not received attention to the best of our knowledge.

In our work, we investigate the stability notion adapted for HRS (henceforth termed as stability) as well as the notion of occupancy-stable matchings. We show that every instance of the HRS admits an occupancy-stable matching, and such a matching can be computed efficiently. We further show that the computation of maximum size occupancy-stable matching is NP-hard. To complement this hardness, we provide an efficient algorithm

with an approximation ratio strictly better than 3 for the maximum size occupancy-stable matching in an HRS instance. Furthermore, to circumvent the hardness of the decision version of the stable matching problem for HRS instances, we investigate restricted cases. Master lists in preferences is a well-studied restriction [7] where preferences of agents in a set are derived from a common ranking of the agents of the other side of the bipartition. We show that under a natural generalised master list restriction, every instance of the HRS setting admits a stable matching which can be efficiently computed.

## 1.1 Notation and problem definition

Let  $G = (\mathcal{A} \cup \mathcal{H}, E)$  be a bipartite graph, with  $n = |\mathcal{A} \cup \mathcal{H}|$  and  $m = |E(G)|$ . Let  $\mathcal{A}$  denote the set of agents and  $\mathcal{H}$  denote the set of hospitals. An edge  $(a, h) \in E$  indicates that  $a$  and  $h$  are mutually acceptable. For each vertex  $v \in \mathcal{A} \cup \mathcal{H}$ , we let  $\mathcal{N}(v)$  denote the set of acceptable partners of  $v$ , that is, the set of vertices from the other set that are adjacent to  $v$  in  $G$ . Each agent  $a \in \mathcal{A}$  has an integral positive size associated with it, denoted as  $s(a)$ . Let  $s_{\max} = \max_{a \in \mathcal{A}} s(a)$ . Every hospital  $h$  has an integral positive capacity  $q(h)$  associated with it, which denotes the maximum total size of agents that it can be assigned to. An agent  $a$  is called a non-unit sized agent if  $s(a) > 1$ .

Each agent  $a$  ranks hospitals  $\mathcal{N}(a)$  in a strict order, denoted as the preference list of  $a$ . Similarly, each hospital  $h$  ranks agents in  $\mathcal{N}(h)$  in a strict order, denoted as the preference list of  $h$ . For any vertex  $v \in \mathcal{A} \cup \mathcal{H}$ , we say that  $v$  prefers  $x$  over  $y$  iff  $x$  is before  $y$  in  $v$ 's preference order, and we denote this by  $x \succ_v y$  (see the example in Fig. 1, discussed later). Let  $\ell_a$  (resp.  $\ell_h$ ) denote the length of the longest preference list of an agent (resp. hospital). We abuse the notation and call the graph  $G = (\mathcal{A} \cup \mathcal{H}, E)$  along with preferences, sizes and capacities an HRS instance. It is easy to see that the HR instance [6] is a special case of HRS instance with  $s_{\max} = 1$ .

A many-to-one matching (called matching here onwards) in  $G$  is an assignment between agents and their acceptable hospitals such that every agent is matched to at most one hospital and for every hospital  $h$ , the sum of the sizes of agents matched to  $h$  is less than or equal to its capacity, that is,  $q(h)$ . Let  $M(a)$  denote the hospital matched to agent  $a$  in matching  $M$ , and  $M(h)$  denote the set of agents matched to hospital  $h$  in matching  $M$ . We assume that every agent prefers to be matched to one of its acceptable partners over being unmatched. The occupancy of  $h$  w.r.t. matching  $M$  is the total size of agents matched to  $h$  in  $M$ , denoted by  $O_M(h)$ . Thus,  $O_M(h) = \sum_{a \in M(h)} s(a)$ .

In an HR instance, an agent-hospital pair  $(a, h)$  is said to *block* matching  $M$ , if  $(a, h) \notin M$  and they both have an incentive to deviate from  $M$  and instead, get matched to each other. That is, either  $a$  is unmatched or  $h \succ_a M(a)$ , and either  $O_M(h) < q(h)$  or there exists an agent  $a' \in M(h)$  and  $a \succ_h a'$ . A matching  $M$  is stable if no pair blocks it. The notion of blocking pair and stability can be extended for the HRS setting as follows [9].

► **Definition 1** (Blocking pair). *Given a matching  $M$  in an HRS instance  $G$ , a pair  $(a, h) \in E \setminus M$  is a blocking pair w.r.t.  $M$  if  $h \succ_a M(a)$  and there exists a set of agents  $X \subseteq M(h)$ , possibly empty, such that for every  $a' \in X$ ,  $a \succ_h a'$  and  $O_M(h) - \sum_{a' \in X} s(a') + s(a) \leq q(h)$ .*

The definition implies that an unmatched edge  $(a, h)$  is a blocking pair if  $a$  is either unmatched or prefers  $h$  to its current matched partner in  $M$ , and  $h$  has sufficient capacity to accommodate  $a$  by replacing a set  $X$  of agents that are currently matched to  $h$  and are lower-preferred over  $a$ . Note that if  $X$  is empty, it implies that  $h$  can accommodate  $a$  without replacing any agent(s) in  $M(h)$ .

$$\begin{array}{lll}
(1) \ a_1 : h_2 \succ h_1 & (1) \ a_2 : h_1 \succ h_2 & (2) \ a_3 : h_2 \\
[1] \ h_1 : a_1 \succ a_2 & [2] \ h_2 : a_2 \succ a_3 \succ a_1 &
\end{array}$$

■ **Figure 1** HRS instance that does not admit a stable matching. The number preceding the agent shows the size of the agent, whereas the number preceding the hospital denotes the hospital capacity. For example,  $s(a_3) = 2$  and  $q(h_2) = 2$ . The preference list of an agent or a hospital is as shown. For example, hospital  $h_2$  prefers agent  $a_2$ , followed by agent  $a_3$ , followed by agent  $a_1$ .

► **Definition 2** (Stable matching). *A matching  $M$  in an HRS instance  $G$  is stable if there does not exist a blocking pair w.r.t.  $M$ .*

It is well-known that an HR instance always admits a stable matching. The celebrated algorithm by Gale and Shapley efficiently computes a stable matching in an HR instance [6]. In contrast to this, an HRS instance may not admit a stable matching, as illustrated in the following example from [9]. Fig. 1 shows an example instance with  $\mathcal{A} = \{a_1, a_2, a_3\}$  and  $\mathcal{H} = \{h_1, h_2\}$ . The preference lists of agents and hospitals are as shown in the Fig. 1. We observe that any matching that leaves  $a_1$  unmatched cannot be stable since  $(a_1, h_1)$  blocks the matching. Similarly, any matching that leaves  $a_2$  unmatched cannot be stable since  $(a_2, h_2)$  blocks the matching. Therefore, if a stable matching exists in the instance, it must match *both*  $a_1$  and  $a_2$ . Irrespective of where  $a_1$  and  $a_2$  are matched, the capacity constraint does not allow  $a_3$  to be matched. Thus, we consider  $M = \{(a_1, h_2), (a_2, h_2)\}$ ,  $M' = \{(a_1, h_2), (a_2, h_1)\}$  and  $M'' = \{(a_1, h_1), (a_2, h_2)\}$ . It is easy to verify that  $M$  is blocked by  $(a_2, h_1)$ ,  $M'$  is blocked by  $(a_3, h_2)$  and  $M''$  is blocked by  $(a_1, h_2)$ . Thus, the instance does not admit a stable matching.

McDermid and Manlove [9] showed that the problem of determining whether an HRS instance admits a stable matching is NP-hard. In light of this, we consider restricted settings of the instance under which the problem becomes tractable. One such restriction that has been investigated is the assumption of *master list*. The set of agents is said to have a master list ordering on them if there exists an ordering of agents such that every hospital has its preference list derived from that ordering [7]. We propose and investigate the notion of a *generalised master list* as defined below. Unlike an arbitrary HRS instance which may fail to admit a stable matching, we show that an HRS instance with the generalised master list ordering always admits a stable matching, and it can be computed efficiently.

**Generalised master list on agents.** We say that the preference lists of hospitals follow the generalised master list ordering on agents if there exists a partition  $\{\mathcal{A}_{t_1}, \mathcal{A}_{t_2}, \dots, \mathcal{A}_{t_f}\}$  of  $\mathcal{A}$  such that all agents in  $\mathcal{A}_{t_i}$  have the same size and there exists an ordering  $\langle \mathcal{A}_{t_1}, \mathcal{A}_{t_2}, \dots, \mathcal{A}_{t_f} \rangle$  on the sets in the partition of  $\mathcal{A}$  such that every hospital  $h \in \mathcal{H}$  prefers its neighbours in  $\mathcal{A}_{t_i}$  over its neighbours in  $\mathcal{A}_{t_j}$  iff  $i < j$ . We note that generalised master list ordering does *not* impose any master list ordering within a fixed set  $\mathcal{A}_{t_i}$  of the partition. We remark that the example in Fig. 1 does not follow a generalised master list since there is no way to partition the three agents satisfying the required properties. Next, we illustrate generalised master lists via an example.

