Popular Matchings in the Hospital-Residents Problem with Two-Sided Lower Quotas

Meghana Nasre ✉
IIT Madras, Chennai, India

Prajakta Nimbhorkar ✉
Chennai Mathematical Institute, India
UMI ReLaX, Chennai, India

Keshav Ranjan ✉
IIT Madras, Chennai, India

Ankita Sarkar ✉
Dartmouth College, Hanover, NH, USA

Abstract

We consider the hospital-residents problem where both hospitals and residents can have lower quotas. The input is a bipartite graph $G = (\mathcal{R} \cup \mathcal{H}, E)$, each vertex in $\mathcal{R} \cup \mathcal{H}$ has a strict preference ordering over its neighbors. The sets $\mathcal{R}$ and $\mathcal{H}$ denote the sets of residents and hospitals respectively. Each hospital has an upper and a lower quota denoting the maximum and minimum number of residents that can be assigned to it. Residents have upper quota equal to one, however, there may be a requirement that some residents must not be left unassigned in the output matching. We call this as the residents' lower quota.

We show that whenever the set of matchings satisfying all the lower and upper quotas is non-empty, there always exists a matching that is popular among the matchings in this set. We give a polynomial-time algorithm to compute such a matching.

2012 ACM Subject Classification Mathematics of computing → Combinatorial algorithms

Keywords and phrases Matching, Popularity, Lower quota, Preferences

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2021.30

Funding *Meghana Nasre*: Partially funded by SERB grant CRG/2019/004757.
Prajakta Nimbhorkar: Partially funded by Infosys Grant and SERB grant CRG/2019/004757.
Ankita Sarkar: Part of the work was done when the author was a Master's student at Chennai Mathematical Institute, India.

1 Introduction

The stable marriage problem and its many-to-one generalization, namely the hospital residents (HR) problem, have been extensively investigated in the literature. In this work, we consider a generalization of the HR problem where hospitals and residents both can specify *demand* constraints. More formally, the input to our problem is a bipartite graph $G = (\mathcal{R} \cup \mathcal{H}, E)$ where $\mathcal{R}$ denotes the set of residents, $\mathcal{H}$ denotes the set of hospitals, and an edge $(r, h) \in E$ denotes that r and h are mutually acceptable to each other. Every resident and hospital specify a strict ranking of acceptable elements to them, and this ranking is called the *preference list* of the agent. Every hospital h has two additional inputs associated with it - $q^+(h)$, the capacity or upper quota of h, and $q^-(h)$, the demand or lower quota of h. The upper quota denotes the maximum number of residents that can be matched to h, and the lower quota denotes the minimum number of residents that must be assigned to h in any feasible assignment. In practical scenarios, residents may also have demands. That is, some residents must be matched in a round of assignments of residents (medical interns) to hospitals. We model this by allowing residents to specify an integral lower quota as a part of the input. Thus, associated with every resident $r \in \mathcal{R}$ we have $q^-(r) \in \{0, 1\}$ and $q^+(r) = 1$. We denote this as the HR2LQ problem. When all residents have lower quota zero, it is denoted as the HRLQ problem, which is well investigated in the literature. A matching M in G is a subset of the edge set E. Our goal for the HR2LQ problem is to compute a *feasible* matching that is *optimal* with respect to the preferences specified by the agents.

▶ **Definition 1** (Feasible matching). *A* feasible matching *M in $G = (\mathcal{R} \cup \mathcal{H}, E)$ is a subset of E such that $q^-(v) \leq |M(v)| \leq q^+(v)$ for each $v \in \mathcal{R} \cup \mathcal{H}$, where $M(v)$ is the set of neighbours of v assigned to v in M.*

Before we discuss the notion of optimality, we outline the challenges that lower quotas pose in the HRLQ problem, a special case of the HR2LQ problem. In the presence of two-sided preferences but no lower quotas, *stability* is a well-accepted notion of optimality. Stable matchings are characterized by the absence of a blocking pair.

▶ **Definition 2** (Blocking pair). *Given a matching M, a pair $(r, h) \in E \setminus M$ is called a* blocking pair *with respect to M if either r is unmatched or r prefers h over its matched partner $M(r)$ and either h is under-subscribed ($|M(h)| < q^+(h)$) or h prefers r over at least one of the residents matched to it, that is, some resident in $M(h)$. A matching M is* stable *if there is no blocking pair with respect to M.*

A stable matching always exists; however, a stable *and* feasible matching need not exist in the presence of lower quotas for even the hospitals alone (HRLQ problem). Nasre and Nimbhorkar [20] used an alternate notion of optimality, namely *popularity* to circumvent the problem. Informally, a matching is *popular* if no *majority* of agents wish to deviate from the matching. In [20], the authors show that for every instance of the HRLQ problem, there exists a feasible matching that is *popular* amongst the set of feasible matchings, and such matching can be computed efficiently. In our work, we show that in the presence of lower quotas on both sides of the bipartition, that is, in the HR2LQ setting, a matching that is popular amongst the set of feasible matchings exists, and it can be efficiently computed.

The setting consisting of two-sided preferences and lower quotas has received a lot of attention, e.g. Huang [9] investigate it for stable matchings where there are classifications along with lower quotas. Fleiner and Kamiyama [6] consider stable matchings with matroid constraints and lower quotas. Mnich and Schlotter [19] investigate lower quotas on both sides in the restricted stable marriage setting (one-to-one). Popularity in the HR2LQ setting has not been investigated so far. The HR2LQ problem is well motivated by several practical applications. Hospital lower quotas are important for the smooth functioning of hospitals. Similarly, some residents may have to be matched because of their economic backgrounds or because they are unallocated from the previous year. Another setting where HR2LQ arise is the allocation of mentors to students. The mentors need a group of students to carry out discussion sessions, whereas it may be required that the students whose CGPA falls below a certain threshold must get a mentor. Another application of HR2LQ is in elective allocation. While allotting electives to students, it is natural to have a lower quota on electives denoting the minimum number of students required for the elective to be offered, and the students who are in their final semester must be assigned an elective.

Notion of popularity. The notion of popularity used here is the same as in [20]. It is based on votes cast by each vertex to compare two given matchings M and N. A resident r votes for M if r prefers $M(r)$ over $N(r)$, and vice versa. If $M(r) = N(r)$ then r is indifferent between the two matchings, and hence does not cast any vote. If r is unmatched in M, we define $M(r) = \bot$, and r prefers any hospital in its preference list over $\bot$. We denote $vote_r(M, N)$ to denote the vote of r between M and N. It takes values 1, -1 or 0 depending on whether r prefers $M(r)$ over $N(r)$ or vice versa, or is indifferent between them.

For a hospital h, there can be up to $q^+(h)$ residents in $M(h)$ and $N(h)$. If h is under-subscribed in M or N then we assume that the remaining positions of h are matched to $\bot$. In this way, we can always assume that $|M(h)| = |N(h)| = q^+(h)$. So the hospital can cast up to $q^+(h)$ votes. The hospital is indifferent between M and N as far as the residents in $M(h) \cap N(h)$ are concerned. For the remaining residents, h needs to decide a correspondence $\mathbf{corr}_h$ for comparing $M(h) \setminus N(h)$ to $N(h) \setminus M(h)$. Then

$$vote_h(M, N, \mathbf{corr}_h) = \sum_{r \in M(h) \setminus N(h)} vote_h(r, \mathbf{corr}_h(r, M, N))$$

Here $\mathbf{corr}_h(r, M, N)$ denotes the resident $r' \in N(h) \setminus M(h)$ corresponding to r, and $vote_h(r, \mathbf{corr}_h(r, M, N))$ is 1 if h prefers r over r', is -1 if h prefers r' over r, and 0 if $r = r'$.

Finally the number of votes that the matching M gets over N is given by

$$\Delta(M, N, \mathbf{corr}) = \sum_{r \in \mathcal{R}} vote_r(M, N) + \sum_{h \in \mathcal{H}} vote_h(M, N, \mathbf{corr}_h)$$

▶ **Definition 3** (Popular Matching [20]). *A matching M is more popular than N (denoted as $M \succ_{\mathbf{corr}} N$) under **corr** if $\Delta(M, N, \mathbf{corr}) > 0$. A matching M is popular if there is no matching N such that $N \succ_{\mathbf{corr}} M$ for any choice of **corr** from N to M.*

Related work and techniques. The notion of popularity was introduced by Gärdenfors [7] in 1975 as a majority assignment in the context of a full stable marriage problem. In 2005, Abraham et al. [1] discussed an efficient algorithm for computing popular matching in a bipartite graph where only one set of the partition has preferences. Since then popular matchings have been well-studied and a vast literature [2, 10, 12, 8, 4, 18, 13] on popular matchings is available.

Huang and Kavitha [10] and subsequently Kavitha [12] studied the popular matchings as an alternative to stability in the stable marriage setting (where there are no lower quotas). Their motivation was to obtain matchings larger in size than the stable matching, which are optimal with respect to the preferences. Subsequently, for matchings in bipartite graphs with two-sided preferences, popularity has been investigated in the many-to-one setting (HR problem) [21], many-to-many setting [3] and the many-to-one setting with hospital lower quotas (HRLQ problem) [20, 18].

Independent of our work, very recently, Kavitha [15] investigates popularity of a matching in a stable marriage instance when both sides have lower quotas – denoted as critical nodes in her work. Her approach is similar to ours and finds a maximum size popular matching amongst the set of critical matchings. A critical matching is one which matches as many critical nodes as possible. Thus, in her work, it is not required to have the guarantee that the input instance admits a feasible matching, which is required in our work. We remark that our algorithm is for the HR2LQ setting which allows lower-quotas (or equivalently critical nodes) on both sides and non-unit upper-quotas on one side of the bipartition. Our algorithm can be easily extended to compute a maximum size feasible popular matching.

We comment on the two related but seemingly different ways of computing a popular matching used in the literature. The first approach used in [12, 18, 3] is what we term as *the modified Gale and Shapley (GS)* approach. This involves the first round of proposals using the standard GS algorithm, followed by the second round of proposals where unmatched vertices are allowed to propose with *increased priority*. This simple and elegant idea has its origins in Kirlay's work [17] on computing constant factor approximation to maximum sized stable matching in the presence of ties in preferences.

The second approach used in [21, 4] is to *simulate* the modified GS algorithm via reducing the original instance to a stable matching instance G'. A stable matching in G' is mapped to a popular matching in G. We call this the *reduction approach*. For example, in [20], the authors use the reduction approach and convert the input HRLQ instance into an HR instance where the standard GS algorithm is used to compute a matching. The reduced instance consists of multiple copies of all the hospitals with a positive lower quota. A modified GS approach on an HRLQ instance would involve giving multiple higher priorities to *deficient* hospitals, i.e. the hospitals whose lower quotas are not met. The two different approaches are indeed equivalent in the HRLQ setting since the reduction essentially simulates the modified GS algorithm.

