Popular Matchings under Lower Quotas and their relationship to Stability under Classifications

by

Ankita Sarkar

A thesis submitted to

Chennai Mathematical Institute

in partial fulfillment

of the requirements for the degree of

Master of Science

Department of Computer Science

2020

This thesis entitled:
Popular Matchings under Lower Quotas and their relationship to Stability under Classifications
written by Ankita Sarkar
has been approved for the Department of Computer Science

____

Dr. Prajakta Nimbhorkar

____

Dr. Meghana Nasre

Date ____________________

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Sarkar, Ankita (M.Sc. in Computer Science)

Popular Matchings under Lower Quotas and their relationship to Stability under Classifications

Thesis supervised by Dr. Prajakta Nimbhorkar

We consider extensions to the classical Stable Marriage and Hospital-Residents problems studied by Gale and Shapley in [1]. We impose the additional constraint of lower quotas, i.e. a mandatory minimum number of partners for some participants. Stable matchings do not always exist under lower quotas, so we explore maximum-cardinality popular matchings. The case with only hospital-sided lower quotas has been studied in [2]. Here, we give a $\mathcal{O}(n^2m)$ algorithm for when lower quotas are imposed on both the hospital and resident sides. We also consider the many-to-many version of Hospital-Residents, called the Students-Courses Problem. Using the notion of classified stability as studied in [3], we give a $\mathcal{O}(m^2)$ algorithm for the case with student-sided lower quotas.

To all teachers, be they faculty or friends.

Acknowledgements

Dr. Prajakta Nimbhorkar has supervised this work with immense patience and constant motivation.
Dr. B. Srivathsan and other faculty advisors have provided the guidance required to make many crucial career decisions correctly. The environment, faculty, and peers in CMI and at off-campus internships have contributed to the knowledge and skills that enable this work.
This work depends on many crucial questions answered, and many more asked, by the related work that precedes this.
The CMI administration has made life on campus very easy, removing any distractions from work.
Family and friends have provided invaluable support and encouragement, especially during the COVID crisis and the resulting weeks of working from home.

Contents

Chapter 1

Introduction

The Marriage Problem is the problem of pairing up men and women to be married, where each person has preferences over his/her prospective partners. This is formalised as a bipartite graph, with the parts being men and women, and each vertex defining a preference order over its neighbours. The goal is to match couples in a manner that ensures some measure of "happiness" among the participants. One such measure can be that no prospective couple has an incentive to defy the matching assigned to them. This property is called *stability*. To clearly define stability, we first define a *blocking pair* with respect to a matching. A man-woman pair is said to *block* a matching when they are not matched to each other, and prefer each other over their assigned partners, i.e.:

- The man is either unmatched or prefers the woman over his assigned woman, and
- The woman is either unmatched or prefers the man over her assigned man.

A *stable matching* is then defined as a matching without blocking pairs. We shall formally define these terms in Section 2.1.

A natural extension of the Marriage Problem is the Hospital Residents Problem (HR), which is a many-to-one version of the Marriage scenario. Here, hospitals want to hire residents, and while each resident will work for a unique hospital, the hospitals have a capacity (or upper quota) of how many residents they wish to hire. In this case, a blocking pair will be a hospital-resident pair not matched to each other, where:

- The hospital either has unfilled positions (is *undersubscribed*) or prefers the resident over at least one of its assigned residents; and
- The resident is either unmatched, or prefers the hospital over its assigned hospital.

A "matching" in HR is no longer a matching in the graph-theoretic sense, which is why it is often referred to by other terms such as an *assignment*. Here, however, we shall continue to call it a matching. Also, we note that the Marriage Problem is simply HR with every hospital having unit upper quota. HR is a model for many situations related to hiring, admissions, allocations, etc.; in fact, HR is also well known as the College Admissions Problem. Multiple stable matchings can exist in HR as well as the Marriage Problem, and they have an interesting property : every stable matching in HR is of the same size, and the number of residents assigned to a hospital is the same in all stable matchings. Moreover, the undersubscribed hospitals match with the same residents in all stable matchings. This is called the *Rural Hospitals Theorem*, and is discussed in [4]. Gale and Shapley, in their famous article, [1], gave efficient algorithms for these stability problems, based on the ideas of provisional engagement and deferred acceptance. These algorithms are surprisingly simple and employ a greedy approach. The Gale-Shapley algorithm for finding a stable matching in HR is as follows :

1. Each hospital proposes to residents, in the order of most to least preferred, until it either fills its upper quota or exhausts the preference list.
2. Each resident provisionally accepts the first proposal it receives. Thereafter, it compares any incoming proposal with its current hospital, and switches to the new hospital if it is more preferred.

The above algorithm has some interesting properties, which Gusfield and Irving have presented in their textbook, [5]. The order of proposals does not affect the output, while the choice of proposing side does. However, whichever side proposes, we obtain a stable matching, and the proposing side is "better off" in some sense. Here, we note that if residents proposed, then each undersubscribed hospital would automatically accept, whereas a full hospital would compare incoming proposals with its least preferred existing resident, and keep the one it prefers.

Beyond Stability

The ideas of the Gale-Shapley algorithm extend to many questions that can be asked about scenarios similar to the above. One such question is about *popular matchings.* A limitation of stability is that stable matchings can be as small as half the size of maximum matchings. An example is the following Marriage instance, with preference lists shown from most to least preferred :

$$m_1 : w_1 \ w_2 \qquad\qquad w_1 : m_1 \ m_2$$
$$m_2 : w_1 \qquad\qquad w_2 : m_1$$

Here, the Gale-Shapley algorithm gives $\{(m_1, w_1)\}$, which by the Rural Hospitals Theorem means that all stable matchings are of this size, while the maximum matching is $\{(m_1, w_2), (m_2, w_1)\}$.

One way to obtain larger matchings while maintaining some notion of "happiness", is to weaken the stability condition to *popularity.* The popularity condition compares two matchings by holding an election between them where all vertices vote, and a popular matching is one that does not lose such an election to any other matching. Existence of popular matchings is not immediate, but stable matchings happen to be popular, so the problem is now to find larger popular matchings, in particular *maximum cardinality popular matchings.* Huang and Kavitha, in [6], solve this problem for the Marriage setting, which has an immediate extension to maximum cardinality popularity in Hospital-Residents. Their solution is via a reduction to stability, and hence gives an efficient algorithm. This is highly useful since, as shown in [7], the size of a maximum cardinality popular matching is at least as large as two-thirds the size of a maximum matching. The algorithm to find maximum cardinality popular matchings uses the idea of repeated proposal with priorities. This idea is as follows for the Marriage Problem :

1. Men propose as in the Gale-Shapley algorithm. During these proposals the men are considered to be at priority 0.
2. Men that exhaust their preference list and remain unmatched, propose again with priority 1. We define women to prefer all priority 1 men over all priority 0 men, with men of the same priority being compared as per the original preferences.

Popularity under Lower Quotas

The notion of popularity is even more useful under additional constraints that can prevent stable matchings from existing. For example, some minimum staff could be required to run each hospital, i.e. we impose a *lower quota* in addition to the capacity (upper quota) of each hospital. We shall call this scenario HRLQ, for "Hospital-Residents with Lower Quotas". By the Rural Hospitals Theorem, if the matching returned by the Gale-Shapley algorithm does not respect lower quotas (is not *feasible*), then we have no hope of finding *feasible stable matchings*. Such instances indeed exist : for example, consider the graph on hospitals $\{h_1, h_2\}$ and residents $\{r_1, r_2\}$. Both the hospitals have an upper quota of 2 and a lower quota of 1. The preference lists, from most to least preferred, are :

$$h_1 : r_1 \ r_2 \qquad\qquad r_1 : h_1 \ h_2$$
$$h_2 : r_1 \ r_2 \qquad\qquad r_2 : h_1 \ h_2$$

h_1 is preferred by both the residents, and wants to hire up to two residents, so both residents end up with h_1. Thus the hospital-proposing Gale-Shapley algorithm yields the matching $\{(h_1, r_1), (h_1, r_2)\}$. Thus the lower quota of h_2 remains unfulfilled.

Given that such instances exist under lower quotas, we instead look to popularity as a measure of "happiness". We look for a matching that is popular among feasible matchings, i.e. a feasible matching that will not lose an election to any other feasible matching. Existence of these is not immediate – however, a constructive algorithm is given by Nasre and Nimbhorkar, in [2], for this question. In fact, they find a *maximum-cardinality* popular matching among feasible matchings. Their algorithm runs in time polynomial in the sum of lower quotas, and produces a solution whenever at least one feasible matching exists. They reduce HRLQ to HR, via repeated proposal with priorities on the hospitals. A large number of priorities with carefully chosen capacities are used, to satisfy both the feasibility and popularity requirements. A comparison of their proofs to those in [6] yield the following observation : the few lower priorities, which are given capacities equal to the original upper quota, contribute to the larger size of the matching; further priorities, with capacities equal to the original lower quota, contribute to feasibility.