Let  $\mathcal{A} = \{a_1, a_2, a_3, a_4, a_5\}$  and  $\mathcal{H} = \{h_1, h_2, h_3\}$ . The agent preferences and the hospital capacities do not matter for the illustration and hence are left unspecified. The preferences of the hospitals, the relationship between the sizes of the agents and a partition of the agent

set are shown in Fig. 2. It is noted here that in the example,  $s(a_1)$  and  $s(a_4)$  could be equal. We also remark that although the generalised master list is defined on all agents in  $\mathcal{A}$ , the individual hospitals may have an incomplete preference list as seen in the example.

$$h_1 : a_2 \succ a_1 \succ a_4 \qquad h_2 : a_1 \succ a_2 \succ a_3 \qquad h_3 : a_5 \succ a_4$$

■ **Figure 2** Illustration of generalised master lists. Assume that  $s(a_1) = s(a_2) \neq s(a_3)$  and  $s(a_4) = s(a_5) \neq s(a_3)$ . Let  $\mathcal{A} = \{\mathcal{A}_{t_1}, \mathcal{A}_{t_2}, \mathcal{A}_{t_3}\}$  where  $\mathcal{A}_{t_1} = \{a_1, a_2\}$ ,  $\mathcal{A}_{t_2} = \{a_3\}$ ,  $\mathcal{A}_{t_3} = \{a_4, a_5\}$ .

It is easy to see that the preferences of hospitals follow the generalised master list ordering on agents because there exists an ordering  $\langle \mathcal{A}_{t_1}, \mathcal{A}_{t_2}, \mathcal{A}_{t_3} \rangle$  on the sets in the partition such that every hospital  $h$  prefers all agents in  $\mathcal{A}_{t_i} \cap \mathcal{N}(h)$  over  $\mathcal{A}_{t_j} \cap \mathcal{N}(h)$  if  $i < j$ . Since the generalised master list ordering does *not* impose any ordering within a fixed set  $\mathcal{A}_{t_i}$ , we note that in Fig. 2 both  $a_1, a_2$  belong to  $\mathcal{A}_{t_1}$  and yet it is acceptable that  $a_2 \succ_{h_1} a_1$  and  $a_1 \succ_{h_2} a_2$ . We use generalised master lists in the context of the standard stability notion.

We now turn to an alternative notion of stability, which is a relaxation of the standard notion and is appealing since we show that it ensures guaranteed existence. McDermid and Manlove [9] discuss the following notion of a blocking pair, which is stronger than the one in Definition 1. We term it an occupancy-blocking pair.

► **Definition 3** (Occupancy-blocking pair). *Given a matching  $M$  in an HRS instance  $G$ , a pair  $(a, h) \in E \setminus M$  is a blocking pair w.r.t.  $M$  if  $h \succ_a M(a)$  and there exists a set of agents  $X \subseteq M(h)$ , possibly empty, such that  $\forall a' \in X, a \succ_h a'$  and  $O_M(h) - \sum_{a' \in X} s(a') + s(a) \leq q(h)$  and  $s(a) \geq \sum_{a' \in X} s(a')$ .*

That is, an edge  $(a, h)$  is an occupancy-blocking pair w.r.t. matching  $M$  if it satisfies Definition 1 and the occupancy of  $h$  does not reduce by replacing the agents in  $X$  with agent  $a$ . When a matching  $M$  does not admit any occupancy-blocking pair, then  $M$  is an occupancy-stable matching.

► **Definition 4** (Occupancy-stable matching). *A matching  $M$  in an HRS instance  $G$  is occupancy-stable if there does not exist an occupancy-blocking pair w.r.t.  $M$ .*

It is easy to note that every stable matching in an HRS instance is also occupancy-stable. For the HRS instance in Fig. 1 that does not admit any stable matching, consider the matching  $N = \{(a_1, h_1), (a_3, h_2)\}$ . The pair  $(a_2, h_2)$  is not an occupancy-blocking pair since  $s(a_3) > s(a_2)$ , hence  $N$  is occupancy stable. In this work, we show that every instance of the HRS problem admits an occupancy-stable matching and it can be efficiently computed.

The size of a matching in an HRS instance is the total occupancy across all hospitals. There are simple instances in which different occupancy-stable matchings have different sizes (see the example in Fig. 3). Hence, a natural question is to ask for an occupancy-stable matching which maximises the size. In this work, we show that computing such a matching is NP-hard. We complement it with a constant factor approximation algorithm.

## 1.2 Our results

We show the following new results for the HRS problem.

**Occupancy-stable matchings.** We show that every HRS instance admits an occupancy-stable matching, and such a matching can be efficiently computed. However, the problem of computing a maximum size occupancy-stable matching is NP-hard.

► **Theorem 5.** *An HRS instance always admits an occupancy-stable matching, and it can be computed in  $O(m + n \log n)$  time.*

► **Theorem 6.** *Given an HRS instance, computing a maximum size occupancy-stable matching is NP-hard, even if the size of each agent and the capacity of each hospital is at most 2, and the lengths of the agents' and hospitals' preference lists are at most 4.*

On the positive side, we show that there is a strictly better than 3-factor approximation for the maximum size occupancy stable matching problem.

► **Theorem 7.** *Given an HRS instance, an occupancy-stable matching with an approximation guarantee of  $\left(3 - \frac{1}{s_{\max}}\right)$  can be computed in  $O(m + n \log n)$  time, where  $s_{\max}$  denotes the maximum size of an agent.*

**Stable matchings.** We show that when the HRS instance admits a generalised master list ordering on agents, a stable matching always exists and it can be efficiently computed.

► **Theorem 8.** *Given an HRS instance with a generalised master list on agents, a stable matching always exists and it can be computed in  $O(m + n)$  time.*

We note that the following are special cases of the generalised master list setting. Therefore, our algorithmic result applies to them.

- **Master list on agents:** Suppose there is a master list ordering on agents as  $a'_1 \succ a'_2 \succ a'_3 \succ \dots$ . Let  $\mathcal{A}_{t_i} = \{a'_i\}$ . Then this is a special case of the generalised master list ordering with the ordering  $\langle \mathcal{A}_{t_1}, \mathcal{A}_{t_2}, \mathcal{A}_{t_3}, \dots \rangle$ .

Suppose that the distinct sizes of agents are  $\{s_1, s_2, \dots, s_{\max}\}$ , where  $s_i < s_j$  whenever  $i < j$ . Let  $\mathcal{A}_{s_i}$  indicate the set containing all agents with size  $s_i$ .

- **All agents with larger size are preferred over all agents with smaller size:** This is a special case of the generalised master list ordering with the order  $\langle \mathcal{A}_{t_1} = \mathcal{A}_{s_{\max}}, \dots, \mathcal{A}_{t_f} = \mathcal{A}_{s_1} \rangle$
- **All agents with a smaller size are preferred over all agents with a larger size:** This is a special case of the generalised master list ordering with the order  $\langle \mathcal{A}_{t_1} = \mathcal{A}_{s_1}, \dots, \mathcal{A}_{t_f} = \mathcal{A}_{s_{\max}} \rangle$

Next, we show an alternate hardness result for the decision version of the stable matching problem in HRS instances.

► **Theorem 9.** *For an HRS instance, deciding whether it admits a stable matching is NP-hard even when each agent with non-unit size has exactly one hospital in its preference list.*

As noted earlier, the NP-hardness proved in [9] holds even when the agents have degree at most 3 and non-unit sized agents have size 2. In our reduction, we have non-unit sized agents with degree only 1, whereas in [9] non-unit sized agents have degree 3.

### 1.3 Related work

The HRS problem is related to the hospital residents problem with couples, where certain pairs of residents are identified as a couple [9]. The couples are allowed to give individual preferences as well as joint preferences. When the preferences of the couples are consistent, then the restriction is termed as Hospital residents with consistent couples. When couples need to be assigned to the same hospital, then they are called inseparable couples [9]. The HRS problem considered here is a generalisation of hospital residents with consistent and inseparable couples. Our work builds on the work done by McDermid and Manlove [9] where they considered the HRS problem. They showed that determining whether a stable matching exists in an HRS instance is NP-hard, even when each preference list has length at most three and hospital capacities are at most two.