Extension to HR2LQ. A natural extension of the modified GS approach for HR2LQ would be to execute one round of GS algorithm with say hospitals proposing, then letting deficient hospitals (if any) propose with multiple higher priorities, and then letting deficient residents (if any) propose with multiple higher priorities. In other words, one side, let's say $\mathcal{H}$, start proposing using the standard GS algorithm to compute round 1 matching M_1, which is stable. If a hospital h remains deficient in M_1 ($|M_1(h)| < q^-(h)$), then h is allowed to propose with an increased priority. A resident will always accept the proposal from a higher priority hospital by rejecting a lower priority one. A hospital keeps proposing with increased priority if it exhausts its preference list and is left deficient at the current priority. The priority of a hospital is increased (by one at a time) until it becomes non-deficient. It can be shown that no hospital is deficient in the matching M_2 obtained at the end of round 2. But a resident may still be deficient and hence, a deficient resident r is allowed to propose (possibly with increased priority) until it becomes non-deficient. In this process, another resident r' may get *rejected* for the *first time* by a hospital h because h has got a proposal from a higher priority resident and $|M(h)| = q^+(h)$. Now at this point, r' has two choices either (a) start proposing from the beginning of its list or (b) propose the hospital just after h in its preference list and continue. If r' is such that $q^-(r') = 0$, then it continues proposing hospitals until either it gets matched to some hospital or has exhausted its preference list. But if $q^-(r') = 1$ and r' has exhausted its preference list without getting any match, then it starts proposing from the beginning of the list with increased priority. In the end, we get the round 3 matching M_3.

Note that in the HRLQ setting, the modified GS approach comprises only round 1 and round 2 and gives a popular matching. But the example in Figure 1 shows that this approach in the case of HR2LQ does not yield a popular matching using any of the two choices mentioned above in round 3.

The round 1 matching computed by $\mathcal{H}$-proposing GS algorithm is $M_1 = \{(h_1, \bot), (h_2, \bot), (h_3, r_3), (h_4, r_1), (h_5, r_2), (h_6, \bot), (h_7, r_5), (h_8, \bot)\}$. Note that the matching M_1 is not feasible as the hospitals h_1, h_2 and h_6 are deficient. In round 2, these hospitals increase their priority to 1 (from 0) and start proposing from the start of the list. In order to remove the deficiency, h_1 increases its priority to 2 whereas h_2 and h_6 got matched while being at priority level 1. The matching M_2 at the end of round 2 is $M_2 = \{(h_1^2, r_1), (h_2^1, r_2), (h_3, r_3), (h_4, \bot), (h_5, \bot), (h_6^1, r_5), (h_7, \bot), (h_8, \bot)\}$. In M_2, the resident r_2 is deficient and hence round 3 starts.

$[0,1]$	r_1 :	h_4	h_2	h_1	
$[0,1]$	r_2 :	h_5	h_2	h_6	h_1
$[1,1]$	r_3 :	h_3	h_2		
$[1,1]$	r_4 :	h_3			
$[0,1]$	r_5 :	h_7	h_6	h_8	

(a) Preference List of Residents with Quotas $[q^-(r), 1]$.

$[1,1]$	h_1 :	r_1	r_2	
$[1,1]$	h_2 :	r_1	r_2	r_3
$[0,1]$	h_3 :	r_3	r_4	
$[0,1]$	h_4 :	r_1		
$[0,1]$	h_5 :	r_2		
$[1,1]$	h_6 :	r_2	r_5	
$[0,1]$	h_7 :	r_5		
$[0,1]$	h_8 :	r_5		

(b) Preference List of Hospitals with Quotas $[q^-(h), q^+(h)]$.

Figure 1 Counter-example for the modified GS approach.

In round 3, r_3 and r_4 try to snatch h_3 from each other and they raise their priority level to 1. The hospital h_2 rejects r_2 when it receives a proposal from r_3^1 and this was the first rejection of r_2 in this round. So, following the first choice of round 3, r_2 proposes h_5 and got matched to it. This results in the matching $M_3' = \{(h_1^2, r_1), (h_2^1, r_3^1), (h_3, r_4^2), (h_4, \bot), (h_5, r_2), (h_6^1, r_5), (h_7, \bot), (h_8, \bot)\}$. When we remove the priority levels we get $M' = \{(h_1, r_1), (h_2, r_3), (h_3, r_4), (h_5, r_2), (h_6, r_5)\}$. If the procedure follows the second choice of round 3, then r_2 proposes to h_6 which in turn reject r_5 and then r_5 proposes to h_8 and get matched to it. Round 3 ends here and it results in the matching $M_3'' = \{(h_1^2, r_1), (h_2^1, r_3^1), (h_3, r_4^2), (h_4, \bot), (h_5, \bot), (h_6^1, r_2), (h_7, \bot), (h_8, r_5)\}$ which maps to $M'' = \{(h_1, r_1), (h_2, r_3), (h_3, r_4), (h_6, r_2), (h_8, r_5)\}$.

We remark that neither M' nor M'' is popular as there exists another matching $M = \{(h_1, r_1), (h_2, r_6), (h_3, r_4), (h_6, r_2), (h_7, r_5)\}$ which is more popular than both.

Although the modified GS approach does not work, a natural extension of the reduction approach works. Thus we present a reduction from the HR2LQ problem to the HR problem and show that a stable matching in the reduced HR instance can be translated to a popular matching in the original instance. Thus the reduction approach works but seems to have no straightforward analogous modified GS approach. Our correctness proof is inspired by the one in [3], which uses LP duality to prove the popularity of their matching. They exhibit a dual assignment as a certificate for the popularity of the matching. Dual certificate for proving the popularity has been used earlier in the literature [13, 11, 5, 14, 16].

However, the algorithm in [3] uses a modified GS approach and has no lower quotas. Also, in [3], only one side of the bipartition gets *one* higher priority. So the dual certificate consists of a $\{\pm 1, 0\}$ assignment to all the dual variables. In contrast, our reduction makes multiple copies of residents and hospitals. As a consequence, exhibiting the dual assignment is more involved and needs weights linear in the size of the input instance.

Formally, our result is stated below:

▶ **Theorem 4.** *In an HR2LQ instance that admits a feasible matching, there always exists a matching that is popular amongst all the feasible matchings. Moreover, such a popular matching can be computed in time polynomial in the size of the input instance.*

Organization of the paper. We describe our reduction in Section 2. The feasibility and popularity proofs appear in Section 3 and Section 4 respectively. Section 5 concludes the paper.

2 Reduction

Given an HR2LQ instance $G = (\mathcal{H} \cup \mathcal{R}, E)$ we construct an HR instance $G' = (\mathcal{H}' \cup \mathcal{R}', E')$ as follows. Let μ_R and μ_H denote respectively the sum of lower quotas of residents and hospitals in the instance G. That is, $\mu_R = \sum_{r \in R} q^-(r)$ and $\mu_H = \sum_{h \in H} q^-(h)$. Let $\mathcal{R}_{lq}$ and $\mathcal{H}_{lq}$ denote the set of residents and set of hospitals with positive lower quota. That is, $\mathcal{R}_{lq} = \{ r \mid r \in \mathcal{R} \textit{ and } q^-(r) = 1\}$ and $\mathcal{H}_{lq} = \{ h \mid h \in \mathcal{H} \textit{ and } q^-(h) > 0\}$. A resident in $\mathcal{R}_{lq}$ is called a *lower-quota* resident and a hospital in $\mathcal{H}_{lq}$ is called *lower-quota* hospital. Our instance G' has the following residents and hospitals:

1. **Resident Copies:** For every resident $r \in \mathcal{R}_{lq}$, we have $\mu_R + 1$ copies of r in G'. These copies are $r^0, r^1, \ldots, r^{\mu_R}$ where r^x is called the level-x copy of r. Note that all $r \in \mathcal{R}_{lq}$ have both upper and lower quota equal to one in the HR2LQ instance G. The capacities in G' for these copies are: level-0 copy has capacity equal to the upper quota of r and all other copies have capacity equal to the lower quota of r. That is,
   $$q(r^x) \quad = \quad 1 \qquad \text{for } 0 \le x \le \mu_R$$
   A resident $r \notin \mathcal{R}_{lq}$ in G has exactly one copy r^0 in G' and we call it the level-0 copy of the resident r. The capacity of the level-0 copy in G' is $q(r^0) = 1$. We denote this set of copies of the residents as $\mathcal{R}'_c$ and call them *true* residents.
2. **Hospital Copies:** For every hospital $h \in \mathcal{H}_{lq}$, we have $\mu_H + 1$ copies of h in G'. These copies are $h^0, h^1, \ldots, h^{\mu_H}$ where h^y is called the level-y copy of h. The capacities in G' for these copies are: level-0 copy has capacity equal to the upper quota of h and all other copies have capacity equal to the lower quota of h. That is,
   $$\begin{aligned} q(h^y) &= q^+(h) && \text{if } y = 0 \\ &= q^-(h) && \text{if } 1 \le y \le \mu_H \end{aligned}$$
   A hospital $h \notin \mathcal{H}_{lq}$ in G has exactly one copy h^0 in G' and we call it the level-0 copy of the hospital. The capacity of the level-0 copy in G' is $q(h^0) = q^+(h)$. We denote this set of copies of the hospitals as $\mathcal{H}'_c$ and call them *true* hospitals.
3. **Dummy Hospitals:** For every $r \in \mathcal{R}_{lq}$ we have μ_R many dummy hospitals. The role of dummy hospitals is to ensure that in a stable matching in G' at most, one true hospital is matched across several copies of a lower-quota resident. We denote the set of dummy hospitals corresponding to $r \in \mathcal{R}_{lq}$ as $\mathcal{D}_r$ which is defined as:
   $$\mathcal{D}_r = \{ \, d_r^y \quad | \quad 0 \le y < \mu_R \, \}$$
4. **Dummy Residents:** For every $h \in \mathcal{H}_{lq}$ we have μ_H sets of dummy residents. As with dummy hospitals, the role of the dummy residents is to ensure that in any stable matching in G', at most $q^+(h)$ many true residents are matched across several copies of the lower-quota hospital. We denote the level-x set of dummy residents corresponding to a lower-quota hospital $h \in \mathcal{H}_{lq}$ as $\mathcal{D}_h^x$. The set is defined as follows:
   $$\mathcal{D}_h^x = \begin{cases} \{d_{h,1}^x, d_{h,2}^x, \ldots, d_{h,q^+(h)}^x\} & \text{for } x = 0 \\ \{d_{h,1}^x, d_{h,2}^x, \ldots, d_{h,q^-(h)}^x\} & \text{for } 1 \le x < \mu_H \end{cases}$$

   We are now ready to define our resident set $\mathcal{R}'$ and hospital set $\mathcal{H}'$ in G'.
   $$\mathcal{R}' = \mathcal{R}'_c \cup \bigcup_{\substack{h \in \mathcal{H}_{lq} \\ 0 \le x \le \mu_H - 1}} \mathcal{D}_h^x \qquad\qquad \mathcal{H}' = \mathcal{H}'_c \cup \bigcup_{r \in \mathcal{R}_{lq}} \mathcal{D}_r$$

We now define our preference lists for the residents and the hospitals in G'. For any vertex $v \in \mathcal{R} \cup \mathcal{H}$, let $\langle list_v \rangle$ denote the preference list of v in G. Let $\langle lqlist_v \rangle$ denote the preference list of v restricted to the lower-quota vertices in its preference list, where the relative ordering of the lower-quota vertices is preserved. Finally, for a particular level t, we denote by $\langle lqlist_v\rangle^t$ as the list of level-t copies of the lower-quota vertices in the preference list of v. For example in G if a resident r has its preference list as h_1, h_2, h_3, h_4 where h_2 and h_4 belong to $\mathcal{H}_{lq}$, then $\langle list_r\rangle = h_1, h_2, h_3, h_4$ and $\langle lqlist_r\rangle = h_2, h_4$. Furthermore say $t = 3$, then $\langle lqlist_r\rangle^3 = h_2^3, h_4^3$. We let the symbol $\circ$ denotes the concatenation of two preference lists.