Based on the above result for HRLQ, we can ask a more general question : what if, in addition to lower quotas on hospitals, we also have some residents that *must* be matched? This is essentially a lower quota for residents, so we shall call this scenario "Hospital-Residents with Both-Sided Lower Quotas", or HR2LQ. We want to find a popular matching among feasible matchings in HR2LQ. The natural extension is to add priorities to both sides : the hospitals propose repeatedly to satisfy their lower quotas, then deficient residents can propose with higher priorities. To implement this, we exploit the reduction to stability : we make copies and dummies on both sides in the same manner as it is done in [2] for hospitals. It turns out that *if* the resulting matching is feasible, then it is also maximum-cardinality popular among feasible matchings. It is the feasibility which is potentially an issue. This is because of a crucial observation, which we detail below.

When multiple copies of a hospital are made due to priorities, a resident could potentially be matched to multiple copies of the same hospital. Thus the transformed instance could provide sufficient residents to a hospital, but when priorities are erased, this could translate to a lower number of residents for that hospital. Hence the final solution might not be feasible. Fortunately, in the algorithm for HRLQ from [2], the resident set is unchanged, so all the copies of an edge share the same resident. Residents have unit capacity, so each resident is matched to at most one copy of a hospital, and the resulting solution is feasible. In extending the idea to HR2LQ, however, we are not so fortunate : residents are also copied, so two copies of the same edge might not share a resident, and thus could both be included in the solution to the transformed instance. Hence, in this generalisation, the solution obtained might not be feasible.

We circumvent this by observing that the algorithms in [2] are also, naturally, designed to respect the upper quotas : in the transformed instance, each hospital is matched, over all its copies, to a number of non-dummy residents that is at most its upper quota. We show that this also holds in our extension to HR2LQ, for both sides – hospital or resident. Hence each resident is matched, over all its copies, to at most one non-dummy hospital. This ensures that our solution for HR2LQ is feasible, and hence also maximum-cardinality popular among feasible matchings.

In our work, we also consider the many-to-many version of HR, as studied in [8]. This can be considered a situation where residents are part-time employees, and hence can work for multiple hospitals at once. Another application is where students are allocated courses in a university : each student has a maximum number of courses they are allowed to take, each course has a capacity, and as usual they have preferences over one another. Because of this application, we shall call this the Students-Courses Problem (SC). For stability questions in SC, a blocking pair is defined in the expected manner – a student-course pair blocks a matching when it does not belong to the matching, and each endpoint is either undersubscribed or prefers the other endpoint to a matched partner. A stable matching in SC can be obtained by employing the natural extension of Gale-Shapley : students propose the same way as hospitals did, and courses accept if they either have unfilled space or can reject a less-preferred student.

Brandl and Kavitha, in [7], consider maximum-cardinality popularity in SC. They modify the Gale-Shapley algorithm by allowing priority-1 proposals, and provide a proof of their algorithm using linear programming techniques. In our work, we first consider popularity in SC under one-sided lower quotas (wlog, student-sided lower quotas). We call this problem SCLQ. Based on the existing work discussed above, we can expect that further priorities added to the algorithm in [7] should give us a solution. Thus we attempt to apply the techniques from [2] to SCLQ – however, ensuring feasibility is not easy here. All vertices have arbitrary upper quotas, so unlike in our solution for HR2LQ, upper quotas do not prevent a course from being matched to two copies of the same student. Our strategy to avoid this is to use the notion of *stability under classifications*, as studied by Huang in [3].

Using Classifications for Popularity

Classifications are studied when a natural categorization can exist among the vertices : for example, students might have course requirements in various departments; or courses might have various quotas on the number of undergraduate students, postgraduate students, and so on. For our work on SCLQ, we use *classifications* to prevent a course from matching with two copies of the same student. A classification is a collection of subsets of the neighbour set of a vertex, upon which we may impose additional constraints.

Huang, in [3], studies classifications in HRLQ, which is interpreted as the problem of allocating applicants and institutes under preferences and lower quotas, but each institute also creates its own classifications of the applicants, and places upper and lower quotas on these classes. In this situation, stability has a slightly weaker definition, since blocking pairs must take into account the upper and lower quotas of the classes. It is shown in [3] that stability in this scenario can be efficiently obtained if the classifications of each institute have a "simple enough" relationship to one another, and that the question is NP-hard otherwise. The "simplicity" criterion is that the classes of an institute must be *laminar* under inclusion, i.e. each pair of them must either be disjoint or comparable under inclusion. Thus, the problem solved in [3] is that of Laminar Classified Stable Matchings (LCSM) in HR.

In our application of classifications to Popularity in SCLQ, after creating copies of students and adding dummies, we add classifications where courses classify all copies of a particular student into a single class with unit upper quota. We hope that a stable matching in this scenario will give us the desired solution. We require a many-to-many version of the work in [3], but for classes with much nicer properties :

- Each course classifies the student-copies in the same way, i.e. the classification does not depend on the course.
- A student-copy is in exactly one class, depending on the original student that it is a copy of. Hence our classification is actually a partition.
- All classes have upper quota one and no lower quotas.
- Members of a class have a very specific order in the preference lists of neighbours, because the most preferred of each class are the highest-priority copies in that class, and so also for the second-most preferred, and so on.

Partition classified stable matching (PCSM) is a very special case of LCSM, and our classes are a further special case. It is shown in [3] that PCSM for HR is a special case of the Student-Project Allocation (SPA) problem studied in [9], and hence a linear-time algorithm must exist for it. Huang's algorithm does not depend on residents having unit upper quota, and is hence directly applicable to our algorithm for SCLQ, but it has a high worst-case complexity. Thus, we briefly study PCSM in the SC setting and obtain a linear-time solution for the case without lower quotas, which suffices for us. We then proceed to prove that this gives us a reduction from SCLQ to PCSM in SC.

As explained above, once we set all copies of the same student to be a class with upper quota one, we ensure that a course only matches with one copy of a student. Thus our solution would indeed be feasible for the given SCLQ instance. To prove maximum-cardinality popularity, we will observe that our proofs of popularity in the HR2LQ scenario do not depend on the unit capacity of residents. Thus, we adapt those proofs to the weaker definition of stability under classifications, and hence find a maximum-cardinality popular matching among feasible matchings in SCLQ.

An interesting observation is that further rounds of proposal cannot decrease the size of the matching, since rejections only occur to favour a preferred partner or higher-priority partner. We also note that priority 1 gives us a maximum-cardinality popular matching. Thus, the maximum-cardinality popular matchings under the above lower-quota conditions are the same size as a maximum-cardinality popular matching in the absence of lower quotas.

The final generalisation we consider, naturally, is both sided lower quotas in SC, i.e. SC2LQ. Here, if we want to combine ideas from our solutions of HR2LQ and SCLQ, it appears that we require a version of PCSM with classifications on both sides. A both-sided version of LCSM appears in [10], but we observe that there are SC2LQ instances where both-sided classifications may not suffice to give us feasibility. We identify the constraints that classifications do not capture, and suggest two possible approaches to mitigate this issue. We also observe an interesting property of instances where these additional constraints are crucial.

Other Approaches to Lower-Quota Constraints

Aside from studying popularity, it is also interesting to ask how far we must deviate from stability in order to guarantee feasibility.

One approach in this direction is to weaken the blocking condition, and study feasibility in the resulting relaxation of stability. Yokoi, in [11], considers such a relaxation, called *envy-freeness*. In that work, envy-freeness is studied in various settings, like HRLQ and LCSM. It is also shown there that envy-freeness coincides with stability in a certain transformed instance that is obtained by lowering the upper quotas.

Deviations from stability can also be studied via minimisation questions related to blocking pairs. Since we cannot eliminate blocking pairs altogether, we instead ask if we can minimise their number. The hardness of approximation for this problem is studied in [12]. They also provide an approximation for a different minimisation question, where the quantity to minimise is the number of residents that are involved in blocking pairs.

Chapter 2

Preliminaries

In the absence of lower quotas, the most general version of the problems stated above is the Students-Courses Problem. This scenario can be formally defined as follows :

THE STUDENTS-COURSES PROBLEM

A bipartite graph $G = (\mathcal{S} \uplus \mathcal{C}, E)$, and for each $v \in \mathcal{S} \uplus \mathcal{C}$:

1. ordered preference lists $\mathbf{pref}(v) = (E(v), <_v)$
2. upper quotas $q^+ : \mathcal{S} \uplus \mathcal{C} \to \mathbb{N}$.

We define a matching as follows:

Definition 1. *A matching $M \subseteq E$ is such that $\forall v \in \mathcal{S} \uplus \mathcal{C}, |M(v)| \leq q^+(v)$.*

2.1 Stability

In the above scenario, we define a *blocking pair* with respect to a matching as follows:

Definition 2. *A pair (u, v), $u \in \mathcal{S}, v \in \mathcal{C}$, is said to* block *a matching $M \subseteq E$, or is a* blocking pair with respect to *the matching M, when all of the three following conditions hold:*

- $(u, v) \in E \setminus M$
- *either* $|M(u)| < q^+(u)$, *or* $\exists v' \in M(u), v' <_u v$
- *either* $|M(v)| < q^+(v)$, *or* $\exists u' \in M(v), u' <_v u$

This allows us to define a *stable matching* as follows:

Definition 3. *A matching $M \subseteq E$ is* stable *if there are no blocking pairs with respect to M.*

Thus, the stability question in SC can be stated as :

STABILITY IN STUDENTS-COURSES
Input : A bipartite graph $G = (\mathcal{S} \uplus \mathcal{C}, E)$, and for each $v \in \mathcal{S} \uplus \mathcal{C}$:

1. ordered preference lists $\mathbf{pref}(v) = (E(v), <_v)$
2. upper quotas $q^+ : \mathcal{S} \uplus \mathcal{C} \to \mathbb{N}$.