The work of Delacretaz *et al.* [5] considered a richer model where agents can have multidimensional constraints, which is a vector of demands from multiple services provided by the hospital. In the 2019 work by Delacretaz [4], the author worked with a restricted version of the HRS problem where only two different sized agents are considered. Since deciding whether a stable matching exists is NP-hard, other fairness notions are considered. Delacretaz [4] characterised a stable matching, if one exists, using these fairness notions.

HRS setting is also relevant to job assignment settings where each job has a size or processing time. Dean *et al.* [3] studied the problem of assigning jobs to machines with preferences, generally known as the “ordinal” assignment problem. The model studied by them is the same as the HRS setting, but their feasible matching definition is relaxed. They allowed a small violation of hospital capacity. They proposed a polynomial-time algorithm that computes an unsplit (integral) job-optimal stable assignment where each machine is over-capacitated by at most the processing time of the least preferred job. In our work, we do not permit capacities to be violated, but instead consider a different notion of stability.

Master lists are motivated by real-world matching problems where the agents involved are ranked based on a common criterion. For example, as mentioned by Irving *et al.* [7], the Medical Training Application Service (MTAS) employed score-derived master lists with extensive ties for allocating medical posts. Another example is the allocation of students to dormitories at the Technion-Israel Institute of Technology [12], where master lists are used. Our generalised master lists are in a similar spirit. In the domain of solving hard stable matching problems, Meeks *et al.* [10] consider instances involving groups of similar agents, namely typed instances. In their case, the *type* of the agent determines how the agent ranks elements in the other set as well as the agent’s preference list. In our case, the agent’s preference list does not depend on the set in the partition that it belongs to.

## 2 Our algorithm

Let  $G$  be an HRS instance and let  $\{\mathcal{A}_{t_1}, \dots, \mathcal{A}_{t_f}\}$  be a partition of  $\mathcal{A}$  of  $G$  such that all agents in set  $\mathcal{A}_{t_i}$  have the same size. We denote such a partition of agents as a *same-sized* partition. Note that all agents in a given partition have the same size, however, it is not necessary that all agents of the same size belong to a single partition. We present an algorithm that takes as input such an instance  $G$  and an ordering on a same-sized partition of  $G$  and computes a matching. Later, we show the following two independent results:

- In Section 2.1, we show that a same-sized partition of  $\mathcal{A}$  and an ordering on the sets in that partition exists such that Algorithm 1 computes an occupancy-stable matching. We further show that this matching has an approximation guarantee of strictly better than 3 with respect to the max-size occupancy-stable matching (Theorem 5 and Theorem 7).



- In Section 2.2 we assume that  $G$  admits a generalised master list ordering on agents. That is, there exists a same-sized partition of agents and an ordering on the sets of this partition. With this ordering as input, Algorithm 1 computes a stable matching (Theorem 8).

Our algorithm iterates over these sets in the given ordering. For a fixed set  $\mathcal{A}_{t_k}$ , it constructs an induced subgraph  $G_k$  of agents in the set  $\mathcal{A}_{t_k}$  and all hospitals. We note that the graph  $G_k$  enjoys the property that all agents in the set  $\mathcal{A}_{t_k}$  have the same size. Suppose the size of agents in  $G_k$  is  $s$ . The standard Gale and Shapley algorithm [6] that works with unit-sized agents can be adapted to work for uniform sized agents as follows. Agents apply in order of their preference and hospitals accept/reject based on preference. Each time an agent of size  $s$  is matched to  $h$ , the hospital  $h$  subtracts  $s$  from its available capacity instead of subtracting 1. When an agent is rejected by a hospital to accept a better preferred agent, there is no change in available capacity since all agents are of the same size. We execute this adapted version of the Gale and Shapley algorithm on  $G_k$  to compute a stable matching  $M_k$  in  $G_k$ . The edges in matching  $M_k$  computed in the  $k$ -th iteration are added to the cumulative union of stable matchings computed till then. That is,  $M^{(t_k)} = \cup_{r=1}^k M_r$ . It finally returns the result, that is,  $M^{(t_f)}$ . Algorithm 1 gives the pseudo-code.

■ **Algorithm 1** Algorithm for HRS with an ordered partition of agents.

---

**Input:** an HRS instance  $G$  with an ordering  $\langle \mathcal{A}_{t_1}, \dots, \mathcal{A}_{t_f} \rangle$  on the sets in same-sized partition of  $\mathcal{A}$

- 1:  $M^{(t_0)} = \emptyset$
- 2: **for**  $k = 1, \dots, f$  **do**
- 3:   **for**  $h \in \mathcal{H}$  **do**
- 4:      $q_k(h) \leftarrow q(h) - O_{M^{(t_{k-1})}}(h)$
- 5:   let  $G_k$  denote the sub-graph of  $G$  induced by the vertices in  $\mathcal{A}_{t_k} \cup \mathcal{H}$
- 6:   let  $M_k$  be a stable matching obtained in  $G_k$  with capacities  $q_k$  by executing the adapted version of the Gale and Shapley agent-proposing algorithm
- 7:    $M^{(t_k)} \leftarrow M^{(t_{k-1})} \cup M_k$
- 8: **return**  $M^{(t_f)}$

---

Let  $M = M^{(t_f)}$  denote the output of Algorithm 1. Before we proceed, we state important observations about our algorithm in the propositions below.

► **Proposition 10.** *The subgraphs  $G_k$  are pairwise edge-disjoint.*

► **Proposition 11.** *An edge  $(a, h) \in M$  if and only if  $(a, h) \in M_k$  such that  $a \in \mathcal{A}_{t_k}$ .*

► **Proposition 12.** *For every hospital  $h$ , the occupancy of  $h$  monotonically increases during the execution of the algorithm, that is, if  $i \leq j$  then  $O_{M^{(t_i)}}(h) \leq O_{M^{(t_j)}}(h)$ .*

A stable matching in the subgraph  $G_k$  can be computed in  $O(|E_k|)$  time using the adapted Gale and Shapley algorithm described above. Here  $E_k$  denotes the set of edges in the graph  $G_k$ . Since the graphs in each iteration are pairwise edge-disjoint (Proposition 10), we conclude that the algorithm runs in linear time in the size of  $G$ .

The agent-proposing adaptation of the Gale and Shapley algorithm used in line 6 of Algorithm 1 is for the sake of concreteness. We remark that in line 6 we can use any stable matching in the graph  $G_k$  and the proof of correctness in the following section will go through.



## 2.1 Occupancy-stable matchings

Let  $G$  be the given HRS instance. Let  $s_1, s_2, \dots, s_k$  denote the distinct sizes of agents in  $G$  and let  $s_1 > s_2 > \dots > s_k \geq 1$ . Let  $\mathcal{A}_{s_i}$  denote the set of agents with size  $s_i$ . It is clear that  $\{\mathcal{A}_{s_1}, \dots, \mathcal{A}_{s_k}\}$  is a same-sized partition of  $\mathcal{A}$ . Consider the ordering  $\langle \mathcal{A}_{s_1}, \dots, \mathcal{A}_{s_k} \rangle$  on the above partition. We note that such a same-sized partition and an ordering can be computed in  $O(n \log n)$  time where  $n = |\mathcal{A} \cup \mathcal{H}|$ . We execute Algorithm 1 using this ordering imposed in the input instance and show that Algorithm 1 computes an occupancy-stable matching in this case. We note that the partition is only with respect to the sizes and here we do *not* have any restriction on the preferences of the hospitals with respect to this partition. Recall that we let  $M = M^{(t_f)}$  denote the output of Algorithm 1.

► **Lemma 13.** *The matching  $M$  computed by Algorithm 1 is occupancy-stable.*

**Proof.** Suppose, for the sake of contradiction, that  $(a, h)$  is an occupancy-blocking pair w.r.t.  $M$ . This implies that  $h \succ_a M(a)$  and one of the following two conditions must hold:

- (i)  $O_M(h) + s(a) \leq q(h)$  or
- (ii) there exists a non-empty set of agents  $X \subseteq M(h)$  such that for every  $a' \in X$ ,  $a \succ_h a'$  and  $s(a) \geq \sum_{a' \in X} s(a')$  and  $O_M(h) - \sum_{a' \in X} s(a') + s(a) \leq q(h)$ .

We show that neither of (i) and (ii) holds, thereby leading to a contradiction to the claimed occupancy-blocking pair.