1. **Preferences of true residents:** For a resident $r \notin \mathcal{R}_{lq}$, we have exactly one copy r^0 in G' whose preference list is obtained by concatenating the r's highest level lower-quota hospitals, followed by the next highest level lower-quota hospitals and so on finally followed by *all* level-0 hospitals in its preference list. Formally, the list for r^0 is defined below.
   $$r^0 \;:\; \langle lqlist_r\rangle^{\mu_H} \circ \langle lqlist_r\rangle^{\mu_H-1} \circ \ldots \circ \langle lqlist_r\rangle^1 \circ \langle list_r\rangle^0$$
   Recall that, for a resident $r \in \mathcal{R}_{lq}$ we have $\mu_R + 1$ many copies of r in G'. Broadly, the preference list of these copies are obtained by prefixing and suffixing dummies to the preference list of r^0 shown above. Hence we find it convenient to use the notation $\langle clonedlist_r\rangle$ to denote the following:
   $$\langle clonedlist_r\rangle = \langle lqlist_r\rangle^{\mu_H} \circ \langle lqlist_r\rangle^{\mu_H-1} \circ \ldots \circ \langle lqlist_r\rangle^1 \circ \langle list_r\rangle^0$$
   The preference lists of the $\mu_R + 1$ copies of r can be defined using the $\langle clonedlist_r\rangle$ as given below.
   $$\begin{array}{lcl} r^0 & : & \langle clonedlist_r\rangle \circ d_r^0 \\ r^1 & : & d_r^0 \circ \langle clonedlist_r\rangle \circ d_r^1 \\ & & \vdots \\ r^{\mu_R-1} & : & d_r^{\mu_R-2} \circ \langle clonedlist_r\rangle \circ d_r^{\mu_R-1} \\ r^{\mu_R} & : & d_r^{\mu_R-1} \circ \langle clonedlist_r\rangle \end{array}$$

2. **Preferences of true hospitals:** For a hospital $h \notin \mathcal{H}_{lq}$, we have exactly one copy h^0 in G' whose preference list is obtained by concatenating the hospital's highest level lower-quota residents, followed by the next highest level lower-quota residents and so on finally followed by *all* level-0 residents in its preference list. Formally, the list for h^0 is defined below.
   $$h^0 \;:\; \langle lqlist_h\rangle^{\mu_R} \circ \langle lqlist_h\rangle^{\mu_R-1} \circ \ldots \circ \langle lqlist_h\rangle^1 \circ \langle list_h\rangle^0$$
   As with lower-quota residents, we have $\mu_H + 1$ many copies of a lower-quota hospitals in G'. We will use the notation $\langle clonedlist_h\rangle$ to denote the following:
   $$\langle clonedlist_h\rangle = \langle lqlist_h\rangle^{\mu_R} \circ \langle lqlist_h\rangle^{\mu_R-1} \circ \ldots \circ \langle lqlist_h\rangle^1 \circ \langle list_h\rangle^0$$
   The preference lists of the $\mu_H + 1$ copies of h can be defined using the $\langle clonedlist_h\rangle$ as given below. Note that for a level-y copy h^y of h we have $q(h^y)$ many leading dummies, followed by the *cloned list* of h followed by $q(h^y)$ many trailing dummies. The leading dummies for h^y are the trailing dummies for h^{y-1} and the the trailing dummies for h^y are leading dummies for h^{y+1}. Recall that $q(h^0) = q^+(h)$ and $q(h^1) = q^-(h)$ and hence $q^-(h)$ trailing dummies of h^0 are the leading dummies of h^1. Thus, k in h^1's preference list denote the value $k = q^+(h) - q^-(h) + 1$.
   $$\begin{array}{lcl} h^0 & : & \langle clonedlist_h\rangle \circ d_{h,1}^0, \ldots, d_{h,q^+(h)}^0 \\ h^1 & : & d_{h,k}^0, \ldots, d_{h,q^+(h)}^0 \circ \langle clonedlist_h\rangle \circ d_{h,1}^1, \ldots, d_{h,q^-(h)}^1 \\ h^2 & : & d_{h,1}^1, \ldots, d_{h,q^-(h)}^1 \circ \langle clonedlist_h\rangle \circ d_{h,1}^2, \ldots, d_{h,q^-(h)}^2 \\ & & \vdots \\ h^{\mu_H-1} & : & d_{h,1}^{\mu_H-2}, \ldots, d_{h,q^-(h)}^{\mu_H-2} \circ \langle clonedlist_h\rangle \circ d_{h,1}^{\mu_H-1}, \ldots, d_{h,q^-(h)}^{\mu_H-1} \\ h^{\mu_H} & : & d_{h,1}^{\mu_H-1}, \ldots, d_{h,q^-(h)}^{\mu_H-1} \circ \langle clonedlist_h\rangle \end{array}$$

3. **Preferences of dummy hospitals:** All dummy hospitals have a preference list of length two. The preference list of a dummy hospital d_r^y for a lower quota resident r is:
   $d_r^y : r^y, r^{y+1}$
4. **Preferences of dummy residents:** We have several sets of dummy residents corresponding to a lower-quota hospital $h \in \mathcal{H}_{lq}$. The dummy residents, except for the first $k-1$ many level-0 dummy residents ($k = q^+(h) - q^-(h) + 1$) have a preference list of length two. The preference lists of the dummy residents is given below.

$$\begin{aligned} d_{h,i}^x \quad &: \quad h^0 && x = 0, \quad i \in \{1, 2, \ldots, k-1\} \\ &: \quad h^0, h^1 && x = 0, \quad i \in \{k, \ldots, q^+(h)\} \\ &: \quad h^x, h^{x+1} && x \in \{1, 2, \ldots, \mu_H - 1\} \end{aligned}$$

   The preference lists of dummy residents and dummy hospitals ensures that a stable matching in G' naturally maps to a popular matching in the original instance G.

This completes the description of our reduced instance G'. We illustrate this reduction in Appendix A.1 using an example, where we convert the HR2LQ instance of Figure 1 to an HR instance. In the next section, we describe the outline of our algorithm and show that if the given HR2LQ instance admits a feasible matching, then the output of our algorithm is a feasible matching.

3 Our algorithm and its feasibility

Given our reduction from an HR2LQ instance G to an HR instance G', our algorithm to compute a popular matching M is straightforward. We simply run the standard Gale-Shapley algorithm on G' and compute a stable matching M_s. We will first prove some useful properties of the stable matching M_s which allows a natural way to obtain a matching M in G.

We call a vertex *under-subscribed* in a matching M if $|M(v)| < q^+(v)$. Note that for residents, under-subscribed is the same as unmatched.

▶ **Definition 5** (Active vertex). *A hospital h^y is* active *in M_s if h^y is matched to at least one true resident in M_s. Otherwise, we call h^y inactive in which case it is matched to all dummy residents. A resident r^x is* active *in M_s if r^x is matched to a true hospital in M_s, else r^x is inactive.*

▶ **Definition 6** (True edges). *An edge $e \in E'$ is called a* true *edge if both the end points are true vertices that is $e = (r^x, h^y)$ where $r^x \in \mathcal{R}'_c$ and $h^y \in \mathcal{H}'_c$.*

Next, we describe some crucial properties of a stable matching M_s in the HR instance G'. Our reduction, more specifically the placement of the set of dummy residents in preference lists, ensures that a hospital h is not matched to more than $q^+(h)$ many true residents across all the level copies $h^0, \ldots, h^{\mu_H}$ in M_s. As M_s is a stable matching, at most two *consecutive* level copies h^y and h^{y+1} of h can be matched to true residents in M_s, otherwise there exists a blocking pair w.r.t. M_s. All lower-level copies (less than y) are completely matched to the respective trailing dummy residents, and all higher-level copies (greater than $y+1$) are completely matched to the respective leading dummy residents. Moreover, only the highest level copy of a hospital can remain under-subscribed, and if it happens then, none of its lower-level copies is matched to a true resident. Similar properties hold for resident copies as well. That is, at most, one *level copy* of a resident is matched to a true hospital, and all other *level copies* are matched to dummy hospitals in any stable matching. Moreover, if a level-x copy of a resident is matched to some true hospital, then all its lower-level copies are matched to the respective trailing dummy hospital, and the higher level copies are matched to the respective leading dummy hospital. We formally list these properties of a stable matching of G' in Lemma 7. The proof appears in Appendix A.2.

▶ **Lemma 7.** *The stable matching M_s in G' satisfies the following properties:*

1. *For any resident $r \in \mathcal{R}$, M_s matches at most one true hospital across all the level copies of r in G'. For any $h \in \mathcal{H}$, M_s matches at most $q^+(h)$ true residents across all the level copies of h in G'.*
2. *The matching M_s leaves only the highest level copy of the vertex (resident or hospital) under-subscribed. In case of hospitals, for $h \in \mathcal{H}_{lq}$, this implies that only h^{μ_H} is possibly under-subscribed in M_s. For a non lower-quota hospital h, the level-0 copy which is the highest level copy may be under-subscribed in M_s. Similar claims hold true for residents.*
3. *If r^x is active in M_s then,*
   - **a.** *Every resident r^i where $0 \le i \le x-1$ is inactive in M_s and matched to its trailing dummy hospital d_r^i.*
   - **b.** *Every resident r^i where $x+1 \le i \le \mu_R$ is inactive in M_s and matched to its leading dummy hospital d_r^{i-1}.*
4. *If h^y is active in M_s then*
   - **a.** *The hospital h^{y-1} must be matched to at least one dummy resident among its trailing dummies, that is, in the set $\mathcal{D}_h^{y-1}$.*
   - **b.** *Every hospital h^j where $0 \le j \le y-2$ is inactive in M_s and fully-subscribed with its trailing dummies, that is residents in $\mathcal{D}_h^j$*
   - **c.** *Every hospital h^j where $y+2 \le j \le \mu_H$ is inactive in M_s and fully-subscribed with its leading dummies, that is, residents in $\mathcal{D}_h^{j-1}$.*
5. *For any resident at most one of its level copy is active in M_s. For any hospital h at most two consecutive level copies are active in M_s.*
6. *If a level y, $y > 0$ copy of a hospital h is active in M_s, then h is matched to at most $q^-(h)$ true residents in M_s. If the highest level copy h^{μ_H} of a hospital h is under-subscribed in M_s then none of its level-j copies for $j < \mu_H$ are active in M_s.*

Lemma 8 below states that a stable matching in G' cannot contain an edge whose both the endpoints are *active* at *higher* levels. This allows us to define a simple *map function* that maps the stable matching M_s in G' to a feasible popular matching M in G.

▶ **Lemma 8.** *For every true edge (r^x, h^y) in G', if r^x and h^y are active in M_s, then at least one of x and y must be 0.*

Proof. For the sake of contradiction, let us assume that there is an edge (r^x, h^y) such that $x, y > 0$ and both are active in M_s. Since, r^x is matched to a true hospital, r^{x-1} must get matched to its last dummy hospital d_r^{x-1} by Part 3a in Lemma 7. So, r^{x-1} prefers h^{y-1} over $M_s(r^{x-1}) = d_r^{x-1}$. Similarly, since h^y is active in M_s, h^{y-1} must be matched to at least one of the trailing dummies in M_s say, $d \in \mathcal{D}_h^{y-1}$. Thus, h^{y-1} prefers r^{x-1} over one of its matched partner d. Hence, (r^{x-1}, h^{y-1}) is a blocking pair w.r.t. a stable matching M_s. ◀

From the above discussion, a *mapping function* that maps M_s of G' to M in G is straight forward. For each edge (r^x, h^0) or (r^0, h^y) in M_s, we include the edge (r, h) in M. We prove that the mapping M is feasible for the original HR2LQ instance G by combining the two claims given in Lemma 9. We give the proofs of these two feasibility claims in Appendix A.2.