Output : A subset $M \subseteq E$ such that :

1. $\forall v \in \mathcal{S} \uplus \mathcal{C}, |M(v)| \leq q^+(v)$
2. M is stable.

Further restrictions on SC give us the classical Hospital-Residents and Marriage settings :

- We set the upper quotas of one side to be 1. We call this side residents $\mathcal{R}$, and the other side hospitals $\mathcal{H}$. This gives us HR.
- In HR, we further restrict the upper quotas of hospitals to be 1, and reinterpret the two sides as men and women. This gives us the Marriage setting.

2.2 Popularity and Lower Quotas

We can add lower quotas q^- to one or both sides of the above problems, to obtain the settings that we study in our work. Now let us define some terminology that is useful to describe the properties of specific vertices in a matching :

Definition 4. *A vertex v is* undersubscribed *or* underfilled *in a matching M if $|M(v)| < q^+(v)$.*

Definition 5. *A vertex v is* deficient *in a matching M if $|M(v)| < q^-(v)$.*

Definition 6. *A matching M is* feasible for a vertex v *if v is not deficient in M. M is* feasible *if it is feasible for all vertices.*

We note that where only upper quotas are present, all matchings are feasible.

As mentioned before, stability is often far too restrictive under lower quotas, hence we shall study popularity. Below, we define popularity and then state the problems that we shall study.

With respect to the problems stated above, on a bipartite graph $G = (U \uplus V, E)$, we can formalise the *vote of a vertex* between two feasible matchings $M, M' \subseteq E$ as follows:

Definition 7. *For a vertex v, $vote_v : E(v) \times E(v) \to \{-1, 1\}$ is defined as*

$$vote_v(u, w) = \begin{cases} 1 & u >_v w \\ -1 & u <_v w \\ 0 & u = w \end{cases}$$

To extend the above notion to the comparison of M and M', we allow every vertex v to pick a correspondence **corr** between $M(v) \setminus M'(v)$ and $M'(v) \setminus M(v)$. This is formally constructed as follows :

Definition 8. *For a vertex v, if $|M(v) \setminus M'(v)| = |M'(v) \setminus M(v)|$ then $\mathbf{corr}_v$ be an arbitrary bijection between these sets. Else, wlog, $|M(v) \setminus M'(v)| < |M'(v) \setminus M(v)|$. Then fix an arbitrary bijection between $M(v) \setminus M'(v)$ and an arbitrary subset of the same size in $M'(v) \setminus M(v)$. This bijection is $\mathbf{corr}_v|_{M(v) \setminus M'(v)}$ and for any remaining vertex u in $M'(v) \setminus M(v)$, $\mathbf{corr}_v(u) = \perp$.*

Given a fixed $\mathbf{corr}_v$ for each vertex v, we can define, by abuse of notation, the vote of v between two matchings:

Definition 9. *For matchings M, M'*

$$vote_v(M, M') = \sum_{u \in M(v) \setminus M'(v)} vote_v(u, corr_v(u))$$

Here we assume that $vote_v(u, \perp)$ is always 1 for any u.

Using the above notion of voting, we define popularity as follows :

Definition 10. *M' is* more popular than *M (denoted $M' \succ M$) if*

$$\sum_{v \in U \uplus V} vote_v(M', M) > 0$$

M is popular *among a set of matchings $\mathcal{M} \subseteq 2^E$ if no matching in $\mathcal{M}$ is more popular than M.*

We shall not require **corr** to be a part of the popularity instances we study, since we will show that our algorithms work for an arbitrary **corr**.

Thus, given a set of constraints, we can talk about popularity among matchings that are feasible under those constraints. Hence we can formalise the notion of popularity in the Students-Courses Problem in the presence of lower quotas :

Popularity in Students-Courses with Both-Sided Lower Quotas
Input : A bipartite graph $G = (\mathcal{S} \uplus \mathcal{C}, E)$, and for each $v \in \mathcal{S} \uplus \mathcal{C}$:

1. ordered preference lists $\mathbf{pref}(v) = (E(v), <_v)$

2. upper and lower quotas $q^+, q^- : \mathcal{S} \uplus \mathcal{C} \to \mathbb{N}$, such that $\forall v \in \mathcal{S} \uplus \mathcal{C}, q^-(v) \leq q^+(v)$.

Output : A matching $M \subseteq E$ such that :

1. M is feasible for the above input.
2. M is popular among all matchings that are feasible for the above input.

Imposing certain restrictions on the above problem statement gives us the popularity questions in the other scenarios discussed earlier.

- Popularity in SCLQ : Fix a side in the bipartite graph, say $\mathcal{C}$, and set that $\forall v \in \mathcal{C}, q^-(v) = 0$.
- Popularity in HR2LQ : Fix a side, and set upper quotas on that side to be 1. For clarity, we rename this side $\mathcal{R}$, and call the other side $\mathcal{H}$.
- Popularity in HRLQ : Fix a side in the bipartite graph in HR2LQ, say $\mathcal{R}$, and set that $\forall v \in \mathcal{R}, q^-(v) = 0$. This popularity question is solved in [2].
- Popularity in the Marriage setting with special men/women that must be matched, can also be obtained by setting unit lower quotas to the special vertices.

Alternating Paths and Cycles

The standard technique for proving popularity in the one-to-one Marriage scenario involves the comparison of two matchings M_1, M_2 by the decomposition of $M_1 \oplus M_2$ into alternating paths and cycles. In that case, the matching we seek is also a matching in the graph-theoretic sense, and hence the decomposition is immediate via the connected components of $G|_{M_1 \oplus M_2}$.

As observed in [2], there is no such natural decomposition in the many-to-one case. And of course, the same issue extends to the many-to-many case. Here, we extend the procedure suggested in [2] for the construction of alternating paths in the many-to-one case, to the many-to-many case. We repeat the procedure below for every connected component in $G|_{M_0 \oplus M_1} = (\mathcal{S} \uplus \mathcal{C}, M_0 \oplus M_1)$. Initially, we pick an $e \in M_1$ (wlog) within the chosen connected component. We set $\rho = \langle e \rangle$. Then, inductively:

1. We pick an endpoint of ρ, say $v \in \mathcal{S}$, such that ρ is of the form $\langle \dots u, v \rangle$, and $(u, v) \in M_b, b \in \{0, 1\}$. Then let $\bar{u} = \mathbf{corr}_v(u)$. If $\bar{u} \neq \bot$, we append $(v, \bar{u})$, which is in M_{1-b}, to ρ.
2. We proceed similarly as above with an endpoint of ρ which is a course vertex.
3. The process is terminated either when ρ can no longer be extended, i.e. when the **corr** of each endpoint is $\bot$; or when we complete a cycle, i.e. we have $\rho = \langle u, v', \dots, u', v \rangle$ such that $u = \mathbf{corr}_v(u')$ and $v = \mathbf{corr}_u(v')$, so we can add (u, v) to ρ.

Now we define some terms related to the size of popular matchings.

Definition 11. *M is* maximum cardinality popular *among the matchings $\mathcal{M}$ if larger matchings in $\mathcal{M}$ are not popular, i.e. for any $N \in \mathcal{M}$ such that $|N| > |M|$, $\exists M_N \in \mathcal{M}, M_N \succ N$.*

If we impose the additional condition that M itself must be a suitable M_N for every N as per the above definition, then we get a stronger property called *dominance*, which was introduced by Cseh and Kavitha in [13].

Definition 12. *M is* dominant *among the matchings in $\mathcal{M}$ if for any $N \in \mathcal{M}$ such that $|N| > |M|$, $M \succ N$.*

In [2], maximum cardinality popularity is proved by proving the stronger dominance condition. In fact, it is shown in [14] that finding a maximum cardinality popular matching that is neither stable nor dominant is NP-hard in the HR setting itself. Hence we can expect that constructive proofs of the existence of a maximum cardinality popular matching will proceed via proof of dominance. That is indeed the case here, since very similar proofs to [2] are sufficient for our cardinality results.

2.3 Stability under Classifications

The proofs in [2], as well as those that we present here, rely on reducing the relevant question to a version without lower quotas, by making multiple copies of certain vertices. To maintain feasibility, however, we need to capture the notion that certain vertices in the transformed instance are in fact copies of the same original vertex, and hence should not be matched to the same partner twice. As mentioned before, we make certain observations to circumvent this issue in HRLQ and HR2LQ; but for SCLQ, we utilise the notion of *stability under classifications*. Hence, let us define some basic terminology regarding matchings under classifications. The classified setting studied in [3] was a generalisation of HR, with hospitals imposing class constraints on residents. Extending this to the SC setting, we obtain the following problem :

STUDENTS-COURSES WITH CLASSIFICATIONS
A bipartite graph $G = (\mathcal{S} \uplus \mathcal{C}, E)$, and for each $v \in \mathcal{S} \uplus \mathcal{C}$:

1. ordered preference lists $\mathbf{pref}(v) = (E(v), <_v)$
2. upper quotas $q^+ : \mathcal{S} \uplus \mathcal{C} \to \mathbb{N}$

and for each course $u \in \mathcal{C}$, a classification $\mathscr{C}_u \subseteq 2^{E(u)}$, with upper and lower quotas $q_u^+, q_u^- : \mathscr{C}_u \to \mathbb{N}$, such that for each $C \in \mathscr{C}_u, q_u^-(C) \leq q_u^+(C)$.