Let  $s_a = s(a)$ . Since  $(a, h)$  is an occupancy-blocking pair,  $(a, h) \notin M$ . Therefore, by Proposition 11,  $(a, h) \notin M_{s_a}$ . Consider the iteration when  $k = s_a$ . Recall that  $O_{M^{(t_k)}}(h)$  denotes the occupancy of  $h$  in matching  $M^{(t_k)}$ , that is,  $O_{M^{(t_k)}}(h)$  denotes the occupancy of  $h$  after computing the union in line 7. Since  $(a, h) \notin M_{s_a}$ , we have  $O_{M^{(t_k)}}(h) + s(a) > q(h)$ . By Proposition 12,  $O_M(h) = O_{M^{(t_f)}}(h) \geq O_{M^{(t_k)}}(h)$ , thus,  $O_M(h) + s(a) > q(h)$  holds at the end of the algorithm. Therefore, (i) does not hold at the end of the algorithm.

Next, we show that (ii) does not hold either. We first consider the case wherein for every agent  $a' \in X$ ,  $s(a') = s_a$ . Due to the stability of matching  $M_{s_a}$ , for all such agents,  $a' \succ_h a$  holds. This leads to a contradiction to the given condition that for every  $a' \in X$ ,  $a \succ_h a'$ . Therefore, there must exist an agent  $a' \in X$  such that  $s(a') \neq s(a)$ .

First, suppose that there exists an agent  $b \in X$  such that  $s(b) > s(a)$ . Then, clearly,  $\sum_{a' \in X} s(a') \geq s(b) > s(a)$ . This leads to a contradiction to the given condition that  $s(a) \geq \sum_{a' \in X} s(a')$ . Therefore, for every  $a' \in X$ ,  $s(a') < s(a)$ . This implies that all such agents  $a'$  are matched to  $h$  after  $M_{s_a}$  and  $M^{(t_k)}$  were computed.

Since  $X \subseteq M$ , we have  $O_{M^{(t_k)}}(h) + \sum_{a' \in X} s(a') \leq O_M(h)$ . Therefore,  $O_M(h) - \sum_{a' \in X} s(a') \geq O_{M^{(t_k)}}(h)$ . Adding  $s(a)$  to both sides, we get  $O_M(h) - \sum_{a' \in X} s(a') + s(a) \geq O_{M^{(t_k)}}(h) + s(a)$ . As observed earlier in this proof,  $O_{M^{(t_k)}}(h) + s(a) > q(h)$ , therefore,  $O_M(h) - \sum_{a' \in X} s(a') + s(a) > q(h)$ . This leads to a contradiction to the given condition that  $O_M(h) - \sum_{a' \in X} s(a') + s(a) \leq q(h)$ . Therefore, (ii) does not hold. This completes the proof. ◀

Before proving the approximation guarantee, we illustrate the execution of Algorithm 1 on an example. Consider the HRS instance shown in Fig. 3. For this instance,  $\mathcal{A}$  is partitioned as  $\{\mathcal{A}_{t_1}, \mathcal{A}_{t_2}\}$  such that  $\mathcal{A}_{t_1} = \{a_1\}$ ,  $\mathcal{A}_{t_2} = \{a_2, a_3\}$  where  $\mathcal{A}_{t_1}$  contains all agents of size 3 and

## 11:10 Stability Notions for Hospital Residents with Sizes

$\mathcal{A}_{t_2}$  contains all agents of size 2. Algorithm 1 computes  $M_1 = \{(a_1, h_1)\}$  and  $M_2 = \emptyset$ , thereby giving the output  $M = M^{(t_2)} = \{(a_1, h_1)\}$ . The matching  $M^* = \{(a_1, h_2), (a_2, h_1), (a_3, h_1)\}$  is the maximum size occupancy-stable matching in this instance. Thus, for this instance, we get an approximation ratio of  $\frac{s(M^*)}{s(M)} = \frac{7}{3} < 3$ . In what follows we prove that Algorithm 1 gives an approximation guarantee of strictly better than 3.

$$\begin{array}{lll} (3) & a_1 : h_1 \succ h_2 & (2) \quad a_2 : h_1 \\ [4] & h_1 : a_2 \succ a_3 \succ a_1 & (2) \quad a_3 : h_1 \\ & & [3] \quad h_2 : a_1 \end{array}$$

■ **Figure 3** Example of HRS instance used to illustrate the execution of Algorithm 1.

**Approximation guarantee.** We show that the matching  $M$  computed by Algorithm 1 is a  $(3 - \frac{1}{s_{\max}})$ -approximation for the maximum size occupancy-stable matching. Recall that the size of a matching is the total occupancy across all hospitals. Let  $s(N)$  denote the size of matching  $N$ . Then  $s(N) = \sum_{h \in \mathcal{H}} O_N(h) = \sum_{a \in \mathcal{A}, N(a) \neq \perp} s(a)$ .

Recall that  $M$  denotes the output of Algorithm 1.

► **Lemma 14.** *Let  $M^*$  be a maximum size occupancy-stable matching in  $G$ . Then  $s(M^*) \leq s(M) \cdot \left(3 - \frac{1}{s_{\max}}\right)$ .*

**Proof.** We partition the set of agents matched in  $M^*$  as follows: let  $X$  be the set of agents matched in  $M^*$  that are matched in  $M$  as well and  $Y$  be the set of agents matched in  $M^*$  but unmatched in  $M$ . That is,

- $X = \{a \mid M(a) \neq \perp \text{ and } M^*(a) \neq \perp\}$
- $Y = \{a \mid M(a) = \perp \text{ and } M^*(a) \neq \perp\}$

Clearly,  $s(M^*) = \sum_{a \in X} s(a) + \sum_{a \in Y} s(a)$ . We note that  $\sum_{a \in X} s(a) \leq s(M)$ , therefore  $s(M^*) \leq s(M) + \sum_{a \in Y} s(a)$ . Let  $N(Y)$  denote the set of hospitals  $h$  such that for some agent  $a \in Y$ ,  $M^*(a) = h$ . That is,  $N(Y)$  denotes the set of matched partners in  $M^*$  of those agents who are unmatched in  $M$ . Therefore,  $\sum_{a \in Y} s(a) \leq \sum_{h \in N(Y)} q(h)$ .

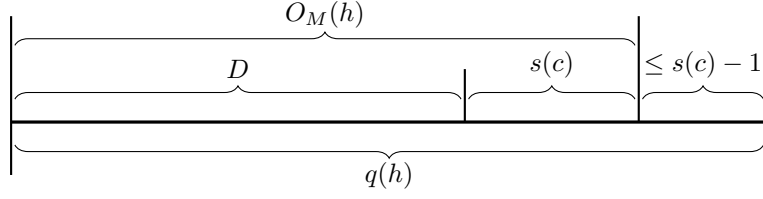
If for every hospital  $h \in N(Y)$ ,  $q(h) \leq O_M(h) \cdot \left(2 - \frac{1}{s_{\max}}\right)$ , then we get the following, thereby proving the lemma.

$$\begin{aligned} s(M^*) &\leq s(M) + \sum_{a \in Y} s(a) \leq s(M) + \sum_{h \in N(Y)} q(h) \leq s(M) + \sum_{h \in N(Y)} O_M(h) \cdot \left(2 - \frac{1}{s_{\max}}\right) \\ &\leq s(M) \cdot \left(3 - \frac{1}{s_{\max}}\right) \end{aligned} \tag{1}$$

The last inequality follows from the fact that  $\sum_{h \in N(Y)} O_M(h) \leq s(M)$ . To complete the proof,

it remains to show that for every  $h \in N(Y)$ ,  $q(h) \leq O_M(h) \cdot \left(2 - \frac{1}{s_{\max}}\right)$ , which we do next.

By the definition of  $N(Y)$ , there exists at least one agent  $a \in Y$  such that  $M^*(a) = h$ . Moreover, by the definition of  $Y$ ,  $M(a) = \perp$ . This implies that during the execution of the algorithm,  $h$  could not accommodate  $a$  during the iteration where  $M_{s(a)}$  was computed.



■ **Figure 4** Relation between  $O_M(h)$  and  $q(h)$  in terms of  $D$  and  $s(c)$ .

By Proposition 12,  $O_M(h) \geq O_{M^{(t_k)}}(h)$  where  $a \in \mathcal{A}_{t_k}$ . Since  $h$  could not occupy  $a$  in  $M_{s(a)} \subseteq M^{(t_k)}$ , we have  $s(a) > q(h) - O_{M^{(t_k)}}(h) \geq q(h) - O_M(h)$ . This implies that  $q(h) < s(a) + O_M(h)$ .

Next observe that at the end of the iteration that computed  $M_{s(a)}$ , at least one agent was matched to  $h$ . This holds, since otherwise  $h$  must have been matched to  $a$ . Moreover, the ordering imposed on the partition of the agents in input instance  $G$  implies that at that point, if  $c \in M(h)$  then  $s(c) \geq s(a)$ . Therefore, there exists an agent  $c \in M(h)$  such that  $s(c) \geq s(a)$ . A simple observation that  $O_M(h) \geq s(c)$  gives us a 3-approximation guarantee as follows: substitute  $s(a) \leq s(c) \leq O_M(h)$  in the earlier bound that we established on  $q(h)$  to get  $q(h) < 2 \cdot O_M(h)$ . Then use this bound in Equation 1.