▶ **Lemma 9.** *Let M be the map of the stable matching M_s in G', then following holds true:*

1. *If G admits a resident-feasible matching, then M is resident-feasible in G.*
2. *If G admits a hospital-feasible matching, then M is hospital-feasible for G.*

4 Popularity of our matching

Given a feasible matching N in the HR2LQ instance $G = (\mathcal{R} \cup \mathcal{H}, E)$ we construct a weighted bipartite graph $\tilde{G}_N$ along with a matching N^*. The weight of an edge in $\tilde{G}_N$ is the sum of the votes by the end vertices of that edge when compared to the matching N, and the matching N^* is a one-to-one matching corresponding to N. The construction of the matching N^* is in such a way that it matches all the *clones* of residents and hospitals in $\mathcal{R} \cup \mathcal{H}$. We call N^* an $(\mathcal{R} \cup \tilde{\mathcal{H}})$- *perfect matching* (where $\tilde{\mathcal{H}}$ is set of all the clones of hospitals). In other words, the weight of an edge represents the sum of the votes by end vertices and the $(\mathcal{R} \cup \tilde{\mathcal{H}})$-perfect matching N^* has weight 0. Hence, to prove that N is popular, it suffices to show that the maximum weight $(\mathcal{R} \cup \tilde{\mathcal{H}})$-perfect matching in $\tilde{G}_N$ has weight at most zero. For this, we write a maximum weight matching LP for $\tilde{G}_N$ and exhibit a feasible dual assignment with value zero. This is inspired from [3]; however, as mentioned earlier, our dual assignment is considerably involved given that we have multiple levels for both residents and hospitals.

4.1 The graph $\tilde{G}_N$ corresponding to a feasible matching N

Now we describe the construction of the weighted graph $\tilde{G}_N$ corresponding to any feasible matching N in G, and the one-to-one matching N^* using the matching N.

1. **Vertex set of $\tilde{G}_N$:** The graph $\tilde{G}_N$ has the vertex set as $\mathcal{R} \cup \tilde{\mathcal{H}} \cup \tilde{\mathcal{L}}$. The set $\mathcal{R}$ denotes the same set of residents as in G. The set $\tilde{\mathcal{H}}$ denotes *clones* of the original hospitals where every hospital in $\mathcal{H}$ has upper quota many clones. That is,
   $$\tilde{\mathcal{H}} = \{\ h_j \ \mid\ h \in \mathcal{H} \ \text{ and } 1 \leq j \leq q^+(h)\}$$
   every $h \in \mathcal{H}$ has $q^+(h)$ many clones in $\tilde{\mathcal{H}}$. Having these upper quota many clones allows us to convert the many-to-one matching N to a one-to-one matching N^*. Finally, the set $\tilde{\mathcal{L}} = \tilde{\mathcal{L}_r} \cup \tilde{\mathcal{L}_h}$ denotes the set of (dummy) last-resort vertices. For every non lower-quota resident r we have a last resort hospital $\ell_r \in \tilde{\mathcal{L}_r}$. For each vertex $h \in \mathcal{H}$ let $d(h) = q^+(h) - q^-(h)$ denote the difference between the upper quota and lower quota of h. Corresponding to h we have $d(h)$ many last-resort residents in $\tilde{\mathcal{L}_h}$. That is,
   $$\tilde{\mathcal{L}_r} = \{\ \ell_r \ \mid\ r \in \mathcal{R} \text{ and } q^-(r) = 0\}$$
   $$\tilde{\mathcal{L}_h} = \{\ \ell_{h_k} \ \mid\ h \in \mathcal{H} \ \text{ and } 1 \leq k \leq d(h)\}$$
   Thus a lower-quota resident and a hospital with a lower quota equal to its upper quota do not have any last-resort vertices corresponding to it. These last-resort vertices are used to convert N to an $(\mathcal{R} \cup \tilde{\mathcal{H}})$-perfect matching in $\tilde{G}_N$. We call these vertices last-resorts to avoid confusion with the dummies used in the reduction in Section 2.
2. **The matching N^*:** Given the feasible (many-to-one) matching N, we construct an $(\mathcal{R} \cup \tilde{\mathcal{H}})$-perfect one-to-one matching N^*. For every edge $(r, h) \in N$ we select an unselected clone of h say h_j and add the edge (r, h_j) to N^*. For a resident $r \in \mathcal{R}$ which is unmatched in N, we add the edge (r, ℓ_r) to N^*. For any $h \in \mathcal{H}$ which is under-subscribed in N, that is $|N(h)| < q^+(h)$, for every unmatched clone of h, say h_j we match it to a unique last-resort say ℓ_{h_j} and hence add the edge (h_j, ℓ_{h_j}) to N^*. Thus our matching N^* is a one-to-one and $(\mathcal{R} \cup \tilde{\mathcal{H}})$-perfect matching.
3. **The unmatched edges E_U in $\tilde{G}_N$:** For every edge $(r, h) \in E \setminus N$, we add $q^+(h)$ many edges to E_U. That is, we add to the edge set the edges (r, h_j) for every clone h_j of h. We also have unmatched edges from clones of hospitals to the last-resorts corresponding to the hospitals. We have two cases depending on whether $|N(h)| > q^-(h)$ or $|N(h)| = q^-(h)$. This construction is important for our dual feasible setting in the next section.

- For a hospital h where $|N(h)| > q^-(h)$ we have a complete bipartite graph between the $q^+(h)$ many clones of h and all the last-resort vertices corresponding to h. Recall that we have $d(h) = q^+(h) - q^-(h)$ many last resorts corresponding to h. Thus we add to E_U edges of the form (ℓ_{h_k}, h_j) where $1 \le k \le d(h)$ and $1 \le j \le q^+(h)$.
- For a hospital h where $|N(h)| = q^-(h)$, we have a complete bipartite graph between the set of clones of h matched to last-resort vertices and all the last resort vertices corresponding to h. Thus we add to E_U edges of the form (ℓ_{h_k}, h_j) where $1 \le k \le d(h)$ and h_j is matched to a last-resort in the above construction.

4. **The edge set $\tilde{E}$ and their weights:** The edge set $\tilde{E} = N^* \cup E_U$. Every edge of N^* is assigned a weight 0 and every edge (r, h_j) of E_U is assigned a weight $=$ $vote_r(h, N^*(r)) + vote_h(r, N^*(h_j))$ where $r \in \mathcal{R}$ and $h_j \in \tilde{\mathcal{H}}$. Every edge of the form (r, ℓ_r) of E_U is assigned a weight $= vote_r(\ell_r, N^*(r))$ where $r \in \mathcal{R}$ and $\ell_r \in \tilde{\mathcal{L}}_r$. Similarly, every edge of the form (h_j, ℓ_{h_k}) of E_U is assigned a weight $= vote_h(\ell_{h_k}, N^*(h_j))$ where $h \in \tilde{\mathcal{H}}$ and $\ell_{h_k} \in \tilde{\mathcal{L}}_h$.

This completes the description of the weighted bipartite graph $\tilde{G}_N$. Now we use Theorem 10, which gives the sufficient condition for the matching N to be popular, to prove the popularity of the matching M computed by our algorithm. We will construct the matching M^* and the graph $\tilde{G}_M$ corresponding to the matching M. Our goal is to show that *every* $(\mathcal{R} \cup \tilde{\mathcal{H}})$-perfect matching in $\tilde{G}_M$ has weight at most 0.

▶ **Theorem 10.** *Let N be a feasible matching in G such that every $(\mathcal{R} \cup \tilde{\mathcal{H}})$-perfect matching in $\tilde{G}_N$ has weight at most 0 then N is popular.*

Proof. For any feasible matching T in G, we show a corresponding matching T^* in $\tilde{G}_N$ such that T^* is an $(\mathcal{R} \cup \tilde{\mathcal{H}})$-perfect matching and $wt(T^*) = \Delta(T, N, \mathbf{corr})$, where $wt(T^*)$ denotes the sum of the weights of the edges in T^*. We construct T^* as described next. We find appropriate index $j \in \{1, ..., q^+(h)\}$ corresponding to each edge $(r, h) \in T$, where $(r, h_j) \in \tilde{E} \cap T^*$. For an unmatched resident r and an under-subscribed hospital h in T, we add (r, ℓ_r) and (h_k, ℓ_{h_j}) edges in T^* to make T^* an $(\mathcal{R} \cup \tilde{\mathcal{H}})$-perfect matching.

(i) For each edge $e = (r, h) \in N \cap T$: if $(r, h_j) \in N^*$ then we add the edge (r, h_j) to T^*.

(ii) For every edge $(r, h) \in T \setminus N$, we need to decide the index j such that $(r, h_j) \in T^*$. While evaluating the votes, h uses the correspondence function $\mathbf{corr}_h$. (a) If $\mathbf{corr}_h(r, T, N) = r'$ then the matching N^* must contain an edge (r', h_j) for some j. We include the edge (r, h_j) in T^*. (b) If $\mathbf{corr}_h(r, T, N) = \perp$ then we include (r, h_j) in T^* for some j such that h_j is unmatched so far in T^* and is not adjacent to any of the corresponding last resorts. If there is no such h_j then we arbitrarily choose a clone h_i such that $(h_i, \ell_{h_i}) \in N^*$ and include (h_i, ℓ_{h_i}) in T^*.

(iii) For any vertex $v_k \in \mathcal{R} \cup \tilde{\mathcal{H}}$ that is left unmatched in the above step, we select an arbitrary but distinct j for $1 \le j \le q^+(v) - q^-(v)$ and include the edge (v_k, ℓ_{v_k}) in T^*.

Since T is a feasible matching in G, all the clones of every hospital $h \in \mathcal{H}$ which are not adjacent to last resort vertices must get matched to a resident $r \in \mathcal{R}$ in T^*. Other clones of h get matched to either a resident or to the last resort. The graph $\tilde{G}_N$ contains exactly $q^+(v) - q^-(v)$ many last resorts for each vertex v and hence all the copies of v which are not matched to a true vertex in step (i) or (ii) above must get matched to one of these last resorts in step (iii). So, all the vertices in $\mathcal{R} \cup \tilde{\mathcal{H}}$ are matched in T^*. It is also easy to see that T^* is a one-to-one matching in $\tilde{G}_N$.

Next, we compute the weight of T^* and show that it is $\Delta(T, N, \mathbf{corr})$. Since every $(\mathcal{R} \cup \tilde{\mathcal{H}})$-perfect matching in $\tilde{G}_N$ has weight at most 0, $\Delta(T, N, \mathbf{corr}) = wt(T^*) \le 0$. This says that there is no feasible matching T which is more popular than N. Hence N is popular matching among all the feasible matchings.