When all students have upper quota 1, the above coincides with LCSM in HR, with hospitals classifying residents, which is the setting studied in [3]. As expected, a matching M is said to be *feasible* for a classified setting if for each vertex u and any class $C \in \mathscr{C}_u, q_u^-(C) \leq |M(u) \cap C| \leq q_u^+(C)$.

Under classifications, a natural question to ask is once again that of *stability*. A stable matching that is feasible for the class quotas may not exist; moreover, the motivation for a blocking pair, which is of a pair that has incentive to defy the assigned matching, must now take class quotas into account. In the aforementioned work, in [3], a *blocking group* is defined to capture this notion. However, it is also proved there that when classes do not have lower quotas, a modified definition of blocking pair suffices to characterise the same notion of classified stability. Since in our application we do not need lower quotas on classes, we use this modified definition of blocking pair, which we call a *feasible blocking pair*. In the absence of classifications, this shall coincide with the usual definition of a blocking pair. We state this definition in the more general SC setting – as exhibited before, HR is obtained from SC by imposing unit upper quotas on one side.

Definition 13. *$(v, u) \in \mathcal{S} \times \mathcal{C}$ is said to be a* feasible blocking pair *with respect to a matching M if it is a blocking pair, and for any class $C_u^v \in \mathscr{C}_u$ containing v, either $|M(u) \cap C_u^v| < q_u^+(C_u^v)$ or $\exists v' \in M(u) \cap C_u^v, v' <_u v$.*

Accordingly, we have a modified definition of stability, which coincides with the usual definition in the absence of classifications.

Definition 14. *A matching M is said to be* stable under classifications *or* class-stable *if there are no feasible blocking pairs with respect to M.*

It is shown in [3] that when the classifications in HR are laminar, a class-stable matching can be found in polynomial time. This problem is called Laminar Classified Stable Matchings (LCSM) in HR. For our purpose, we have classes that are not just laminar but in fact a partition; moreover we do not require class lower quotas. Hence we shall adapt the algorithm from [3] to Partition Classified Stable Matchings (PCSM) in SC.

Remark. SCLQ (under course-sided lower quotas) is also a special case of the classified setting when for $\forall u \in \mathcal{C}, \mathscr{C}_u = \{E(u)\}$. Hence LCSM (in fact, PCSM) in SC will find a stable feasible matching in SCLQ, if one exists. Our goal here is different – we shall use PCSM as a tool in an algorithm to find *popular* matchings in SCLQ. The correctness of our algorithm will exhibit that if feasible matchings exist in an SCLQ instance, then there is a matching that is popular among them.

2.4 Complexity Calculations

We use n to be the total number of vertices and $m = |E|$ in any instance. We note that in a feasible instance, the sum of lower quotas on each side must not exceed the sum of upper quotas on the other side. Thus we will assume that instances that do not meet this criterion are eliminated before using the algorithm, and hence this criterion holds for the purpose of calculating time complexity.

"Linear time", for us, means linear in the number of edges. By [1], the Gale-Shapley algorithm takes linear time.

Chapter 3

Popularity in HR2LQ

Popularity in Hospital-Residents with Both-Sided Lower Quotas

Input : A bipartite graph $G = (\mathcal{H} \uplus \mathcal{R}, E)$, and for each $v \in \mathcal{H} \uplus \mathcal{R}$:

1. ordered preference lists $\mathbf{pref}(v) = (E(v), <_v)$
2. upper and lower quotas $q^+, q^- : \mathcal{H} \uplus \mathcal{R} \to \mathbb{N}$, such that $\forall v \in \mathcal{H} \uplus \mathcal{R}, q^-(v) \leq q^+(v)$, and $q^+(\mathcal{R}) = \{1\}$

Output : A subset $M \subseteq E$ such that is popular among all matchings feasible for the above input.

We shall give an algorithm to find a matching that is maximum-cardinality popular among all feasible matchings in the HR2LQ setting. We obtain this algorithm via a reduction to HR *without* lower quotas.

3.1 Transformation to a Gale-Shapley instance

We build the following Gale-Shapley instance from the above instance. Let

$$\mu_H = \sum_{h \in \mathcal{H}} q^-(h) + 2$$

and

$$\mu_R = \sum_{r \in \mathcal{R}} q^-(r) + 2$$

We create copies $h^0, \dots, h^{\mu_H - 1}$ for each hospital h, and similarly μ_R copies for each resident. For each $a \in \mathcal{H} \uplus \mathcal{R}$, we set the upper quotas of its copies as follows :

$$q^+(a^s) = \begin{cases} q^+(a) & s \in \{0, 1\} \\ q^-(a) & \text{otherwise} \end{cases}$$

We also add dummy vertices $\mathcal{D}_{a^s} = \{d^i_{a^s} \mid i \in [q^+(a^s)]\}$ for each s except the highest. The collection of all dummy vertices is called $\mathcal{D}$. The preference lists are as follows, where $\mathbf{pref}(a)^s$ is the s-level copy of $\mathbf{pref}(a)$. For each hospital h :

$$\begin{aligned}
h^{\mu_H-1} &: d^1_{h^{\mu_H-2}} \ \dots \ d^{q^-(h)}_{h^{\mu_H-2}} \ \mathbf{pref}(h)^{\mu_R-1} \ \dots \ \mathbf{pref}(h)^0 \\
\forall 2 < s < \mu_H - 1, h^s &: d^1_{h^{s-1}} \ \dots \ d^{q^-(h)}_{h^{s-1}} \ \mathbf{pref}(h)^{\mu_R-1} \ \dots \ \mathbf{pref}(h)^0 \ d^1_{h^s} \ \dots \ d^{q^-(h)}_{h^s} \\
h^2 &: d^{q^+(h)-q^-(h)+1}_{h^1} \ \dots \ d^{q^+(h)}_{h_1} \ \mathbf{pref}(h)^{\mu_R-1} \ \dots \ \mathbf{pref}(h)^0 \ d^1_{h^2} \ \dots \ d^{q^-(h)}_{h^2} \\
h^1 &: d^1_{h^0} \ \dots \ d^{q^+(h)}_{h^0} \ \mathbf{pref}(h)^{\mu_R-1} \ \dots \ \mathbf{pref}(h)^0 \ d^1_{h^1} \ \dots \ d^{q^+(h)}_{h^1} \\
h^0 &: \mathbf{pref}(h)^{\mu_R-1} \ \dots \ \mathbf{pref}(h)^0 \ d^1_{h^0} \ \dots \ d^{q^+(h)}_{h^0}
\end{aligned}$$

and similarly for each resident r, except r^1 also has upper quota $q^-(r)$.

Remark. Maximum-cardinality popularity is given by one higher priority with full upper quota on *any one side*, as in [6]. Here, just like in [2], we choose this side to be $\mathcal{H}$. For feasibility, we use a large number of further priorities with the upper quotas capped to the original lower quotas. For each side, we need this number to be larger than the sum of all lower quotas on that side. Extra priorities beyond this are not useful, but do not affect correctness either, since only the lower bound on this number is crucial. This means that we could have used one less priority for $\mathcal{R}$ here; but using similar expressions for μ_H and μ_R leads to simpler proofs, so we proceed as above.

For the dummy residents, the preferences are as follows:

$$\begin{aligned}
\forall i \in [q^+(h)], d^i_{h^0} &: h^0 \ h^1 \\
\forall i \in \{1, \dots, q^+(h) - q^-(h)\}, d^i_{h^1} &: h^1 \\
\forall i \in \{q^+(h) - q^-(h) + 1, \dots, q^+(h)\}, d^i_{h^1} &: h^1 \ h^2 \\
\forall i \in [q^-(h)], 2 \le s < \mu_H - 1, d^i_{h^s} &: h^s \ h^{s+1}
\end{aligned}$$

and similarly for the dummy hospitals. Dummy hospitals have upper quota 1. We call the Gale-Shapley output on this instance M'. We obtain a solution to the original instance from M' as follows :

$$M = \{(h, r) \mid \exists s, t, (h^s, r^t) \in M'\}$$

We illustrate the algorithm via an example below.

Example

The original instance is as follows. All upper quotas are 1. Lower quotas for h_4, r_3, r_4 are 1, and for others are 0.