We refine this analysis further to achieve the claimed guarantee of  $\left(3 - \frac{1}{s_{\max}}\right)$ . Using  $s(c) \geq s(a)$ , we get  $q(h) < s(c) + O_M(h) \leq s(c) - 1 + O_M(h)$ . Since  $c \in M(h)$ , we write  $O_M(h) = s(c) + D$ . Therefore,  $q(h) \leq D + 2 \cdot s(c) - 1$ . (See Fig. 4.) Thus, we get

$$\frac{q(h)}{O_M(h)} \leq \frac{D + 2 \cdot s(c) - 1}{D + s(c)} \leq \frac{2 \cdot s(c) - 1}{s(c)} = 2 - \frac{1}{s(c)} \leq 2 - \frac{1}{s_{\max}}$$

The last inequality follows from the fact that  $s(c) \leq s_{\max}$ . This completes the proof. ◀

Recall that computing an ordering on the same-sized partition takes  $O(n \log n)$  time, and Algorithm 1 runs in  $O(m + n)$  time. These facts along with Lemma 13 establish Theorem 5. Moreover, Lemma 14 establishes Theorem 7.

## 2.2 Stable matchings under generalised master list

In this section, we consider HRS instances that admit a generalised master list ordering on agents. This implies that there exists a same-sized partition  $\langle \mathcal{A}_{t_1}, \mathcal{A}_{t_2}, \mathcal{A}_{t_3}, \dots \rangle$  of  $\mathcal{A}$  and an ordering on it such that every hospital  $h \in \mathcal{H}$  prefers its neighbours in  $\mathcal{A}_{t_i}$  over its neighbours in  $\mathcal{A}_{t_j}$  iff  $i < j$ . We show that Algorithm 1 outputs a stable matching when such an ordering is given as input.

► **Lemma 15.** *Suppose that  $G$  admits a generalised master list ordering on agents. When this ordering is given as input, the matching  $M$  computed by Algorithm 1 is stable.*

**Proof.** Assume for the sake of contradiction that there is a blocking pair  $(a, h)$  w.r.t. the matching  $M$ . Suppose that  $a \in \mathcal{A}_{t_i}$  for some  $i$ . Since  $(a, h)$  blocks  $M$ ,  $(a, h) \notin M$ . By Proposition 11,  $(a, h) \notin M_{t_i}$  that was computed during the iteration when  $k = i$ .

Since  $(a, h)$  blocks  $M$ , there exists a set  $X \subseteq M(h)$  such that for every  $a' \in X$ ,  $a \succ_h a'$  and  $O_M(h) - \sum_{a' \in X} s(a') + s(a) \leq q(h)$ . Let  $s(X) = \sum_{a' \in X} s(a')$ . Since  $(a, h) \notin M_{t_i}$ , we have  $O_{M^{(t_i)}}(h) + s(a) > q(h)$ . Moreover, due to the stability of  $M_{t_i}$  and due to the assumption that the input admits a generalised master list ordering on agents,  $h$  prefers all agents matched to it in  $M^{(t_i)}$  over  $a$ . Therefore, all agents in  $X$  must be matched to  $h$  after

the  $i^{th}$  iteration as they are lower-preferred by  $h$  over  $a$ . To complete the proof, we need to show that even if we unmatch all agents in  $X$ ,  $h$  cannot accommodate the agent  $a$ , thereby getting a contradiction to the claimed blocking pair.

By Proposition 12,  $O_M(h) \geq O_{M^{(t_i)}}(h)$ . In fact, since all agents in  $X$  are matched to  $h$  after  $M^{(t_i)}$  was computed, we have  $O_M(h) \geq O_{M^{(t_i)}}(h) + s(X)$ . Thus,  $O_M(h) - s(X) \geq O_{M^{(t_i)}}(h)$ . Substituting  $O_{M^{(t_i)}}(h) > q(h) - s(a)$ , we get  $O_M(h) - s(X) > q(h) - s(a)$ . Hence,  $O_M(h) - s(X) + s(a) > q(h)$ , leading to contradiction that  $O_M(h) - s(X) + s(a) \leq q(h)$ . ◀

We assume that the ordering on the partitions implied by the generalised master list is part of the input instance  $G$ . This assumption, together with the fact that Algorithm 1 runs in  $O(m + n)$  time establish Theorem 8.

### 3 NP-hardness of max-size occupancy-stable matching

In this section, we present a polynomial-time reduction to establish NP-hardness of the problem of deciding whether an  $\mathcal{A}$ -perfect occupancy-stable matching exists in an HRS instance, thereby proving Theorem 6. An  $\mathcal{A}$ -perfect occupancy-stable matching is an occupancy-stable matching that matches all agents.

The Stable Marriage problem with Incomplete lists (SMI) is the restriction of the HR problem where capacities of all hospitals are one. Conventionally, the two sets in the Stable Marriage problem are denoted as men and women instead of agents and hospitals. The Stable Marriage problem with Ties and Incomplete lists (SMTI) is a generalisation of SMI in which preference lists can have ties. The definition of a blocking pair is suitably adapted for tied instances. A pair  $(m, w)$  that is not assigned to each other in a matching  $M$  blocks  $M$  if both  $m$  and  $w$  strictly prefer each other to their current partners in  $M$ . A matching  $M$  in an SMTI instance is stable if there is no blocking pair with respect to  $M$ .

Consider a restricted instance of SMTI with an equal number of men and women, in which every woman's preference list is a strict list of length at most three; and each man's preference list is either a strict list of length exactly three or is a single tie of length two. The goal is to decide whether there exists a complete stable matching in such a restricted SMTI instance. This is the (3,3)-COM-SMTI problem, which is known to be NP-hard [9].

Consider a (3,3)-COM-SMTI instance  $I$ , with  $n$  men  $\{m_1, m_2, \dots, m_n\}$  and  $n$  women  $\{w_1, w_2, \dots, w_n\}$ . We construct an HRS instance  $G = (\mathcal{A} \cup \mathcal{H}, E)$  from  $I$  as follows. For each woman  $w_i$  we have one hospital  $h_i$  with capacity 2. For each man  $m_j$  in  $I$  with a preference list which is a tie of length two, say  $(w_a, w_b)$  where  $a < b$ , we create a gadget consisting of 6 agents  $\{a_j^1, a_j^2, a_j^3, a_j^4, a_{j,\alpha}^1, a_{j,\alpha}^2\}$  where  $a_j^1$  and  $a_j^2$  have size 2 and rest have size 1; and 4 hospitals  $\{h_j^1, h_j^2, h_{j,\alpha}^1, h_{j,\alpha}^2\}$  with capacities 2. See Fig. 5 (i) and (ii). For each man  $m_s$  with a strict preference list in  $I$ , we create a gadget containing two agents  $a_s, a_{s,\beta}$  with size two and one respectively and one hospital  $h_{s,\beta}$  with capacity 2. See Fig. 5 (iii) and (iv). This completes the description of agents and hospitals in  $G$ . Now we describe the preference lists.

- **Preference list of hospital  $h_i$  corresponding to woman  $w_i$ :** Let the preference list of  $w_i$  be  $l_{w_i}$ . We start with preference list of  $h_i$  to be set as  $l_{h_i} = l_{w_i}$ . For  $1 \leq t \leq |l_{w_i}|$ , the man  $m_{i_t}$  in  $l_{w_i}$  either has a strict preference list in  $I$  or has a single tie of length two.
  - If  $m_{i_t}$  has a strict preference list in  $I$ , then we denote the man as  $m_s$  and replace  $m_s$  in  $l_{h_i}$  by the corresponding agent  $a_s$ .
  - If  $m_{i_t}$  has a tie in its preference list, then we denote the man as  $m_j$  and denote the preference list of  $m_j$  as  $(w_i, w_b)$  in  $I$ . If  $i < b$ , replace  $m_j$  with  $a_j^1$  in  $l_{h_i}$ , else with  $a_j^2$ .
 This gives us  $l_{h_i}$ , which is the preference list of hospital  $h_i$ .