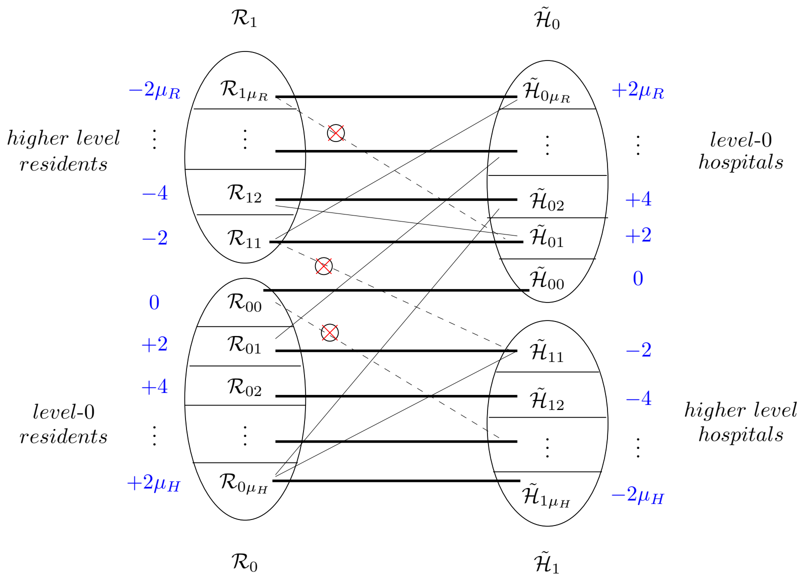

Figure 2 The graph $\tilde{G}_M$ corresponding to our feasible matching M. The bold edges represent the edges in M^*. The values outside the ellipse denote the dual setting and are useful in Section 4.4.

$$
\begin{aligned}
wt(T^*) = \sum_{e \in T^*} wt(e) &= \sum_{(r,h_j)\in T^*} wt(r,h_j) + \sum_{(v_k,\ell_{v_k})\in T^*} wt(v_k,\ell_{v_k}) \\
&= \sum_{(r,h_j)\in T^*} (vote_r(T^*(r), N^*(r)) + vote_{h_j}(T^*(h_j), N^*(h_j))) \\
&\quad + \sum_{(v_k,\ell_{v_k})\in T^*} vote_{v_k}(\ell_{v_k}, N^*(v_k)) \\
&= \sum_{r\in\mathcal{R}} vote_r(T^*(r), N^*(r)) + \sum_{h\in\mathcal{H}} \sum_{i=1}^{q^+(h)} vote_h(T^*(h_i), N^*(h_i)) \\
&= \sum_{r\in\mathcal{R}} vote_r(T, N) + \sum_{h\in\mathcal{H}} vote_h(T, N, \mathbf{corr}_h) \\
&= \Delta(T, N, \mathbf{corr})
\end{aligned}
$$

Thus it follows that $\Delta(T, N, \mathbf{corr}) = wt(T^*)$ which is at most 0 and hence N is popular. ◀

4.2 The graph $\tilde{G}_M$ corresponding to M obtained from M_s

For the HR2LQ instance G, consider the feasible matching M obtained from the stable matching M_s in the reduced HR instance G'. We now use the construction described in Section 4.1 to obtain the graph $\tilde{G}_M$ and the one-to-one $(\mathcal{R} \cup \tilde{\mathcal{H}})$-perfect matching M^*. Since M was obtained from the stable matching M_s in G', an edge $(r, h) \in M$ corresponds to an edge $(r^x, h^y) \in M_s$ where r^x and h^y are level-x and level-y copies of the respective vertices. Further, by Lemma 8 we know that at least one of x or y is zero. We use this property crucially to partition the vertex set of $\tilde{G}_M$. We partition the vertices as described below. Figure 2 shows the high level partition of the vertex set as $\mathcal{R}_0 \cup \mathcal{R}_1 \cup \tilde{\mathcal{H}}_0 \cup \tilde{\mathcal{H}}_1$. Each partition is further refined, for instance $\mathcal{R}_0$ is partitioned as $\mathcal{R}_{00} \cup \mathcal{R}_{01} \cup \ldots \cup \mathcal{R}_{0\mu_H}$. Recall the vertex set of $\tilde{G}_M$ which is given by $\mathcal{R} \cup \tilde{\mathcal{H}} \cup \tilde{\mathcal{L}}_h \cup \tilde{\mathcal{L}}_r$. We partition the residents (including last-resort residents) $\mathcal{R} \cup \tilde{\mathcal{L}}_h$ as $\mathcal{R}_0 \cup \mathcal{R}_1$ and the hospitals (including last-resort hospitals) $\tilde{\mathcal{H}} \cup \tilde{\mathcal{L}}_r$ as $\tilde{\mathcal{H}}_0 \cup \tilde{\mathcal{H}}_1$. Note that the edges of M^* are obtained from the edges of M and the matching M is obtained from the stable matching M_s in G'. We use the edges of M_s, in particular, the levels of the end points of the matched edges, to partition the vertices of $\tilde{G}_M$. The vertices of $\tilde{G}_M$ includes the upper quota many clones for every hospital and the last-resort vertices.

Partition of vertices of $\tilde{G}_M$. Here, we define the sets $\mathcal{R}_{0x}, \mathcal{R}_{1x}, \mathcal{H}_{0y}, \mathcal{H}_{1y}$ based on the edges of M_s.

- Let (r^x, h^y) be an edge in M_s. We consider three cases based on the values of x and y. Note that the case $x > 0, y > 0$ does not arise due to Lemma 8.
  1. If $x = 0$ and $y > 0$, then add r to $\mathcal{R}_{0y}$ and add $M^*(r)$ to $\tilde{\mathcal{H}}_{1y}$. We would like to emphasize that we are using $M^*(r)$ and *not* $M_s(r)$. Since M^* is a one-to-one $(\mathcal{R} \cup \tilde{\mathcal{H}})$-perfect matching, $M^*(v)$ is a uniquely defined vertex of $\tilde{G}_M$ for any vertex v of $\tilde{G}_M$.
  2. If $x > 0$ and $y = 0$ then add r to $\mathcal{R}_{1x}$ and add $M^*(r)$ to $\tilde{\mathcal{H}}_{0x}$.
  3. If $x = 0$ and $y = 0$ then add r to $\mathcal{R}_{00}$ and add $M^*(r)$ to $\tilde{\mathcal{H}}_{00}$.
- For any resident $r \in \mathcal{R}$ that is unmatched in M_s, we add r to $\mathcal{R}_{00}$ and $M^*(r)$ to $\mathcal{H}_{00}$. Note that $M^*(r)$ is a last-resort hospital.
- For any hospital $h \in \mathcal{H}$ that is under-subscribed in M_s, let h_j be a clone of h which is matched to a last-resort in M^*. We add h_j to $\mathcal{H}_{00}$ and $M^*(h_j)$ to $\mathcal{R}_{00}$.
- Finally, any last-resort resident not yet added to any partition is added to $\mathcal{R}_{00}$. Similarly any last-resort hospital not yet added to any partition is added to $\mathcal{H}_{00}$.

We note that the set $\mathcal{R}_0 = \bigcup_{x=1}^{\mu_H} \mathcal{R}_{0x}$. The sets $\mathcal{R}_1, \mathcal{H}_0, \mathcal{H}_1$ are defined similarly. Figure 2 shows the graph $\tilde{G}_M$. It is convenient to have the sets $\mathcal{R}_0$ and $\mathcal{H}_1$ drawn on the lower part and the sets $\mathcal{R}_1$ and $\mathcal{H}_0$ drawn in the upper part. Furthermore, inside $\mathcal{R}_0$ we have the sets $\mathcal{R}_{01}, \mathcal{R}_{02}, \ldots, \mathcal{R}_{0\mu_H}$ arranged from top to bottom. Similarly, we arrange the sets inside $\mathcal{R}_1, \mathcal{H}_0, \mathcal{H}_1$ as shown in Figure 2. We now state the properties of the edges of the graph viewed via the lens of the partition of the vertices in Lemma 11 and Lemma 12. For an edge (u, v) where $u \in \mathcal{R}_{ax}$ and $v \in \tilde{\mathcal{H}}_{by}$ we say the edge is of the form $\mathcal{R}_{ax} \times \tilde{\mathcal{H}}_{by}$.

▶ **Lemma 11.** *The graph $\tilde{G}_M$ does not contain an edge of the form:*
1. $\mathcal{R}_1 \times \tilde{\mathcal{H}}_1$. *That is, there is no edge in $\tilde{G}_M$ from the top right set of residents to the lower left set of hospitals.*
2. $\mathcal{R}_{1x} \times \tilde{\mathcal{H}}_{0y}$ *where $y < x - 1$. That is, in the top set of residents and hospitals, there is no steep downward edge in $\tilde{G}_M$.*
3. $\mathcal{R}_{0x} \times \tilde{\mathcal{H}}_{1y}$ *for $y > x + 1$. That is, in the bottom set of residents and hospitals there is no steep downward edge in $\tilde{G}_M$.*

Proof.

- *Proof of 1*: Proof is immediate from Lemma 8. Because, if there exists an edge (r, h) in G such that $r \in \mathcal{R}_1$ and $h_j \in \tilde{\mathcal{H}}_1$ then in G', there must exist an edge (r^x, h^y) with $x, y > 0$. But then the edge (r^x, h^y) blocks M_s as they both prefer each other over some of their matched partners.
- *Proof of 2*: We prove this by contradiction. Suppose there exist an edge $e = (r, h_j)$ such that $r \in \mathcal{R}_{1x}$ and $h_j \in \tilde{\mathcal{H}}_{0y}$ for $y \le x - 2$. This means r^x is matched to some h'^0 and one of the matched partner of h^0 is some resident r'^y. Now, consider the resident r^{x-1}. This resident must be matched to its trailing dummy hospital and hence prefers h^0 over its

matched partner. Any hospital prefers higher level resident over any lower level resident and hence h^0 prefers r^{x-1} over r'^{y}. Thus, (r^{x-1}, h^0) blocks the stable matching M_s. This is a contradiction.

- *Proof of 3*: Proof is the same as the proof of 2 above. ◀

In Figure 2, the edges not present in $\tilde{G}_M$ are the dashed edges marked with a red cross inside a circle. We now state the properties of the weights on the edges of $\tilde{G}_M$. Note that the weight of an edge of $\tilde{G}_M$ denotes the sum of the votes of the end-points when compared to the matching M. Thus for $e \in \tilde{E}$, we have $-2 \leq wt(e) \leq 2$.

▶ **Lemma 12.** *Let $e = (r, h_j)$ be any edge in $\tilde{G}_M$ such that $r \in \mathcal{R}$ and $h_j \in \tilde{\mathcal{H}}$. Then,*

1. *If $e \in \mathcal{R}_{1x} \times \tilde{\mathcal{H}}_{0(x-1)}$ then $wt(e) = -2$.*
2. *If $e \in \mathcal{R}_{1x} \times \tilde{\mathcal{H}}_{0x}$ then $wt(e) \in \{-2, 0\}$*
3. *If $e \in \mathcal{R}_{1x} \times \tilde{\mathcal{H}}_{0y}$ for $y > x$ then $wt(e) \leq 2$.*
4. *If $e \in \mathcal{R}_{0} \times \tilde{\mathcal{H}}_{0}$ then $wt(e) \leq 2$. Moreover, if $e \in \mathcal{R}_{00} \times \tilde{\mathcal{H}}_{00}$ then $wt(e) \leq 0$.*
5. *If $e \in \mathcal{R}_{0x} \times \tilde{\mathcal{H}}_{1y}$ for $y < x$ then $wt(e) \leq 2$.*
6. *If $e \in \mathcal{R}_{0x} \times \tilde{\mathcal{H}}_{1x}$ then $wt(e) = 0$ or -2.*
7. *If $e \in \mathcal{R}_{0x} \times \tilde{\mathcal{H}}_{1(x+1)}$ then $wt(e) = -2$.*

Proof.