$h_1 : r_1 \ r_2 \ r_3 \ r_4$	$r_1 : h_1 \ h_2 \ h_3 \ h_4$
$h_2 : r_1 \ r_2 \ r_3$	$r_2 : h_1 \ h_2$
$h_3 : r_1$	$r_3 : h_1 \ h_2$
$h_4 : r_1$	$r_4 : h_1$

We get $\mu_H = 3, \mu_R = 4$. As per the remark above, 3 priorities for residents is actually enough, and this change does not affect correctness. Thus, we will be concise by using only 3 priorities on either

side in the transformed instance below.
The copies and dummies are made as follows. If a vertex has lower quota 0, its higher copies have upper quota 0, i.e. will never be matched. So we have omitted these copies and their corresponding dummies. Also, upper quotas are all one, hence each vertex will have at most one associated dummy, so we shall omit superscripts on dummies.
The transformed instance has no (i.e. all-zero) lower quotas, and upper quotas of all the vertices shown below are 1.

$$\begin{aligned}
h_1^0 &: r_3^2\ r_4^2\ r_3^1\ r_4^1\ r_1^0\ r_2^0\ r_3^0\ r_4^0\ d_{h_1^0} \\
h_1^1 &: d_{h_1^0}\ r_3^2\ r_4^2\ r_3^1\ r_4^1\ r_1^0\ r_2^0\ r_3^0\ r_4^0 \\
h_2^0 &: r_3^2\ r_3^1\ r_1^0\ r_2^0\ r_3^0\ d_{h_2^0} \\
h_2^1 &: d_{h_2^0}\ r_3^2\ r_3^1\ r_1^0\ r_2^0\ r_3^0 \\
h_3^0 &: r_1^0\ d_{h_3^0} \\
h_3^1 &: d_{h_3^0}\ r_1^0 \\
h_4^0 &: r_1^0\ d_{h_4^0} \\
h_4^1 &: d_{h_4^0}\ r_1^0\ d_{h_4^1} \\
h_4^2 &: d_{h_4^1}\ r_1^0 \\
d_{r_3^0} &: r_3^0\ r_3^1 \\
d_{r_3^1} &: r_3^1\ r_3^2 \\
d_{r_4^0} &: r_4^0\ r_4^1 \\
d_{r_4^1} &: r_4^1\ r_4^2
\end{aligned}$$

$$\begin{aligned}
r_1^0 &: h_4^2\ h_1^1\ h_2^1\ h_3^1\ h_4^1\ h_1^0\ h_2^0\ h_3^0\ h_4^0 \\
r_2^0 &: h_1^1\ h_2^1\ h_1^0\ h_2^0 \\
r_3^0 &: h_1^1\ h_2^1\ h_1^0\ h_2^0\ d_{r_3^0} \\
r_3^1 &: d_{r_3^0}\ h_1^1\ h_2^1\ h_1^0\ h_2^0\ d_{r_3^1} \\
r_3^2 &: d_{r_3^1} h_1^1\ h_2^1\ h_1^0\ h_2^0 \\
r_4^0 &: h_1^1\ h_1^0\ d_{r_4^0} \\
r_4^1 &: d_{r_4^0}\ h_1^1\ h_1^0\ d_{r_4^1} \\
r_4^2 &: d_{r_4^1}\ h_1^1\ h_1^0 \\
d_{h_1^0} &: h_1^0\ h_1^1 \\
d_{h_2^0} &: h_2^0\ h_2^1 \\
d_{h_3^0} &: h_3^0\ h_3^1 \\
d_{h_4^0} &: h_4^0\ h_4^1 \\
d_{h_4^1} &: h_4^1\ h_4^2
\end{aligned}$$

The hospital-proposing stable matching in the above instance is

$$\begin{aligned}
M' = \{&(h_1^0, r_4^2), (h_1^1, d_{h_1^0}), (h_2^0, r_3^1), (h_2^1, d_{h_2^0}), \\
&(h_3^0, d_{h_3^0}), (h_4^0, d_{h_4^0}), (h_4^1, d_{h_4^1}), (h_4^2, r_1^0), \\
&(d_{r_3^0}, r_3^0), (d_{r_3^1}, r_3^2), (d_{r_4^0}, r_4^0), (d_{r_4^1}, r_4^1)\}
\end{aligned}$$

Erasing priorities and dummies, we get the following matching in the original instance :

$$M = \{(h_1, r_4), (h_2, r_3), (h_4, r_1)\}$$

which matches h_4, r_3, r_4 as required by their lower quotas.

Now we shall prove that M is indeed a maximum-cardinality popular matching among feasible matchings in the original instance.

3.2 Correctness and Complexity

First, we shall prove some properties of M, which shall aid us in establishing the correctness of the above algorithm.

We want to prove certain results that apply to both sides, hospitals or residents. Hence, we shall fix an arbitrary side $V \in \{\mathcal{H}, \mathcal{R}\}$, and pick μ to be μ_H or μ_R accordingly. We shall call the other side U. For example, when $V = \mathcal{H}$ then $\mu = \mu_H$ and $U = \mathcal{R}$.

Lemma 15. *M is a matching, i.e. for $v \in V$, $\left|\cup_{s=0}^{\mu-1} M'(v^s) \setminus \mathcal{D}\right| \leq q^+(v)$.*

Proof. Dummies that appear in the beginning of any preference list must be matched. Thus $\left|\cup_{s=0}^{\mu-1} M'(v^s) \cap \mathcal{D}\right| = \sum_{s=1}^{\mu-1} q^+(v^s)$. But $\left|\cup_{s=0}^{\mu-1} M'(v^s)\right| \leq \sum_{s=0}^{\mu-1} q^+(v^s)$. So $\left|\cup_{s=0}^{\mu-1} M'(v^s) \setminus \mathcal{D}\right| \leq q^+(v^0) = q^+(v)$. ∎

The above result, due to the unit capacity of residents, ensures that multiple copies of the same edge do not appear in M', thus allowing us to prove feasibility of M via M'.

Definition 16. *For some $v \in V$, we call its s-level copy v^s* active *if $M'(v^s) \setminus \mathcal{D} \neq \emptyset$. we say v is* active at level s *if v^s is active.*

Lemma 17. *Let $v \in V$ be active at level s. Then*

1. *$M'(v^{s-1}) \cap \mathcal{D}_{v^{s-1}} \neq \emptyset$*
2. *For $j < s-1, M'(v^j) \subseteq \mathcal{D}_{v^j}$*
3. *For $j > s+1, M'(v^j) \subseteq \mathcal{D}_{v^{j-1}}$*

Proof. Fix $\bar{u}^{\bar{t}} \in M'(v^s) \setminus \mathcal{D}$.

1. Suppose not, then all dummies in $\mathcal{D}_{v^{s-1}}$ are either unmatched or matched with v^s. But $|\mathcal{D}_{v^{s-1}}| \geq q^+(v^s)$ and v^s is active at level s, so $\exists d_{v^{s-1}} \in \mathcal{D}_{v^{s-1}}$ which is unmatched. But $d_{v^{s-1}} >_{v^s} \bar{u}^{\bar{t}}$, so $(d_{v^{s-1}}, v^s)$ blocks M'.
2. Suppose not. Then pick a vertex in $M'(v^j) \setminus \mathcal{D}_{v^j}$. Suppose this is a non-dummy vertex u^t. But $v^{s-1} >_{u^t} v^j$, and $u^t >_{v^{s-1}} d_{v^{s-1}}$. So (v^{s-1}, u^t) would block M', but we know M' is stable.
   Thus if $j = 0$ then we are done. Else, $M'(v^j) \setminus \mathcal{D}_{v^j} \subseteq \mathcal{D}_{v^{j-1}}$. Since we have proved the result for the case $j = 0$, inductively assume that $M'(v^{j-1}) \subseteq \mathcal{D}_{v^{j-1}}$. But the dummies in $\mathcal{D}_{v^{j-1}}$ prefer v^{j-1} over v^j. Hence $M'(v^j) \cap \mathcal{D}_{v^{j-1}} = \emptyset$.
3. If $\exists u^t \in M'(v^j) \setminus \mathcal{D}$, then by Part 1, $\exists d_{v^{j-1}} \in M'(v^{j-1}) \cap \mathcal{D}_{v^{j-1}}$. But $(v^{j-1}, \bar{u}^{\bar{t}})$ would block M'.

∎

Corollary 18. *Any $v \in V$ is active at not more than 2 levels, which must be consecutive.*

Proof. If v is not active at any level, we are done. Else let s be the lowest level at which v is active. By Part 3 of Lemma 17, we are done. ∎

Definition 19. *The set of vertices in V that are active at level j is called V^j. Consider $u \in U$. Vertices in $M(V^j)$ are said to be in U_j. Also, by convention, $u \in U_0$ if $|M(u)| < q^+(u)$.*

Lemma 20. *Let $v \in V, u \in U$. Let s be the lowest level at which v is active. If $\exists j < s-1, u \in U_j \setminus M(v)$ then $(u,v) \notin E$.*

Proof. Suppose not. Then pick such a j. Since $u \in U_j, \exists \bar{v}$ such that $(\bar{v}^j, u^t) \in M'$. Also by Part 1 of Lemma 17, $\exists d_{v^{s-1}} \in M(v^s) \cap \mathcal{D}_{v^{s-1}}$. Then (v^{s-1}, u^t) would block M'. ■

Corollary 21. *If $v \in V$ is active at level $s+1$, then it has no neighbours in $U_j \setminus M(v)$ for any $j \leq s-1$.*

Proof. Immediate from Lemma 20 and Corollary 18. ■

Lemma 22. *Consider $v \in V$. Then*

1. *If $|M'(v^s)| < q^+(v^s)$, then $s = \mu - 1$.*
2. *If $|M(v)| < q^-(v)$, then v is active at level $\mu - 1$, and has no edges to U_j for any $j < \mu - 1$.*
3. *If $q^-(v) < |M(v)|$ then v is not active at levels above 1.*
4. *If $V = \mathcal{H}, |M(v)| < q^+(v)$, then $v \notin V^0$.*

Proof.