(2) $a_j^1$ :	$h_j^1 \succ h_a \succ h_{j,\alpha}^1$	[2] $h_j^1$ :	$a_j^4 \succ a_j^1 \succ a_j^3$
(2) $a_j^2$ :	$h_j^2 \succ h_b \succ h_{j,\alpha}^2$	[2] $h_j^2$ :	$a_j^3 \succ a_j^2 \succ a_j^4$
(1) $a_j^3$ :	$h_j^1 \succ h_j^2$	[2] $h_{j,\alpha}^1$ :	$a_{j,\alpha}^1 \succ a_j^1$
(1) $a_j^4$ :	$h_j^2 \succ h_j^1$	[2] $h_{j,\alpha}^2$ :	$a_{j,\alpha}^2 \succ a_j^2$
(1) $a_{j,\alpha}^1$ :	$h_{j,\alpha}^1$		
(1) $a_{j,\alpha}^2$ :	$h_{j,\alpha}^2$		
(i)		(ii)	
(2) $a_s$ :	$h_e \succ h_f \succ h_g \succ h_{s,\beta}$	[2] $h_{s,\beta}$ :	$a_{s,\beta} \succ a_s$
(1) $a_{s,\beta}$ :	$h_{s,\beta}$		
(iii)		(iv)	

■ **Figure 5** Agents and hospitals corresponding to a man in the SMTI instance.

- **Preference list of vertices in the gadget corresponding to man  $m_j$  with a tie:** Recall that the preference list of  $m_j$  is a tie of length two, namely  $(w_a, w_b)$  where  $a < b$ . The preferences of the agents and hospitals in the gadget corresponding to  $m_j$  are shown in Fig. 5(i) and (ii). Note that the hospitals  $h_a$  and  $h_b$  corresponding to  $w_a$  and  $w_b$  appear only in the preferences of  $a_j^1$  and  $a_j^2$ , respectively.
- **Preference list of vertices in the gadget corresponding to man  $m_s$  with strict preference list:** Let the preference list of  $m_s$  be  $w_e, w_f, w_g$ . The preference list of the agent  $a_s$  has the corresponding hospitals followed by the gadget hospital  $a_{s,\beta}$ . See Fig. 5(iii) and (iv) for the preferences of the vertices in the gadget corresponding to  $m_s$ .

This completes the description of the reduction. Next, we prove its correctness.

► **Lemma 16.** *In any occupancy-stable matching  $M$  of  $G$ , either  $\{a_j^3, a_j^4\}$  are matched to the same hospital or both are unmatched.*

**Proof.** Assume  $a_j^3$  is matched to  $h_j^1$  and  $a_j^4$  is not in the occupancy-stable matching  $M$ . Then,  $(a_j^1, h_j^1)$  blocks  $M$ , contradicting the assumption that  $M$  is  $\mathcal{A}$ -perfect. Say,  $a_j^3$  is matched to  $h_j^2$ , and  $a_j^4$  is not in an occupancy-stable matching  $M$ . Since  $h_j^2$  has positions to accommodate  $a_j^4$  and  $h_j^2$  is  $a_j^4$ 's first choice,  $(a_j^4, h_j^2)$  would block  $M$ . ◀

► **Lemma 17.** *In any  $\mathcal{A}$ -perfect occupancy-stable matching  $M$  of  $G$ , exactly one of  $\{a_j^1, a_j^2\}$  is matched to their first choice.*

**Proof.** Assume both  $\{a_j^1, a_j^2\}$  are matched to their first choice in  $M$ . Then,  $\{a_j^3, a_j^4\}$  are unmatched in  $M$ , contradicting the assumption that  $M$  is  $\mathcal{A}$ -perfect. Now, assume both  $\{a_j^1, a_j^2\}$  are not matched to their first choice. By Lemma 16, without loss of generality, assume both  $a_j^3$  and  $a_j^4$  are matched to  $h_j^1$ . Then,  $(a_j^2, h_j^2)$  blocks  $M$ , as  $h_j^2$  has the capacity to accommodate  $a_j^2$ , contradicting the assumption that  $M$  is occupancy-stable. ◀

► **Lemma 18.** *In any  $\mathcal{A}$ -perfect occupancy-stable matching  $M$  of  $G$ , none of the agents in  $\{a_j^1, a_j^2, a_s\}$  are matched to their last choice hospital.*

**Proof.** Since  $M$  is  $\mathcal{A}$ -perfect, all  $\{a_j^1, a_j^2, a_s\}$  are matched to some hospital. By Lemma 17, without loss of generality, say  $a_j^1$  is matched to  $h_j^1$ . Assume  $a_j^2$  is matched to its last choice  $h_{j,\alpha}^2$ . Then  $a_{j,\alpha}^2$  remains unmatched in  $M$ , contradicting the assumption that  $M$  is  $\mathcal{A}$ -perfect. Similarly,  $a_s$  cannot be matched to  $h_{s,\beta}$ .  $\blacktriangleleft$

► **Lemma 19.** *If  $I$  admits a complete stable matching  $M$ , then  $G$  admits an  $\mathcal{A}$ -perfect occupancy-stable matching  $M'$ .*

**Proof.** Given a complete stable matching  $M$ , we describe below how to construct an  $\mathcal{A}$ -perfect occupancy-stable matching  $M'$  starting with an empty set. We need the following two sets to be defined.

$$\begin{aligned} T_j^a &= \{(a_j^1, h_a), (a_j^2, h_j^2), (a_j^3, h_j^1), (a_j^4, h_j^1), (a_{j,\alpha}^1, h_{j,\alpha}^1), (a_{j,\alpha}^2, h_{j,\alpha}^2)\} \\ T_j^b &= \{(a_j^1, h_j^1), (a_j^2, h_b), (a_j^3, h_j^2), (a_j^4, h_j^2), (a_{j,\alpha}^1, h_{j,\alpha}^1), (a_{j,\alpha}^2, h_{j,\alpha}^2)\} \end{aligned}$$

- For a man  $m_j$  with ties in his preference list  $(w_a, w_b)$  where  $a < b$ .

$$M' = M' \cup \begin{cases} T_j^a & \text{if } (m_j, w_a) \in M \\ T_j^b & \text{if } (m_j, w_b) \in M \end{cases}$$

- For a man  $m_s$  with strict preference list in  $I$ . If  $(m_s, h_i) \in M$ , then

$$M' = M' \cup \{(a_s, h_i), (a_{s,\beta}, h_{s,\beta})\}$$

Note that,  $M'$  is feasible and  $\mathcal{A}$ -perfect. To prove that  $M'$  is occupancy-stable by contradiction, assume there exists a blocking pair with respect to  $M'$ . Without loss of generality, assume  $T_j^a \in M'$ . Then,  $a_j^2, a_j^3, a_{j,\alpha}^1, a_{j,\alpha}^2$  will not be part of a blocking pair as they are matched to their first choice. Edge  $(a_j^1, h_j^1)$  will not block  $M'$  as  $h_j^1$  doesn't have enough capacity to accommodate  $a_j^1$  without unmatching higher preferred agents. Edge  $(a_j^4, h_j^2)$  will not block as  $h_j^2$  is full with higher preferred agents. If an edge  $(a_s, h_i)$  blocks  $M'$ , where  $a_s$  corresponds to a man with strict preference, the respective  $(m_s, w_i)$  would be blocking  $M$ , contradicting the assumption.  $\blacktriangleleft$

► **Lemma 20.** *If  $G$  admits an  $\mathcal{A}$ -perfect occupancy-stable matching  $M'$ , then  $I$  admits a complete stable matching  $M$ .*

**Proof.** Given  $M'$ , construct  $M$  as follows. Let  $M = \emptyset$ .

- For any  $j$  such that  $m_j$  is a man with ties in his preference list,  $(w_a, w_b)$

$$M = M \cup \begin{cases} \{(m_j, h_a)\} & \text{if } (a_j^1, h_a) \in M' \\ \{(m_j, h_b)\} & \text{if } (a_j^2, h_b) \in M' \end{cases}$$

- For any  $s$  such that  $m_s$  is a man with strict preference list.

$$M = M \cup \{(m_s, h_i)\} \quad \text{if } (a_s, h_i) \in M'$$

By Lemma 18, exactly one of  $\{a_j^1, a_j^2\}$  is matched to its top choice and other to its second choice, so there is no scenario where both  $\{(m_j, h_a), (m_j, h_b)\}$  are matched. Note that the sizes of agents and capacities of hospitals are set such that every hospital that corresponds to women has almost one match. Hence,  $M$  is feasible. By Lemma 18, every man  $m_j$  that has ties is matched to either  $w_a$  or  $w_b$ . Similarly, every man  $m_s$  with a strict preference list is matched to a woman by construction. Since all men with ties in their preference list are matched, they do not block  $M$ . Suppose that there exists a man-woman pair  $(m_s, w_i)$  which blocks  $M$  where  $m_s$  is a man with a strict preference list. Then the corresponding  $(a_s, h_i)$  blocks  $M'$  in  $G$ . Therefore,  $M$  is stable in  $I$ .  $\blacktriangleleft$

Note that, deciding the existence of an  $\mathcal{A}$ -perfect occupancy-stable matching is the same as deciding the existence of an occupancy-stable matching of size  $8|M_j| + 3|M_s|$ , where  $|M_j|$  are  $|M_s|$  are the number of men with ties in their preference list and number of men with strict preference list. Hence, Lemma 19 and Lemma 20 together establish Theorem 6.