- *Proof of 1*: The weight $wt(e)$ for an edge $e = (r, h_j)$ is defined as $wt(r, h_j) = vote_r(h_j, M^*(r)) + vote_{h_j}(r, M^*(h_j))$. So, our goal here is to show that $vote_r(h_j, M^*(r)) = vote_{h_j}(r, M^*(h_j)) = -1$. Assume for the sake of contradiction that $vote_r(h_j, M^*(r)) \neq -1$ and $vote_{h_j}(r, M^*(h_j)) \neq -1$. Hence, we have three other possibilities [(+1,+1), (+1,-1) and (-1,+1)] and all these three are covered in below two cases: (a) $vote_r(h_j, M^*(r)) = 1$ and (b) $vote_r(h_j, M^*(r)) = -1$. Suppose $r \in \mathcal{R}_{1x}$ and $h_j \in \tilde{\mathcal{H}}_{0(x-1)}$. Consider the same edge in the reduced graph G'. This is an edge between r^x and h^0, and h^0 is matched with some resident $r'^{(x-1)}$. First, we show that $vote_r(h_j, M^*(r)) \neq 1$ which implies $vote_r(h_j, M^*(r)) = -1$. If $vote_r(h_j, M^*(r)) = 1$ then (r^x, h^0) is a blocking edge w.r.t. the stable matching M_s in G' because h^0 always prefers higher level resident r^x over the lower level resident $r'^{(x-1)}$, one of its matched partner. Now, as $vote_r(h_j, M^*(r)) = -1$, the only other possibility left is $vote_r(h_j, M^*(r)) = -1$ and $vote_{h_j}(r, M^*(h_j)) = +1$ but then consider the vertex r^{x-1} in G', this vertex must be matched to its trailing dummy and hence prefers h^0 more. As h prefers r over its matched partner r', h^0 must prefer r^{x-1} over $r'^{(x-1)}$. Again, we have a blocking edge (r^{x-1}, h^0) w.r.t. the stable matching M_s in G'. Hence, $wt(e) = -2$.
- *Proof of 2*: If $e \in \mathcal{R}_{1x} \times \tilde{\mathcal{H}}_{0x}$ then $wt(e)$ cannot be $+2$, otherwise the same edge in G' will be a blocking edge w.r.t. M_s. Hence, $wt(e)$ can only be -2 or 0.
- *Proof of 3*: The maximum possible weight for any e is 2.
- *Proof of 4*: The largest possible weight $wt(e)$ is $+2$. If $e \in \mathcal{R}_{00} \times \tilde{\mathcal{H}}_{00}$ and $wt(e) = 2$ then the same edge in G' blocks M_s.

Proofs of 5, 6 and 7 follow from the earlier claims. ◀

4.3 Linear Program and its Dual

Given the weighted graph $\tilde{G}_M$ we use the standard linear program (LP) to compute a maximum weight $(\mathcal{R} \cup \tilde{\mathcal{H}})$-perfect matching in $\tilde{G}_M$. Recall that every edge e has a weight associated with it which denotes the sum of the votes of the endpoints of the edge with respect to the matching M. The LP and its dual (dual-LP) are given below. For the (primal) LP we have a variable x_e for every edge in $\tilde{E}$. We let $\delta(v)$ denote the set of edges incident on the vertex v in the graph $\tilde{G}_M$.

$$\textbf{LP:} \qquad \max \sum_{e \in \tilde{E}} wt(e) \cdot x_e$$

subject to:

$$\begin{aligned}
\sum_{e \in \delta(v)} x_e &= 1 && \forall v \in \mathcal{R} \cup \tilde{\mathcal{H}} \\
\sum_{e \in \delta(\ell_{h_k})} x_e &\leq 1 && \forall \ell_{h_k} \in \tilde{\mathcal{L}}_h \\
x_e &\geq 0 && \forall e \in \tilde{E}
\end{aligned}$$

We obtain the dual of the above LP by associating a variable α_v for every $v \in \mathcal{R} \cup \tilde{\mathcal{H}} \cup \tilde{\mathcal{L}}$.

$$\textbf{dual-LP:} \qquad \min \sum_{r \in \mathcal{R}} \alpha_r + \sum_{h_j \in \tilde{\mathcal{H}}} \alpha_{h_j} + \sum_{\ell_{h_k} \in \tilde{\mathcal{L}}_h} \alpha_{\ell_{h_k}}$$

subject to:

$$\begin{aligned}
\alpha_r + \alpha_{h_j} &\geq wt(r, h_j) && \forall (r, h_j) \in \tilde{E} \text{ where } r \in \mathcal{R},\ h_j \in \tilde{\mathcal{H}} && (1) \\
\alpha_{\ell_{h_k}} + \alpha_{h_j} &\geq wt(\ell_{h_k}, h_j) && \forall (\ell_{h_k}, h_j) \in \tilde{E} \text{ where } \ell_{h_k} \in \tilde{\mathcal{L}}_h, h_j \in \tilde{\mathcal{H}} && (2) \\
\alpha_r &\geq wt(r, \ell_r) && \forall r \in \mathcal{R} \text{ and } q^-(r) = 0 && (3) \\
\alpha_{\ell_{h_k}} &\geq 0 && \forall \ell_{h_k} \in \tilde{\mathcal{L}}_h && (4)
\end{aligned}$$

4.4 Dual Assignment and its correctness

In this section we present an assignment of values to the dual variables of the **dual-LP** and prove that it is feasible as well as the sum of the dual values is zero. The dual assignment is shown in Figure 2 in blue.

- Set $\alpha_r = +2x$ for all $r \in \mathcal{R}_{0x}$ where $0 \leq x \leq \mu_H$.
- Set $\alpha_{h_j} = -2y$ for all $h_j \in \tilde{\mathcal{H}}_{1y}$ where $1 \leq y \leq \mu_H$.
- Set $\alpha_r = -2x$ for all $r \in \mathcal{R}_{1x}$ where $1 \leq x \leq \mu_R$.
- Set $\alpha_{h_j} = +2y$ for all $h_j \in \tilde{\mathcal{H}}_{0q}$ where $0 \leq y \leq \mu_R$.
- Set $\alpha_{\ell_{h_k}} = 0$ for the last resorts corresponding to a hospital $h \in \mathcal{H}$ are 0.

▶ **Lemma 13.** *The above dual assignment is feasible, and the sum of the dual values is zero.*

Proof. We prove that the dual assignment satisfies all (1), (2), (3), (4) of the dual LP. Eq (4) holds because the last resorts corresponding to a hospital are assigned α-values 0. It is clear from the partition of the vertices of $\mathcal{R} \cup \tilde{\mathcal{H}}$ that all the non-lower quota residents are only in $\mathcal{R}_0$. The α-values for all such residents are non-negative. Moreover, the $wt(r, \ell_r)$ for a non lower-quota resident is at most 0. This implies that all the non-lower-quota residents satisfy the Eq (3). Recall that there is no last resort corresponding to a lower-quota resident.

Next, we show that our dual assignment satisfies Eq (2). Since, $wt(\ell_{h_k}, h_j)$ is at most 0, it is sufficient to show that the LHS of the second inequality $\alpha_{\ell_{h_k}} + \alpha_{h_j}$ is at least 0. From the partition of the vertices into subsets, it is easy to see that no *non lower-quota copy* of a hospital is in $\tilde{\mathcal{H}}_1$ and, hence no vertex in $\tilde{\mathcal{H}}_1$ is connected to the corresponding last resorts. This implies that all the copies (of hospitals) which are connected to corresponding last resorts are in $\tilde{\mathcal{H}}_0$ and they are assigned α-value at least 0. Since $\alpha_{\ell_{h_k}} = 0$, the LHS of the second inequality is at least 0.

Now, to show that the first inequality of the dual LP holds true for the above assignments, we use the Lemma 11 and Lemma 12.

- Lemma 11 excludes all the edges which can not be there in $\tilde{G}_M$.
- Lemma 12(1) states that for every edge $e \in \mathcal{R}_{1x} \times \tilde{\mathcal{H}}_{0(x-1)}$, we have $wt(e) = -2$. As per our dual assignment $\alpha_r + \alpha_{h_j} = -2x + 2(x-1) = -2 \geq wt(e)$.
- Lemma 12(2) states that for every edge $e \in \mathcal{R}_{1x} \times \tilde{\mathcal{H}}_{0x}$, we have $wt(e) = -2$ or 0. As per our dual assignment $\alpha_r + \alpha_{h_j} = -2x + 2x = 0 \geq wt(e)$.
- Lemma 12(3) states that for every edge $e \in \mathcal{R}_{1x} \times \tilde{\mathcal{H}}_{0y}$ such that $y > x$, we have $wt(e)$ is at most 2. As per our dual assignment $\alpha_r + \alpha_{h_j} = -2x + 2y \geq 2 \geq wt(e)$.
- Lemma 12(4) states that (a) for every edge $e \in \mathcal{R}_{00} \times \tilde{\mathcal{H}}_{00}$, we have $wt(e) \leq 0$. As per our dual assignment $\alpha_r + \alpha_{h_j} = 0 \geq wt(e)$. (b) for all other edges $e \in \mathcal{R}_0 \times \tilde{\mathcal{H}}_0$, we have $wt(e) \leq 2$. Our dual assignment ensures that $\alpha_r + \alpha_{h_j} \geq 2 \geq wt(e)$.
- Lemma 12(5) states that for every edge $e \in \mathcal{R}_{0x} \times \tilde{\mathcal{H}}_{1y}$ for $y < x$, we have $wt(e) \leq 2$. Our dual assignment ensures that $\alpha_r + \alpha_{h_j} = 2x - 2y \geq 2 \geq wt(e)$.
- Lemma 12(6) states that for every edge $e \in \mathcal{R}_{0x} \times \tilde{\mathcal{H}}_{1x}$, we have $wt(e) \leq 0$. As per our dual assignment $\alpha_r + \alpha_{h_j} = 2x - 2x = 0 \geq wt(e)$.
- Lemma 12(7) states that for every edge $e \in \mathcal{R}_{0x} \times \tilde{\mathcal{H}}_{1(x+1)}$, we have $wt(e) = -2$. In this case also, $\alpha_r + \alpha_{h_j} = 2x - 2(x+1) = -2 \geq wt(e)$.

Hence, all the edges of $\tilde{E}$ satisfy inequality (1). As per our assignment, all the matched edges (r, h_j) has $\alpha_r + \alpha_{h_j} = 0$, and $\alpha_r = 0$ for all the residents r matched to last resorts and $\alpha_{h_j} = 0$ for all the clones h_j of hospital h such that h_j are matched to last resorts. Hence it follows that $\sum_{v \in \mathcal{R} \cup \tilde{\mathcal{H}}} \alpha_v = 0$. ◀

Lemma 13 and the weak duality theorem together implies that the optimal value of the primal LP is at most 0. That is, every matching in $\tilde{G}_M$ that matches all vertices in $\mathcal{R} \cup \tilde{\mathcal{H}}$ has weight at most 0. Thus, by using Theorem 10, we establish the main result of this paper stated in Theorem 4.

5 Discussion

In this paper, we addressed the problem of computing a popular, feasible matching in the many-to-one setting when both sides are having lower quotas. This approach can suitably be modified to compute a maximum size popular feasible matching. We comment on the natural generalizations of our problem as follows:

- **Many-to-many setting with one sided lower-quotas:** We call this setting as the student course allocation problem with lower quotas for courses (SCLQ problem). The simple modification of our reduction presented in the paper works for the SCLQ setting.
- **Many-to-many setting with two-sided lower-quotas:** We denote this as the SC2LQ problem. For the SC2LQ setting, there are non-trivial challenges in extending our reduction. These are posed by the presence of capacities as well as lower quotas on both the sides of the bipartition. We remark that these difficulties do not arise in the SCLQ setting where only one side has lower quotas as well as in [3] there are no lower quotas on either side of the bipartition.