1. If $s < \mu - 1$, consider any dummy d in $\mathcal{D}_{v^s}$. If d is unmatched in M' or matched to v^{s+1}, then (v^s, d) would block M', since $v^s >_d v^{s+1}$. Thus $s = \mu - 1$.
2. Here $q^-(v)$ must be positive. The key fact for this proof is that the highest level does not have dummies at the end of its list, so $\sum_{s=0}^{\mu-1} q^+(v^s) - \sum_{s=0}^{\mu-1} |\mathcal{D}_{v^s}| = q^-(v) \leq q^+(v^s)$ for all levels s. So $\exists s, |M'(v^s)| < q^+(v^s)$. So by Part 1, $s = \mu - 1$. Now suppose v is active at level $s' < \mu - 1$, with a non-dummy partner u^t. Then $(v^{\mu-1}, u^t)$ would block M'. So v can only be active at level $\mu - 1$, and since all dummies would be matched to lower levels, v must be active at level $\mu - 1$.
3. Suppose the highest level at which v is active is level s. If $s > 1$, consider two cases. If $\left|M'(v^{s-1})\right| \setminus \mathcal{D} \geq q^-(v)$, then $q^-(v)$ dummies from $\mathcal{D}_{v^{s-1}}$ must be matched to v^s. Thus v^s cannot be active. Else, suppose $\left|M'(v^{s-1})\right| = k < q^-(v)$. Then at least $q^+(v^{s-1}) - k$ dummies from $\mathcal{D}_{v^{s-1}}$ are matched to v^{s-1}, and the remaining k are matched to v^s. Thus v^s can be matched to at most $q^-(v) - k$ non-dummies, so $|M(v)| \leq q^-(v)$. Hence we conclude that $s \leq 1$.

4. If $V = \mathcal{H}, v \in V^0$, and $|M(v)| < q^+(v)$, then $q^+(v^1) = q^+(v)$ so $\left|M'(v^0)\right| \setminus \mathcal{D} = k < q^+(v)$. Hence $q^+(v) - k$ dummies from $\mathcal{D}_{v^0}$ are matched to v^0, and the remaining k are matched to v^1. $M'(v^1) = q^+(v)$ by Part 1. Now suppose some $d \in \mathcal{D}_{v^1}$ is matched to v^1. Consider $u^t \in M'(v^0) \setminus \mathcal{D}$. But then (u^t, v^1) would block M'. So $\left|M'(v^1) \setminus \mathcal{D}\right| = q^+(v) - k$. But then $|M(v)| = q^+(v)$. Hence $v \notin V^0$.

■

Theorem 23. *If the original instance admits a feasible matching, then M is feasible for the original instance.*

Proof. In this proof, v_i's belong to V and u_i's belong to U.
Suppose for some $v_0 \in V, |M(v_0)| < q^-(v_0)$. For a feasible matching N, consider the decomposition of $M \oplus N$ into alternating paths and cycles. Since $|M(v_0)| < q^-(v_0) \leq |N(v_0)|$, there must be a path ρ in $M \oplus N$ which ends with an N-edge incident on v_0. The other endpoint may be in V or in U. We consider either case below:

- **Other endpoint in V :** Then ρ is of the form $\langle v_0, u_1, v_1, \ldots, u_k, v_k, u_{k+1}, v_{k+1}\rangle$, where $(u_1, v_1), (u_k, v_k) \in M$. By Lemma 22, $v_0 \in V^{\mu-1}$, and hence $u_1 \in U_{\mu-1}$, so $v_1 \in V^{\mu-1}$. Now by Lemma 20, $u_2 \in U_{\mu-1} \cup U_{\mu-2}$ and $v_2 \in V^{\mu-1} \cup V^{\mu-2}$. Continuing this reasoning, any v_i in this path is not in V^j for $j < \mu - i$. But since $(u_{k+1}, v_{k+1}) \in M$, $|M(v_{k+1})| > |N(v_{k+1})| \geq q^-(v_{k+1})$ hence by Lemma 22, $v_{k+1} \in V^1 \cup V^0$. But then $1 \geq \mu - k - 1$ i.e.

$$k \geq \mu - 2$$

  But this also means that $v_k \notin V^3 \cup \cdots \cup V^{\mu-1}$, so by Lemma 20, since $v_0 \in V^{\mu-1}$, $v_0, v_1, \ldots, v_k$ contains vertices from each of $V^2, \ldots, V^{\mu-1}$, but by Lemma 22, each such v_i can appear in ρ at most $q^-(v_i)$ times. Thus $k + 1 \leq \sum_{v \in V} q^-(v) < \mu - 1$, i.e.

$$k < \mu - 2$$

  This is a contradiction.

- **Other endpoint in U :** This argument is overall similar to the previous case. Here ρ is of the form $\langle v_0, u_1, v_1, \ldots, u_k, v_k, u_{k+1}\rangle$ where $(u_1, v_1), (u_k, v_k) \in M$. As above, $v_0 \in V^{\mu-1}$ and for any $v_i \in \rho$, $v_i \notin V_j$ for $j < \mu - i$. Now $|M(u_{k+1})| < |N(u_{k+1})| \leq q^+(u_{k+1})$, so $u_{k+1} \in U_0$, so by Lemma 20 $v_k \in V^0 \cup V^1$, hence $1 \geq \mu - k$, i.e.

$$k \geq \mu - 1$$

  Also by Lemma 20 and Corollary 18, $v_k \notin V^2 \cup \cdots \cup V^{\mu-1}$, but $v_0 \in V^{\mu-1}$, so by Lemma 20, $v_0, v_1, \ldots, v_k$ contain vertices from each of $V^2, \ldots, V^{\mu-1}$, which by Lemma 22 contain at most $q^-(v_i)$ occurrences of each v_i, giving us as before

$$k < \mu - 2$$

  which is a contradiction.

■

To prove popularity, we shall consider an arbitrary feasible matching N, and show that

$$\sum_{v \in \mathcal{H} \uplus \mathcal{R}} vote_v(N, M) \leq 0$$

To accomplish this, we again consider alternating paths and cycles in the symmetric difference $M \oplus N$. For each such path or cycle ρ, we label each vertex v that appears in ρ by its vote between its two neighbours in ρ. Votes in favour of M are labelled by $-$, and votes in favour of N by $+$, to correspond to the signs in the above expression. We want to show that there are at least as many $-$ votes as $+$ votes.

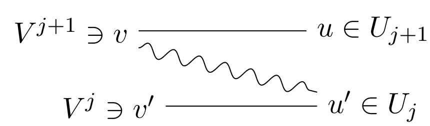
The labelling on vertices induces a labelling on the edges of either matching, and edges labelled $(+,-)$ represent opposing votes cancelling out. We will see that certain $(+,+)$-labelled N-edges can be ruled out via easy stability arguments. Hence, our strategy is to consider N-edges, and count the $(+,+)$ and $(-,-)$ labels.

In the figures accompanying subsequent arguments, the curly lines represent the edges of this arbitrary N, while the straight lines represent the edges of our M.

Observation 24. *At each level j, there are no $(+,+)$ edges in $V^j \times U_j$.*

Hereafter, we will denote the number of $(+,+)$ edges on a path or cycle ρ as $\#(+,+)_\rho$, and similarly for $(-,-)$ edges.

Lemma 25. *Consider the following edges, where $\mathbf{corr}_v(u) = u', \mathbf{corr}_{u'}(v) = v'$*



Then (u', v) is a $(-,-)$ edge.

Proof. $v \in V^{j+1}$, so $\exists d \in \mathcal{D}, (d, v^j) \in M'$, and for any level t, $u'^t >_{v^j} d$. Also if $v >_{u'} v'$ then $v^j >_{u'^t} v'^j$. And indeed $\exists t, (u'^t, v'^j) \in M'$, so (u'^t, v^j) would block M'. Also if $u' >_v u$, then for any level t, $u'^t >_{v^{j+1}} u^t$, but $v^{j+1} >_{u'^t} v'^j$, so (u'^t, v^{j+1}) would block M'. Thus by the stability of M', (u', v) is a $(-,-)$ edge. ■

Lemma 26. *Consider an N−N alternating subpath from V to U, call it $\rho = \langle v_0, u_1, v_1, \ldots, u_k, v_k, u_{k+1} \rangle$. Also let $v_0 \in V^{p+1}, u_{k+1} \in U_q$. Then $\#(+,+)_\rho \leq \#(-,-)_\rho + q - p$.*

Proof. We induct on $\#(-,-)_\rho$.

Base case. $\#(-,-)_\rho = 0$. Since $v_0 \in V^{p+1}$, by Lemma 20, $u_1 \notin U_0 \cup \cdots \cup U_{p-1}$. But if $u_1 \in U_p$, then by Lemma 25, (v_0, u_1) would be a $(-,-)$ edge. Thus $u_1 \notin U_0 \cup \cdots \cup U_p$. So $v_1 \in V^{p+1} \cup \cdots \cup V^{\mu-1}$.