#### 4 NP-hardness for stable matchings in HRS

In this section, we establish the NP-hardness of the decision version of the stable matching problem. Our reduction shows that the problem remains hard even when all non-unit sized agents have degree 1. This reduction is also from (3,3)-COM-SMTI. Consider a (3,3)-COM-SMTI instance  $I$ , with  $n$  men  $\{m_1, m_2, \dots, m_n\}$  and  $n$  women  $\{w_1, w_2, \dots, w_n\}$ . We construct an HRS instance  $G = (\mathcal{A} \cup \mathcal{H}, E)$  from  $I$  as follows. For each woman  $w_i$  we have one hospital  $h_i$  with capacity 1. For each man  $m_j$  in  $I$  with a preference list which is a tie of length two say  $(w_a, w_b)$  where  $a < b$ , we create a gadget consisting of twelve agents and eight hospitals as follows. See Fig. 6(i) and (ii).

- Agents  $\{a_j^1, \dots, a_j^6\}$  where  $\{a_j^1, a_j^2, a_j^3, a_j^4\}$  have unit size whereas  $\{a_j^5, a_j^6\}$  have size 3.
- For each  $k = 1, 2$  we have three agents  $\{r_{j,t}^k \mid t = 1, 2, 3\}$  where  $r_{j,1}^k, r_{j,2}^k$  have unit size and  $r_{j,3}^k$  has size 3.
- Two hospitals  $h_j^1$  and  $h_j^2$  each with capacity 3.
- For each  $k = 1, 2$ , we have three hospitals  $\{p_{j,t}^k \mid t = 1, 2, 3\}$  where  $p_{j,3}^k$  has capacity 3 and the remaining hospitals have capacity 1.

For each man  $m_s$  with a strict preference list in  $I$ , we create a gadget containing 4 agents and 3 hospitals. See Fig. 6 (iii) and (iv).

- A unit sized agent  $a_s$ .
- Three agents  $\{r_{s,t} \mid t = 1, 2, 3\}$  where  $r_{s,3}$  has size 3 and remaining agents are unit sized.
- Three hospitals  $\{p_{s,t} \mid t = 1, 2, 3\}$  where  $p_{s,3}$  has capacity 3 and the remaining hospitals have capacity 1.

This completes the description of the agents and hospitals in  $G$ . Now we describe the preference lists.

- **Preference list for hospital  $h_i$  corresponding to woman  $w_i$ :** Let the preference list of  $w_i$  be  $l_{w_i} = m_{i_1}, m_{i_2}, m_{i_3}$ . We start with preference list of  $h_i$  to be set as  $l_{h_i} = l_{w_i}$ . For  $t = 1, 2, 3$  the man  $m_{i_t}$  in  $l_{w_i}$  either has a strict preference list in  $I$  or has a single tie of length two.
  - If  $m_{i_t}$  has a strict preference list, then we denote the man as  $m_s$  and replace  $m_s$  in  $l_{h_i}$  by the corresponding agent  $a_s$ .
  - If  $m_{i_t}$  has a tie in its preference list, then we denote the man as  $m_j$  and denote the preference list of  $m_j$  as  $(w_i, w_b)$ . If  $i < b$ , replace  $m_j$  with  $a_j^1$  in  $l_{h_i}$ , else with  $a_j^2$ .

This gives us  $l_{h_i}$  which is the preference list of hospital  $h_i$ .

- **Preference lists of the vertices in the gadget corresponding to man  $m_j$  with a tie:** Recall that the preference list of  $m_j$  is a tie of length two, namely  $(w_a, w_b)$  where  $a < b$ . The preferences of the agents and hospitals in the gadget corresponding to  $m_j$  are shown in Fig. 6(i) and (ii). Note that the hospitals  $h_a$  and  $h_b$  corresponding to  $w_a$  and  $w_b$  appear only in the preferences of  $a_j^1$  and  $a_j^2$  respectively.
- **Preference lists of the vertices in the gadget corresponding to man  $m_s$  with strict preference list:** Let the preference list of  $m_s$  be  $w_e, w_f, w_g$ . Then the preference list of the agent  $a_s$  has the corresponding hospitals followed by the gadget hospital  $p_{s,1}$ . See Fig. 6(iii) and (iv) for the preferences of vertices in the gadget corresponding to  $m_s$ .

This completes the description of the reduction. We prove the correctness below.



(1) $a_j^1 :$	$h_j^1 \succ h_a \succ p_{j,1}^1$	[3] $h_j^1 :$	$a_j^3 \succ a_j^5 \succ a_j^1 \succ a_j^4$
(1) $a_j^2 :$	$h_j^2 \succ h_b \succ p_{j,1}^2$	[3] $h_j^2 :$	$a_j^4 \succ a_j^6 \succ a_j^2 \succ a_j^3$
(1) $a_j^3 :$	$h_j^2 \succ h_j^1$	[1] $p_{j,1}^k :$	$a_j^k \succ r_{j,1}^k$
(1) $a_j^4 :$	$h_j^1 \succ h_j^2$	[1] $p_{j,2}^k :$	$r_{j,2}^k \succ r_{j,1}^k$
(3) $a_j^5 :$	$h_j^1$	[3] $p_{j,3}^k :$	$r_{j,1}^k \succ r_{j,3}^k \succ r_{j,2}^k$
(3) $a_j^6 :$	$h_j^2$		
(1) $r_{j,1}^k :$	$p_{j,1}^k \succ p_{j,2}^k \succ p_{j,3}^k$		
(1) $r_{j,2}^k :$	$p_{j,3}^k \succ p_{j,2}^k$		
(3) $r_{j,3}^k :$	$p_{j,3}^k$		
(i)		(ii)	
(1) $a_s :$	$h_e \succ h_f \succ h_g \succ p_{s,1}$	[1] $p_{s,1} :$	$a_s \succ r_{s,1}$
(1) $r_{s,1} :$	$p_{s,1} \succ p_{s,2} \succ p_{s,3}$	[1] $p_{s,2} :$	$r_{s,2} \succ r_{s,1}$
(1) $r_{s,2} :$	$p_{s,3} \succ p_{s,2}$	[3] $p_{s,3} :$	$r_{s,1} \succ r_{s,3} \succ r_{s,2}$
(3) $r_{s,3} :$	$p_{s,3}$		
(iii)		(iv)	

■ **Figure 6** Agents and hospitals corresponding to a man in the SMTI instance.

► **Lemma 21.** *Let  $m_j$  be a man with a tie in his preference list in  $I$ , and let  $k \in \{1, 2\}$ . If  $G$  admits a stable matching  $N$ ,  $N$  leaves  $(a_j^k, p_{j,1}^k)$  unmatched. Furthermore,  $N$  cannot leave  $a_j^k$  unmatched.*

**Proof.** Assume there exists  $j, k$  as above such that  $(p_{j,1}^k, a_j^k) \in N$ . Hence the hospital  $p_{j,1}^k$  will be removed from  $r_{j,1}^k$ 's preference list. Now, the subgraph induced by  $\{r_{j,1}^k, r_{j,2}^k, r_{j,3}^k, p_{j,2}^k, p_{j,3}^k\}$  is exactly the instance in Fig. 1 except the sizes being 3 instead of 2 and thus doesn't admit a stable matching. The second statement of the lemma follows from the fact that if  $a_j^k$  is unmatched in  $N$  then  $(a_j^k, p_{j,1}^k)$  blocks  $N$ . ◀

Using similar arguments as above, we get the following lemma.

► **Lemma 22.** *Let  $m_s$  be a man with a strict preference list in  $I$ . Then, any matching  $N$  in  $G$  containing  $(a_s, p_{s,1})$  is not stable. Furthermore,  $N$  cannot leave  $a_s$  unmatched.*

The following two lemmas (proofs in the Appendix) establish the correctness of the reduction.

► **Lemma 23.** *Let  $j$  be such that  $m_j$  is a man with a tie in his preference list in  $I$ . If  $G$  admits a stable matching  $N$  then one of the following holds. Either  $T_j^a \subseteq N$  or  $T_j^b \subseteq N$ , where  $T_j^a = \{(a_j^1, h_a), (a_j^2, h_j^2), (a_j^3, h_j^2), (a_j^4, h_j^2), (a_j^5, h_j^1)\}$  and  $T_j^b = \{(a_j^1, h_j^1), (a_j^2, h_b), (a_j^3, h_j^1), (a_j^4, h_j^1), (a_j^6, h_j^2)\}$*

► **Lemma 24.** *The instance  $I$  admits a complete stable matching if and only if the reduced HRS instance  $G$  admits a stable matching.*

Lemma 24 establishes Theorem 9.