References

1 David J Abraham, Robert W Irving, Telikepalli Kavitha, and Kurt Mehlhorn. Popular matchings. *SIAM Journal on Computing*, 37(4):1030–1045, 2007.

2 Péter Biró, Robert W Irving, and David F Manlove. Popular matchings in the marriage and roommates problems. In *International Conference on Algorithms and Complexity*, pages 97–108. Springer, 2010.

3 Florian Brandl and Telikepalli Kavitha. Two problems in max-size popular matchings. *Algorithmica*, 81(7):2738–2764, 2019.

4 Ágnes Cseh and Telikepalli Kavitha. Popular edges and dominant matchings. *Mathematical Programming*, 172(1):209–229, 2018.

5 Yuri Faenza and Telikepalli Kavitha. Quasi-popular matchings, optimality, and extended formulations. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 325–344. SIAM, 2020.

6 Tamás Fleiner and Naoyuki Kamiyama. A matroid approach to stable matchings with lower quotas. *Math. Oper. Res.*, 41(2):734–744, 2016. doi:10.1287/moor.2015.0751.

7 Peter Gärdenfors. Match making: assignments based on bilateral preferences. *Behavioral Science*, 20(3):166–173, 1975.

8 Mizuki Hirakawa, Yukiko Yamauchi, Shuji Kijima, and Masafumi Yamashita. On the structure of popular matchings in the stable marriage problem-who can join a popular matching. In *the 3rd International Workshop on Matching under Preferences (MATCH-UP)*, 2015.

9 Chien-Chung Huang. Classified stable matching. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010*, pages 1235–1253, 2010. doi:10.1137/1.9781611973075.99.

10 Chien-Chung Huang and Telikepalli Kavitha. Popular matchings in the stable marriage problem. *Information and Computation*, 222:180–194, 2013.

11 Chien-Chung Huang and Telikepalli Kavitha. Popularity, mixed matchings, and self-duality. *Mathematics of Operations Research*, 2021.

12 Telikepalli Kavitha. A size-popularity tradeoff in the stable marriage problem. *SIAM Journal on Computing*, 43(1):52–71, 2014.

13 Telikepalli Kavitha. Popular half-integral matchings. In *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.

14 Telikepalli Kavitha. Popular matchings with one-sided bias. In *47th International Colloquium on Automata, Languages, and Programming (ICALP 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.

15 Telikepalli Kavitha. Matchings, critical nodes, and popular solutions. In *41st IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2021 (To Appear)*, 2021.

16 Telikepalli Kavitha. Maximum Matchings and Popularity. In *48th International Colloquium on Automata, Languages, and Programming (ICALP 2021)*, volume 198, pages 85:1–85:21, 2021. doi:10.4230/LIPIcs.ICALP.2021.85.

17 Zoltán Király. Better and simpler approximation algorithms for the stable marriage problem. *Algorithmica*, 60(1):3–20, 2011.

18 A. M. Krishnapriya, Meghana Nasre, Prajakta Nimbhorkar, and Amit Rawat. How good are popular matchings? In *17th International Symposium on Experimental Algorithms, SEA 2018*, pages 9:1–9:14, 2018. doi:10.4230/LIPIcs.SEA.2018.9.

19 Matthias Mnich and Ildikó Schlotter. Stable matchings with covering constraints: A complete computational trichotomy. *Algorithmica*, 82(5):1136–1188, 2020.

20 Meghana Nasre and Prajakta Nimbhorkar. Popular matchings with lower quotas. In *37th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2017*, pages 44:1–44:15, 2017. doi:10.4230/LIPIcs.FSTTCS.2017.44.

21 Meghana Nasre and Amit Rawat. Popularity in the generalized hospital residents setting. In *International Computer Science Symposium in Russia*, pages 245–259. Springer, 2017.

A Appendix

A.1 Illustration of Reduction Method

Following the reduction given in Section 2 we convert the HR2LQ instance of Figure 1 to an HR instance. The resulting reduced instance is shown in Figure 3. The map of the hospital proposing stable matching in the reduced instance is $M = \{(h_1, r_1), (h_2, r_3), (h_3, r_4), (h_6, r_2), (h_7, r_5)\}$ which is popular. The level structure and dual assignment is shown in Figure 4.

$r_1^0 :\ h_2^3\ h_1^3\ h_2^2\ h_1^2\ h_2^1\ h_1^1\ h_4^0\ h_2^0\ h_1^0$

$r_2^0 :\ h_2^3\ h_6^3\ h_1^3\ h_2^2\ h_6^2\ h_1^2\ h_2^1\ h_6^1\ h_1^1\ h_5^0\ h_2^0\ h_6^0\ h_1^0$

$r_3^0 :\ h_2^3\ h_2^2\ h_2^1\ h_3^0\ h_2^1\ d_{r_3}^0$

$r_3^1 :\ d_{r_3}^0\ h_2^3\ h_2^2\ h_2^1\ h_3^0\ h_2^1\ d_{r_3}^1$

$r_3^2 :\ d_{r_3}^1\ h_2^3\ h_2^2\ h_2^1\ h_3^0\ h_2^1$

$r_4^0 :\ h_3^3\ h_3^2\ h_3^1\ h_3^0\ d_{r_4}^0$

$r_4^1 :\ d_{r_4}^0\ h_3^3\ h_3^2\ h_3^1\ h_3^0\ d_{r_4}^1$

$r_4^2 :\ d_{r_4}^1\ h_3^3\ h_3^2\ h_3^1\ h_3^0$

$r_5^0 :\ h_6^3\ h_6^2\ h_6^1\ h_7^0\ h_6^0\ h_8^0$

$d_{h_{1,1}}^0 :\ h_1^0\ h_1^1$

$d_{h_{1,1}}^1 :\ h_1^1\ h_1^2$

$d_{h_{1,1}}^2 :\ h_1^2\ h_1^3$

$d_{h_{2,1}}^0 :\ h_2^0\ h_2^1$

$d_{h_{2,1}}^1 :\ h_2^1\ h_2^2$

$d_{h_{2,1}}^2 :\ h_2^2\ h_2^3$

$d_{h_{6,1}}^0 :\ h_6^0\ h_6^1$

$d_{h_{6,1}}^1 :\ h_6^1\ h_6^2$

$d_{h_{6,1}}^2 :\ h_6^2\ h_6^3$

(a) Preference List of Residents with Quota=1.

$h_1^0 :\ r_1^0\ r_2^0\ d_{h_{1,1}}^0$

$h_1^1 :\ d_{h_{1,1}}^0\ r_1^0\ r_2^0\ d_{h_{1,1}}^1$

$h_1^2 :\ d_{h_{1,1}}^1\ r_1^0\ r_2^0\ d_{h_{1,1}}^2$

$h_1^3 :\ d_{h_{1,1}}^2\ r_1^0\ r_2^0$

$h_2^0 :\ r_3^2\ r_3^1\ r_1^0\ r_2^0\ r_3^0\ d_{h_{2,1}}^0$

$h_2^1 :\ d_{h_{2,1}}^0\ r_3^2\ r_3^1\ r_1^0\ r_2^0\ r_3^0\ d_{h_{2,1}}^1$

$h_2^2 :\ d_{h_{2,1}}^1\ r_3^2\ r_3^1\ r_1^0\ r_2^0\ r_3^0\ d_{h_{2,1}}^2$

$h_2^3 :\ d_{h_{2,1}}^2\ r_3^2\ r_3^1\ r_1^0\ r_2^0\ r_3^0$

$h_3^0 :\ r_3^2\ r_4^2\ r_3^1\ r_4^1\ r_3^0\ r_4^0$

$h_4^0 :\ r_1^0$

$h_5^0 :\ r_2^0$

$h_6^0 :\ r_2^0\ r_5^0\ d_{h_{6,1}}^0$

$h_6^1 :\ d_{h_{6,1}}^0\ r_2^0\ r_5^0\ d_{h_{6,1}}^1$

$h_6^2 :\ d_{h_{6,1}}^1\ r_2^0\ r_5^0\ d_{h_{6,1}}^2$

$h_6^3 :\ d_{h_{6,1}}^2\ r_2^0\ r_5^0$

$h_7^0 :\ r_5^0$

$h_8^0 :\ r_5^0$

$d_{r_3}^0 :\ r_3^0\ r_3^1$

$d_{r_3}^1 :\ r_3^1\ r_3^2$

$d_{r_4}^0 :\ r_4^0\ r_4^1$

$d_{r_4}^1 :\ r_4^1\ r_4^2$

(b) Preference List of Hospitals with Quota=1.

Figure 3 Reduced HR instance corresponding to the counter example given in Figure 1.

A.2 Proofs from Section 3

Proof of Lemma 7.

- *Proof of 1:* If $r \notin \mathcal{R}_{lq}$ then r has only one copy in G' with quota 1 and the result holds trivially. So without loss of generality let us assume that $r \in \mathcal{R}_{lq}$ and hence G' has $\mu_R + 1$ copies of r. The total number of dummy hospitals corresponding to r is μ_R. Each dummy hospital d_r^i contains only two true residents r^i and r^{i+1}. Also, each d_r^i is the top choice

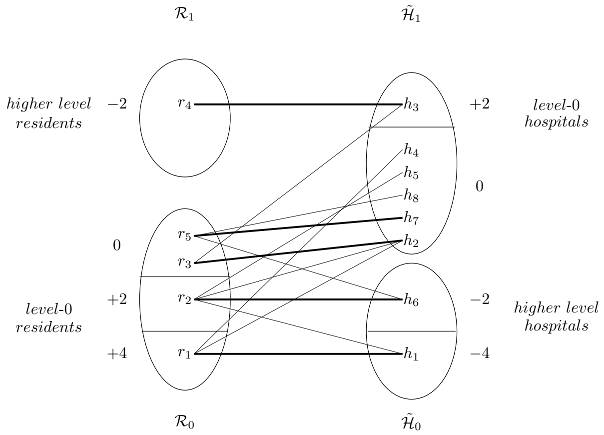

Figure 4 The graph $\tilde{G}_M$ corresponding to the example.

of r^{i+1} for $0 \le i \le \mu_R$. So, none of the dummy hospital can remain unmatched. This implies that at most 1 copy of r can get matched to a true hospital.

If $h \notin \mathcal{H}_{lq}$ then h has only one copy h^0 in G' with quota $q^+(h)$ and the result holds trivially. So without loss of generality, let us assume that $h \in \mathcal{H}_{lq}$ and hence G' has $\mu_H + 1$ copies of h. The total number of dummy residents for h in G' is $\alpha = q^+(h) + q^-(h) \cdot (\mu_H - 1)$, and the total capacity of all the copies of h in G' is $\beta = q^+(h) + q^-(h) \cdot \mu_H$. Consider the set of dummy residents $\mathcal{D}_h^0 \cup \ldots \cup \mathcal{D}_h^{\mu_H - 1}$ corresponding to a lower quota hospital h. For any $y < \mu_H$, except $y = 1$, $\mathcal{D}_h^{y-1}$ are the most preferred $q(h^y)$ dummy residents of h^y. Thus, these dummy residents can never remain unmatched in a stable matching. The dummy residents that can possibly remain unmatched are the subset of $\{d^0_{h,1}, \ldots, d^0_{h,q^+(h)-q^-(h)}\}$ as these are the only dummy residents that are not the top choice of any copy of the hospital h. Hence the number of dummy residents that can remain unmatched in any stable matching of G' is at most $\gamma = q^+(h) - q^-(h)$. This implies that the total number of true residents matched to h in M_s is at most $\beta - \alpha + \gamma = q^+(h)$.