Repeating this reasoning, each $v_i \in V^{p+1} \cup \cdots \cup V^{\mu-1}$. But by Observation 24 edges of the form (v_i, u_{i+1}) are not $(+,+)$ if $v_i \in V^j, u_{i+1} \in U_j$ for some j. Hence, let us reconsider the above argument to maximise $\#(+,+)_\rho$. We have the following.
$v_0 \in V^{p+1}$, so $u_1 \notin U_0 \cup \cdots \cup U_p$, and $v_1 \in V^{p+1} \cup \cdots \cup V^{\mu-1}$. Then $u_2 \notin U_0 \cup \cdots \cup U_p$, but for (v_1, u_2) to be $(+,+)$, we actually need $u_2 \notin U_{p+1}$. So $v_2 \in V^{p+2} \cup \cdots \cup V^{\mu-1}$. Now we repeat the same reasoning subsequently, and see that for all (v_i, u_{i+1}) to be $(+,+)$, we require that $\forall i \in \{0, \ldots, k+1\}, u_i \notin U_0 \cup \cdots \cup U_{p+i-1}$.
But since $u_{k+1} \in U_q$, $q > p+k$ i.e. $q - p \geq k+1 \geq \#(+,+)_\rho$ since we have $k+1$ N-edges in ρ. So $\#(+,+)_\rho \leq q - p = \#(-,-)_\rho + q - p$.
Induction Hypothesis. The lemma holds when $\#(-,-)_\rho < c$.
Induction Step. Consider the case where $\#(-,-)_\rho = c$. Choose one such $(-,-)$ edge, say (v_t, u_{t+1}). Consider the subpaths $\rho_1 = \langle v_0, \ldots, u_t \rangle$ and $\rho_2 = \langle v_{t+1}, \ldots, u_{k+1} \rangle$. Let $v_t \in V^a, v_{t+1} \in V^b$, so $u_t \in U_a, u_{t+1} \in U_b$. We have $\#(-,-)_{\rho_1} < c, \#(-,-)_{\rho_2} < c$, so the induction hypothesis holds for them. Hence $\#(+,+)_{\rho_1} \leq \#(-,-)_{\rho_1} + a - p, \#(+,+)_{\rho_2} \leq \#(-,-)_{\rho_2} + q - b$. Also since $v_t \in V^a$, by Lemma 20, $u_{t+1} \notin U_0 \cup \cdots \cup U_{a-2}$ so $b \geq a - 1$ i.e. $a - 1 - b \leq 0$. Thus

$$\begin{aligned}\#(+,+)_\rho &= \#(+,+)_{\rho_1} + \#(+,+)_{\rho_2} \\ &\leq \#(-,-)_{\rho_1} + \#(-,-)_{\rho_2} + a - p + q - b \\ &= \#(-,-)_\rho - 1 + a - p + q - b \\ &= \#(-,-)_\rho + q - p\end{aligned}$$

■

Theorem 27. *M is popular among all matchings that are feasible in G.*

Proof. We consider an arbitrary path or cycle ρ in $M \oplus N$. Within this proof, the sum of votes on ρ will be called $vote(\rho)$, which we will show is non-positive for any ρ.

Case 1. ρ is an $N-N$ path. Let it be of the form $\langle v_0, u_1, v_1, \ldots, u_k, v_k, u_{k+1} \rangle$, $v_0 \in V, u_{k+1} \in U$. $|M(u_{k+1})| < |N(u_{k+1})| \leq q^+(u_{k+1})$, so $u_{k+1} \in U_0$. Since the two ends of ρ are on opposite sides, we can wlog assume $V = \mathcal{H}$. Then by Lemma 22, since $|M(v_0)| < q^+(v_0)$, $v_0 \in V^1 \cup \cdots \cup V^{\mu-1}$, suppose $v_0 \in V^{p+1}$. Now by Lemma 26, $\#(+,+)_\rho \leq \#(-,-)_\rho + 0 - p \leq \#(-,-)_\rho$. Since the end edges are from N, this already accounts for the votes of the endpoints. Thus $vote(\rho) \leq 0$.

Case 2. ρ is an $N - M$ path. Let it be of the form $\langle v_0, u_1, v_1, \ldots, u_k, v_k \rangle$. As before, $v_0 \in V^1 \cup \cdots \cup V^{\mu-1}$, so we can choose $p \geq 1, v_0 \in V^{p+1}$. Now $|M(v_k)| > |N(v_k)| \geq q^-(v_k)$, so $v_k \in V^0 \cup V^1$, hence $u_k \in U_0 \cup U_1$. Hence $\exists q \leq 1, u_k \in U_q$. Consider the subpath $\rho' = \langle v_0, \ldots, u_k \rangle$. By Lemma 26, $\#(+,+)_{\rho'} \leq \#(-,-)_{\rho'} + q - p \leq \#(-,-)_{\rho'}$, hence the same holds for ρ. Also the votes of either endpoint cancel out, so $vote(\rho) \leq 0$.

Case 3. ρ is an $M - M$ path. Let it be of the form $\langle u_1, v_1, \ldots, u_k, v_k \rangle$. If every $v_i \in V^0$, then every $u_i \in U_0$, so there are no $(+,+)$ edges. Otherwise, consider the smallest t such that $v_t \notin V^0$. As before, $v_k \in V^0 \cup V^1$ so $u_k \in U_0 \cup U_1$, i.e. $\exists q \leq 1, u_k \in U_q$.

Consider the subpath $\rho_2 = \langle v_t, \ldots, u_k \rangle$. By Lemma 26 with $p \geq 0$ and $q \leq 1$, $\#(+,+)_{\rho_2} \leq \#(-,-)_{\rho_2} + q - p \leq \#(-,-)_{\rho_2} + 1$. In $\rho_1 = \langle u_1, \ldots, u_t \rangle$, there are no $(-,-)$ edges, hence $\#(+,+)_\rho \leq \#(-,-) + 1$. However, the endpoints of ρ contribute two $-$ votes. Thus $vote(\rho) \leq 0$.

Case 4. ρ is a cycle. Consider an M-edge in ρ. Call it e. $\rho \setminus \{e\}$ is an $N - N$ alternating path from V to U, where as per Lemma 26, $p = q$. Thus $vote(\rho) \leq 0$.

■

Theorem 28. *For any matching N that is feasible in G, if $|N| > |M|$, then $M \succ N$.*

Proof. Since $|N| > |M|$, there must be an $N - N$ alternating path in $M \oplus N$. Call this path ρ. Let ρ be of the form $\langle v_0, u_1, v_1, \ldots, u_{k-1}, v_{k-1}, u_k \rangle$. Then $u_k \in U_0$. Also since the two endpoints of ρ are on opposite sides, we can wlog assume $V = \mathcal{H}$, so by Lemma 22, $v_0 \notin V^0$. Thus, since an M-edge is always between some V^s and U_s, there must be an N-edge in ρ of the form (v_i, u_{i+1}) such that $v_i \in V^s$ for some $s > 0$ and $u_{i+1} \in U_{s'}$, $s' < s$. By Corollary 21, $s' = s - 1$. Then by Lemma 25, (v_i, u_{i+1}) is labelled $(-,-)$. Now consider the $N - N$ subpaths $\rho_1 = \langle v_0, \ldots, u_i \rangle$ and $\rho_2 = \langle v_{i+1}, \ldots, u_k \rangle$. Applying Lemma 26 to ρ_1 with $p \geq 0, q = s$ we get $\#(+,+)_{\rho_1} \leq \#(-,-)_{\rho_1} + s$. Similarly for ρ_2, with $p = s - 1, q = 0$, we get $\#(+,+)_{\rho_2} \leq \#(-,-)_{\rho_2} - s + 1$. Thus $\#(+,+)_\rho \leq \#(-,-)_\rho$. ■

Theorem 29. *The above algorithm runs in $\mathcal{O}(n^2 m)$ time.*

Proof. By Section 2.4, the algorithm runs in $\mathcal{O}(m')$ time, where m' is the number of edges in the transformed instance. Each edge in the original instance is copied $\mu_H \mu_R$ times in the transformed instance. Hence

$$
\begin{aligned}
m' &= \mu_H \mu_R m \\
&= \left(\sum_{h \in H} q^-(h) + 2 \right) \left(\sum_{r \in R} q^-(r) + 2 \right) m \\
&\leq (|\mathcal{R}| + 2)(\sum_{r \in \mathcal{R}} q^+(r) + 2) m && \text{. . . by Section 2.4} \\
&\leq (|\mathcal{R}| + 2)^2 m && \text{. . . since } q^+(\mathcal{R}) = \{1\} \\
&= \mathcal{O}(n^2 m)
\end{aligned}
$$

■

Chapter 4

LCSM in Students-Courses

Laminar classifications in the SC setting have been studied via matroid constraints in [10], where they give an algorithm for the 2LCSM problem in SC, i.e. for class-stability under *both-sided* laminar classifications in SC. In our application to SCLQ, we only require classifications on one side, so we have employed a simpler approach. Our approach is similar to the algorithm given in [3] for LCSM in HR, where the hospitals impose laminar classifications over the residents.

First, we show that LCSM in SC reduces to LCSM in HR. Then we shall simplify it for a special case, which will improve the time complexity of our application of LCSM to SCLQ.

4.1 Reduction to LCSM in HR

Given an LCSM instance with students $\mathcal{S}$, courses $\mathcal{C}$, preferences $\mathbf{pref}$, and classes $\mathscr{C}$ imposed by the courses on students, we build the following instance of LCSM in HR, with new preference lists $\mathbf{pref}'$.