**Discussion.** In this paper we consider the HRS problem with a focus on the occupancy-stable matchings. We prove that computing max-sized occupancy-stable matching is NP-hard. Our algorithm gives an approximation guarantee of strictly better than 3 for the max-size occupancy-stable matching. A natural direction is to improve the approximation guarantee and establish the hardness of approximation.

## References

- 1 Atila Abdulkadiroğlu and Tayfun Sönmez. School Choice: A Mechanism Design Approach. *American Economic Review*, 93(3):729–747, 2003. doi:10.1257/000282803322157061.
- 2 Surender Baswana, Partha Chakrabarti, Sharat Chandran, Yash Kanoria, and Utkarsh Patange. Centralized admissions for engineering colleges in india. *INFORMS Journal on Applied Analytics*, 49(5):338–354, 2019. doi:10.1287/inte.2019.1007.
- 3 Brian C. Dean, Michel X. Goemans, and Nicole Immorlica. The unsplittable stable marriage problem. In *Fourth IFIP International Conference on Theoretical Computer Science- TCS 2006*, pages 65–75, 2006. doi:10.1007/978-0-387-34735-6\_9.
- 4 David Delacrétaz. Stability in matching markets with sizes. Accessed: 2025-05-05, 2017. URL: <https://daviddelacretaz.net/wp-content/uploads/2019/01/DDelacretaz-SMMS-Jan19.pdf>.
- 5 David Delacrétaz, Scott Duke Kominers, and Alexander Teytelboym. Matching mechanisms for refugee resettlement. *American Economic Review*, 113(10):2689–2717, October 2023. doi:10.1257/aer.20210096.
- 6 D. Gale and L. S. Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962. URL: <http://www.jstor.org/stable/2312726>.
- 7 Robert W. Irving, David F. Manlove, and Sandy Scott. The stable marriage problem with master preference lists. *Discrete Applied Mathematics*, 156(15):2959–2977, 2008. doi:10.1016/j.dam.2008.01.002.
- 8 Joint Seat Allocation Authority (JoSAA). Josaa - joint seat allocation authority official website. <https://josaa.nic.in/>, 2025. Accessed: 2025-05-23.
- 9 Eric J. McDermid and David F. Manlove. Keeping partners together: algorithmic results for the hospitals/residents problem with couples. *Journal of Combinatorial Optimization*, 19(3):279–303, 2010. doi:10.1007/s10878-009-9257-2.
- 10 Kitty Meeks and Baharak Rastegari. Solving hard stable matching problems involving groups of similar agents. *Theoretical Computer Science*, 844:171–194, 2020. doi:10.1016/j.tcs.2020.08.017.
- 11 National Resident Matching Program. National resident matching program website. <https://www.nrmp.org>, 2023. Accessed: 2025-05-23.
- 12 Nitsan Perach, Julia Polak, and Uriel G Rothblum. A stable matching model with an entrance criterion applied to the assignment of students to dormitories at the technion. *International Journal of Game Theory*, 36:519–535, 2008. doi:10.1007/s00182-007-0083-4.
- 13 Alvin E. Roth. On the Allocation of Residents to Rural Hospitals: A General Property of Two-Sided Matching Markets. *Econometrica*, 54(2):425–427, 1986. URL: <http://www.jstor.org/stable/1913160>.

## A Omitted proofs

**Proof of Lemma 23.** Since  $N$  is stable, from Lemma 21 the agent  $a_j^1$  should be matched to one of  $\{h_j^1, h_a\}$ . We divide the proof based on the matched partner of  $a_j^1$  in  $N$ .

- Assume  $(a_j^1, h_a) \in N$ . Here we establish that  $T_j^a \subseteq N$ . In this case for  $N$  to be stable, the hospital  $h_j^1$  has to be fully subscribed in  $N$  with agents higher preferred than  $a_j^1$ . This implies that  $(a_j^5, h_j^1) \in N$ . In this case the agent  $a_j^3$  cannot be left unmatched otherwise

- $(a_j^3, h_j^1)$  blocks  $N$ . Thus,  $(a_j^3, h_j^2) \in N$ . Since  $(a_j^3, h_j^2) \in N$ ,  $(a_j^6, h_j^2) \notin N$ . The hospital  $h_j^1$  is matched to  $a_j^5$  and therefore cannot accommodate  $a_j^4$ . The matching  $N$  cannot leave  $a_j^4$  unmatched, else  $(a_j^4, h_j^2)$  blocks  $N$ . This implies that  $(a_j^4, h_j^2) \in N$ . Given the current set of edges in  $N$ , if  $N$  does not match  $a_j^2$  to  $h_j^2$ , then  $(a_j^2, h_j^2)$  blocks  $N$ . Thus  $(a_j^2, h_j^2) \in N$ . Thus we have proved that if  $(a_j^1, h_a) \in N$  then  $T_j^a \in N$ .
- Assume  $(a_j^1, h_j^1) \in N$ . Here, we establish that  $T_j^b \subseteq N$ . Since  $a_j^1$  occupies unit capacity in  $h_j^1$ ,  $a_j^5$  cannot occupy  $h_j^1$ . Hence,  $(a_j^5, h_j^1) \notin N$ . Furthermore,  $(a_j^3, h_j^1) \in N$  else  $(a_j^5, h_j^1)$  blocks  $N$ . If  $a_j^4$  is not matched to  $h_j^1$ ,  $(a_j^4, h_j^1)$  blocks  $N$ . Therefore,  $(a_j^4, h_j^1) \in N$ . Given the current set of edges in  $N$ , if  $a_j^6$  is not matched to  $h_j^2$  then  $(a_j^6, h_j^2)$  block  $N$ . So,  $(a_j^6, h_j^2) \in N$ . From Lemma 21  $a_j^2$  cannot be unmatched and  $h_j^2$  cannot accommodate  $a_j^2$ . Therefore,  $(a_j^2, h_b) \in N$ . This concludes that  $T_j^b \subseteq N$ .

This completes the proof.  $\blacktriangleleft$

**Proof of Lemma 24.** Consider a complete stable matching  $M$  of  $I$ . Let  $G$  be the reduced HRS instance of  $I$ . Construct a matching  $M'$  as follows, starting with  $M' = \phi$ .

- For a man  $m_j$  with ties in his preference list  $(w_a, w_b)$  where  $a < b$ .

$$M' = M' \cup \begin{cases} T_j^a & \text{if } (m_j, w_a) \in M \\ T_j^b & \text{if } (m_j, w_b) \in M \end{cases}$$

Furthermore for the man  $m_j$ , we add  $M' = M' \cup S_j^1 \cup S_j^2$ .

- For a man  $m_s$  with strict preference list in  $I$

$$M' = M' \cup \{(a_s, h_i)\} \cup S_s$$

It can be verified that  $M'$  is feasible in  $G$ . From Lemma 23,  $M'$  is stable.

To prove the other direction, consider a stable matching  $M'$  of  $G$ . Construct a matching  $M$  in  $I$  as follows.

- For any  $j$  such that  $m_j$  is a man with ties in his preference list,  $(w_a, w_b)$

$$M = M \cup \begin{cases} \{(m_j, h_a)\} & \text{if } T_j^a \subseteq M' \\ \{(m_j, h_b)\} & \text{if } T_j^b \subseteq M' \end{cases}$$

- For any  $s$  such that  $m_s$  is a man with strict preference list.

$$M = M \cup \{(m_s, h_i)\} \quad \text{if } (a_s, h_i) \in M'$$

Since the capacity of  $h_i$  is 1, the corresponding woman  $w_i$  in  $I$  is also matched to at most one man. Every  $a_j$  corresponding to agent  $m_j$  with ties in his preference list will either be matched to  $h_a$  or  $h_b$ , so is  $m_j$  to  $m_a$  or  $m_b$  in  $I$ . By Lemma 22,  $a_s$  has to be matched to one of hospital in  $\{h_e, h_f, h_g\}$  in  $M$ . So, all men with a strict preference list in  $I$  are matched to some woman in  $I$ . Therefore,  $M$  is feasible and complete in  $I$ . Since all men with ties in their preference list are matched, they do not block  $M$ . Say, there exists a man-woman pair  $(m_s, w_i)$  which blocks  $M$  where  $m_s$  is a man with a strict preference list. Then corresponding  $(a_s, h_i)$  blocks  $M'$  in  $G$ . Therefore,  $M$  is stable in  $I$ .  $\blacktriangleleft$