- *Proof of 2:* If $h \notin \mathcal{H}_{lq}$ then it has only one copy h^0 with quota $q^+(h)$. So let us assume that $h \in \mathcal{H}_{lq}$. For each copy h^y, where $y < \mu_H$, there are exactly $q(h^y)$ dummy residents of level-y as their top choice. Thus, h^y cannot remain under-subscribed in any stable matching M_s of G', otherwise these dummy resident(s) form blocking pair(s) with h^y. This implies that only h^{μ_H} can possibly be left under-subscribed in M_s.
If $r \notin \mathcal{R}_{lq}$, then its highest level copy r^0 remains unmatched. So let us assume that $r \in \mathcal{R}_{lq}$. We note that none of the μ_R many dummy hospitals $d_r^0, d_r^1, \ldots, d_r^{\mu_R - 1}$ corresponding to r can be left unmatched in any stable matching. Otherwise , the unmatched dummy hospital d_r^i forms a blocking pair with r^{i+1}. So, at most, one copy of r can potentially be left unmatched. Now, if a copy r^i for $0 \le i \le \mu_R - 1$ is left unmatched then r^i with its last dummy d_r^i forms a blocking pair w.r.t. M_s.
- *Proof of 3:* There are $\mu_R + 1$ copies of a resident $r \in \mathcal{R}_{lq}$ and μ_R dummy hospitals corresponding to it. From the proof of 1 above, we know that none of the dummy hospitals corresponding to r can remain unmatched. The preference list of a dummy hospital d_r^i contains r^i and r^{i+1}, and r^x is active in M_s. This implies that the only possible way to match r^i in a stable matching is to match it with d_r^i, the corresponding trailing dummy, where $0 \le i \le x - 1$. Similarly, the only possible way to match r^i in a stable matching is to match it with d_r^{i+1}, the corresponding leading dummy, where $x + 1 \le i \le \mu_R$.

- *Proof of 4a:* For the sake of contradiction, assume that h^{y-1} is not matched to any resident $d \in D_h^{y-1}$ and still h^y is active. Note that there are exactly $q(h^y)$ many dummy residents in the preference list of h^y from the set $\mathcal{D}_h^{y-1}$. Also, h^y prefers all such dummy residents over any true resident. Each dummy resident from the $(y-1)$-th set, $\mathcal{D}_h^{y-1}$ has only h^{y-1} and h^y in its preference list. It means there is a dummy resident $d_h^j \in D_h^{y-1}$ which is unmatched in M_s. But then (h^y, d_h^j) forms a blocking pair w.r.t. M_s.
  Proof of 4b: If h^y is active and h^j is matched to a true resident r for some $0 \leq j \leq y-2$, then (r, h^{y-1}) is a blocking pair w.r.t. M_s. This is because, as proved above, h^{y-1} must be matched to at least one resident in $\mathcal{D}_h^{y-1}$ and h^{y-1} prefers any true resident over any dummy resident in $\mathcal{D}_h^{y-1}$.
  Proof of 4c: If h^y is active then h^j cannot be active for $y+2 \leq j \leq \mu_H$, otherwise, h^{j-1} must be matched to a resident from $\mathcal{D}_h^{j-1}$. In this case, each true resident which is matched to h^y in M_s forms a blocking pair with h^{j-1}, contradicting the stability of M_s. Now we claim that each such h^j is fully-subscribed with its leading dummies. This is because if h^y is active and h^j is matched to any trailing dummy $d \in \mathcal{D}_h^j$ then a resident $r' \in M_s(h^y)$ forms a blocking pair with h^j.
- *Proof of 5:* The claim for a resident immediately follows from Part 1 above. So, let us prove it for a hospital.
  For the sake of contradiction, let us assume that $h \in \mathcal{R}_{lq}$ is a lower quota hospital such that h^{x_1} and h^{x_2} are active where $x_2 < x_1 - 1$. Also, assume that h^{x_1} and h^{x_2} are matched to r_1 and r_2 respectively. Then, h^{x_1-1} must be matched to at least one dummy residents from $\mathcal{D}_h^{x_1-1}$. But, Then, (r_2, h^{x_1-1}) forms a blocking pair w.r.t. M_s.
- *Proof of 6:* This follows from the fact that for any h, all the dummy residents of all the copies get matched in a stable matching in G', except possibly the $q^+(h) - q^-(h)$ trailing dummies of h^0. This is because all of them are the top choice of some h^i. The other part is true because otherwise a true resident matched to the level-j copy and h^{μ_H} form a blocking pair with respect to M_s. ◀

Proof of Lemma 9. In Lemma 7, we proved that each resident is matched to at most one hospital and each hospital h is matched to at most $q^+(h)$ many residents in G. Here, first we show that each $r \in \mathcal{R}_{lq}$ gets matched to at least one hospital and, then we show that each $h \in \mathcal{H}_{lq}$ gets matched to at least $q^-(h)$ many residents.

Let us assume for the sake of contradiction that M is not resident-feasible, and hence $M(r) = \perp$ for an $r \in \mathcal{R}_{lq}$, but there exists a feasible matching N in G where r is matched. Consider the decomposition of $M \oplus N$ into (possibly non-simple) alternating paths and cycles. The decomposition we use is the same as the one used in [21, 20]. As r is unmatched in M but matched in N there must exist an alternating path ρ in $M \oplus N$ ending at r. Moreover, the highest level copy r^{μ_R} must remain unmatched in M_s by Part 2 of Lemma 7.

Case 1: The other end-point of ρ is a resident r_k: Let $\rho = \langle r, h_1, r_1, h_2, r_2, \ldots, h_k, r_k \rangle$, where $(h_i, r_i) \in M$ and rest of the edges are in N. We show that such a path cannot exist and hence M must be feasible. The length of this path is even and hence r_k remains unmatched in N. It implies that r_k is a non-lower quota resident. Since we do not have multiple copies of a non-lower quota resident r, r_k^0 is matched to a non-dummy copy of a hospital h_k in M_s. Since r^{μ_R} is unmatched in M_s, all the residents $r \in M_s(h_1)$, and hence r_1, must also be the highest level copy. If not, then $(r^{\mu_R}, h_1^0$ blocks M_s because r^{μ_R} is unmatched and any copy of h_1 prefers r^{μ_R} over lower level copy of any resident. The copy of r_2 which is matched to h_2 must be either $r_2^{\mu_R}$ or $r_2^{\mu_R-1}$, otherwise, $(h_2^0, r_1^{\mu_R-1})$ blocks M_s. Continuing in this way, we see that the matched copy of r_3 must be in $\{r_3^{\mu_R}, r_3^{\mu_R-1}, r_3^{\mu_R-2}\}$, and matched copy of r_k must be in $\{r_k^{\mu_R}, r_k^{\mu_R-1}, \ldots, r_k^{\mu_R-(k-1)}\}$.

This implies that the path ρ goes downwards to level 0 but by at most one level for each resident on ρ. Since r_k is the 0^{th} level copy, ρ must contain at least $\mu_R + 1$ residents with non-zero lower quota. But there are only μ_R residents with lower quota 0. So such a path ρ cannot exist.

Case 2: The other end of ρ is a hospital h: Let $\rho = \langle r, h_1, r_1, h_2, r_2, \ldots, h_k, r_k, h\rangle$, where $(h_i, r_i) \in M$ and rest of the edges are in N. It is clear that $|M(h)| < |N(h)| \leq q^+(h)$, that is, h is under-subscribed in M. Consider the first r_p on ρ such that r_p^0 is active in M_s. Such an r_p must exist as proved in Claim 14 below. So, consider the sub-path $\rho' = r, h_1, r_1, \ldots, r_p$ of ρ. Using the same argument as in the previous case, ρ' must contain at least $\mu_R + 1$ lower-quota residents. Therefore ρ', and consequently ρ cannot exist.

▷ Claim 14. An alternating path ρ as considered in Case 2 above contains a resident r_p such that r_p^0 is active in M_s.

Proof of Claim 14. We consider the following two cases: (a) when h is only active at level 0 and, (b) when h is active at higher levels. In the first case, as h is under-subscribed in M_s, r_k^0 must be active in M_s. This is because if r_k^i is active for any $i > 0$ then (r_k^0, h^0) blocks M_s. In the second case, from Lemma 8, r_k^j cannot be active for any $j > 0$. So r_k^0 must be active. ◁

Now, we prove Part 2 of Lemma 9 by contradiction. Let us assume that M is not hospital-feasible but there exists a matching N which is hospital-feasible. That is, there exists $h \in \mathcal{H}_{lq}$ such that $|M(h)| < q^-(h) \leq |N(h)|$. Consider the decomposition of $M \oplus N$ into (possibly non-simple) alternating paths and cycles.

As $|M(h)| < |N(h)|$ there must exists a path ρ in $M \oplus N$ ending at h. Since h is deficient in M, the highest level copy h^{μ_H} must remain under-subscribed in M_s by Part 2 of Lemma 7.

Case 1: The other end of ρ is a hospital h_k: Let $\rho = \langle h, r_1, h_1, \ldots, r_{k-1}, h_{k-1}, r_k, h_k\rangle$, where $(r_i, h_i) \in M$ and rest of the edges are in N. We show that such a path cannot exist and hence M must be feasible. The length of this path ρ is even and $|M(h_k)| > |N(h_k)| \geq q^-(h_k)$. This implies that the higher level copies (level p for $p > 0$) of h_k are not active (Part 4 of Lemma 7). As h^{μ_H} remains under-subscribed in M_s, the copy of hospital h_1 which is matched to r_1 must be $h_1^{\mu_H}$, otherwise, (r_1^0, h^{μ_H}) forms a blocking pair w.r.t. M_s. From Part 4 of Lemma 7, $h_1^{\mu_H - p}$ for $p > 1$ cannot be active in M_s. Similarly, the copies $h_2^{\mu_H - p}$ for $p > 2$ of h_2 cannot be active in M_s. And, the copies $h_k^{\mu_H - p}$ for $p > k$ cannot be active in M_s. In other words, the only active copy of h is h^{μ_H}, the active copies of h_1 are in $\{h_1^{\mu_H}, h_1^{\mu_H - 1}\}$, the active copies of h_2 are in $\{h_2^{\mu_H}, h_2^{\mu_H - 1}, h_2^{\mu_H - 2}\}$ and so on. This implies that we may go downwards in this way but by at most one level for each hospital on ρ. As the only active copy of h_k is h_k^0 and hence, ρ must contain at least $\mu_H + 1$ copies of lower quota hospitals but the sum of all the lower quotas of hospitals is only μ_H. This is a contradiction.

Case 2: Other end of ρ is a resident r: Let $\rho = \langle h, r_1, h_1, r_2, h_2, \ldots, r_k, h_k, r\rangle$, where $(r_i, h_i) \in M$ and rest of the edges are in N. We know that the only active copy of h is h^{μ_H}. Here, r is unmatched in M. If $q^-(r) = 0$ then r^0 remains unmatched in M_s and hence, h_k cannot be active at level above 0. In this case, the same argument as in the previous case suffice to prove that such a path ρ cannot exist. The other case, when $q^-(r) = 1$, is not possible because of Part 1, which says that M is resident feasible but r is unmatched. ◀