- $\mathcal{H} = \{h_u \mid u \in \mathcal{C}\}$. $\forall u \in \mathcal{C}, q^+(h_c) = q^+(c)$.
- $\mathcal{R} = \{r_v^1, \ldots, r_v^{q^+(v)} \mid v \in \mathcal{S}\}$. $\forall r_v^i \in \mathcal{R}, q^+(r_v^i) = 1$.
- $\forall v \in \mathcal{S}, i \in [q^+(v)], \mathbf{pref}'(r_v^i) = h_{u_1} \ \ldots \ h_{u_k}$, where $\mathbf{pref}(v) = u_1 \ \ldots \ u_k$.
- $\forall u \in \mathcal{C}$, $\mathbf{pref}'(h_u) = r_{v_1}^1 \ \ldots \ r_{v_1}^{q^+(v_1)} \ \ldots \ r_{v_k}^1 \ \ldots \ r_{v_k}^{q^+(v_k)}$, where $\mathbf{pref}(u) = v_1 \ \ldots \ v_k$.
- $\forall u \in \mathcal{C}, C \in \mathscr{C}_u$, $\mathscr{C}_{h_u} = \mathscr{C}_{h_u}^{old} \cup \mathscr{C}_{h_u}^{new}$, where
  - If $C \in \mathscr{C}_u, C' = \{r_v^i \mid v \in C, i \in [q^+(v)]\} \in \mathscr{C}_{h_u}^{old}$ with quotas $q_{h_u}^-(C') = q_u^-(C), q_{h_u}^+(C') = q_u^+(C)$.
  - $\forall v \in \mathcal{S}, C' = \{r_v^i \mid i \in [q^+(v)]\} \in \mathscr{C}_{h_u}^{new}$, with quotas $q_{h_u}^+(C') = 1, q_{h_u}^-(C') = 0$.

First, we prove that the above is indeed an instance of LCSM in HR.

Lemma 30. *$\forall h_u \in \mathcal{H}, \mathscr{C}_{h_u}$ is laminar.*

Proof. Suppose not, for some h_u. Then $\exists C_1', C_2' \in \mathscr{C}_{h_u}$, such that $C_1' \setminus C_2', C_2' \setminus C_1', C_1' \cap C_2'$ are all non-empty. We consider the following cases:

Case 1. $C_1', C_2' \in \mathscr{C}_{h_u}^{old}$. The corresponding original classes be $C_1, C_2 \in \mathscr{C}_u$. Then due to our construction, we are done by laminarity of $\mathscr{C}_u$.

Case 2. $C_1', C_2' \in \mathscr{C}_{h_u}^{new}$. But by our construction, $\mathscr{C}_{h_u}^{new}$ is a partition of $\mathscr{C}_{h_u}$.

Case 3. (Wlog) $C_1' \in \mathscr{C}_{h_u}^{old}, C_2' \in \mathscr{C}_{h_u}^{new}$. But then, by our construction, either $C_2' \subseteq C_1'$, or $C_2' \cap C_1' = \emptyset$.

■

Now, suppose M' is a class-stable matching in the above transformed instance. We define M in the original instance as

$$M = \{(v, u) \mid \exists i \in [q^+(v)], (r_v^i, h_u) \in M'\}$$

Lemma 31. *M is feasible in the original instance.*

Proof. For a student v, consider $M(v)$ and $M'(\mathcal{R}_v)$, where $\mathcal{R}_v = \{r_v^1, \ldots, r_v^{q^+(v)}\}$. Consider the map $t : M'(\mathcal{R}_v) \to M(v)$, where $t(h_u) = u$. This map is surjective by construction of M, and is injective by construction of $\mathscr{C}_{h_u}^{new}$. Thus $|M(v)| = |M'(\mathcal{R}_v)| \leq |\mathcal{R}_v| = q^+(v)$.
For a course u, we can consider $q^+(u)$ to simply be $q_u^+(E(u))$. Hence, we shall proceed by proving feasibility for the class quotas. Consider a class $C \in \mathscr{C}_u$, and the corresponding class $C' \in \mathscr{C}_{h_u}^{old}$. Consider the map $t : |M'(h_u) \cap C'| \to M(u) \cap C$ where $t(r_v^i) = v$. This map is surjective by construction of M, and injective by construction of $\mathscr{C}_{h_u}^{new}$. Thus $|M(u) \cap C| = |M'(h_u)| \geq q_{h_u}^-(C') = q_u^-(C)$, and similarly for the upper quota. ■

Lemma 32. *M is class-stable in the original instance.*

Proof. Suppose not, i.e. there is a feasible blocking pair (u, v) in M. Then $\forall i \in q^+(v), r_v^i \notin M'(h_u)$, and:

- Either u is undersubscribed in M, which by our maps in the proof of Lemma 31, means that h_u is undersubscribed in M'; or $\exists u' \in M'(v), u' <_v u$, so by construction, $\exists i_1, h_{u'} \in M(r_v^{i_1}), h_{u'} <_{r_v^{i_1}} h_u$.
- Either v is undersubscribed in M, which by above maps means that $\exists i, r_v^i$ is unmatched in M'; or $\exists v' \in M(u), v' <_u v$, so by construction, $\exists i_2, r_{v'}^{i_2} \in M'(h_u), \forall i \in [q^+(v)], r_{v'}^{i_2} <_{h_u} r_v^i$, in particular for $i = i_1$. Thus $(h_u, r_v^{i_1})$ is a blocking pair in M'.

Also $(u, v) \notin M$, so $\forall i \in q^+(v), r_v^i \notin M'(h_u)$. So where for any class $C' \ni r_v^{i_1}$, $M'(h_u) \cap C = \emptyset$, where C. So $(h_u, r_v^{i_1})$ is a feasible blocking pair. ■

Lemma 33. *If there is a class-stable matching in the original instance, then there is a class-stable matching in the transformed instance.*

Proof. Let the matching N be class-stable in the original instance. Construct N' as follows : $\forall v \in \mathcal{S}$, if $N(v) = \{u_1, \ldots, u_k\}$, then include the edges $(h_{u_1}, r_v^1), \ldots, (h_{u_k}, r_v^k)$ in N'. This is possible since N is a matching, so $k \leq q^+(v)$.
Suppose there is a feasible blocking pair $(h_u, r_v^{i_1}) \in N'$. Then :

- By construction above, h_u gets exactly one copy of each student in $N(u)$. So either h_u is undersubscribed in N', i.e. u is undersubscribed in N; or $\exists r_{v'}^{i_2} \notin N'(h_u), r_{v'}^{i_2} <_{h_u} r_v^{i_1}$, i.e. $v' <_u v$.
- If $r_v^{i_1}$ is unmatched in N', then by construction, v is undersubscribed in N. Else, $\exists h_{u'}, h_{u'} \in M(r_v^{i_1}), h_{u'} <_{r_v^{i_1}} h_u$, so $u' <_v u$.

So (u, v) is a blocking pair in N. By $\mathscr{C}_{h_u}^{new}$, we know $v' \neq v$. Now consider a class C in $\mathscr{C}_u$, containing v but not v', such that $|N \cap C| = q_u^+(C)$. Then the corresponding $C'\mathscr{C}_{h_u}^{old}$ contains $r_v^{i_1}$ but not $r_v i_2$, and by our maps in the proof of Lemma 31, $|N' \cap C'| < q_u^+(C) = q_{h_u}^+(C')$, which contradicts our assumption. ■

Theorem 34. *LCSM in SC reduces to LCSM in HR. The running time increases by a factor of $\mathcal{O}(m^2)$.*

Proof. By Lemma 30, the reduction above is indeed to LCSM in HR. Lemma 33 and Lemma 32 show that the original instance has a class-stable solution iff the new instance has a class-stable solution, which the above algorithm finds. By Lemma 31, the matching that is found is feasible, i.e. indeed a solution, for the original constraints.
The above reduction can be carried out in time $\mathcal{O}(m')$, where m' is the new number of edges, since the length of the new preference lists is $2m'$. Running Huang's algorithm in the new instance takes time $\mathcal{O}(m'^2)$. For each edge $(v, u) \in \mathcal{S} \times \mathcal{C}$, we make $q^+(v)$ copies of it. Hence

$$\begin{aligned} m' &= \sum_{v \in \mathcal{S}} |E(v)|\, q^+(v) \\ &\leq \sum_{v \in \mathcal{S}} |E(v)|^2 \\ &\leq (\sum_{v \in \mathcal{S}} |E(v)|)^2 \\ &= m^2 \end{aligned}$$

So we obtain a running time of $\mathcal{O}(m^4)$. ■

The above running time is already poor, and as we see in Chapter 5, combined with the reduction from SCLQ to LCSM in SC, we obtain a worst-case running time of $\mathcal{O}(m^8)$. Fortunately, the worst-case instances of LCSM shall not arise for our purpose. Moreover, we shall avoid the blowup in reducing to HR by emulating the proposal scheme of the reduced instance in the original instance itself. Below, we use this to design a simpler adaptation of Huang's algorithm to the SC setting, which suffices for our application to SCLQ.

4.2 Special Case for SCLQ

For our purpose, less constraints will suffice : we only need partition classifications, and we do not need lower quotas on classes. We can also use the same classification across all courses, but we shall not utilise that restriction here. The following is the problem that we need to solve for our application.

PARTITION CLASSIFIED STABILITY IN STUDENTS-COURSES
Input : A bipartite graph $G = (\mathcal{S} \uplus \mathcal{C}, E)$, and for each $v \in \mathcal{S} \uplus \mathcal{C}$:

1. ordered preference lists $\mathbf{pref}(v) = (E(v), <_v)$
2. upper quotas $q^+ : \mathcal{S} \uplus \mathcal{C} \to \mathbb{N}$, such that for each $C \in \mathscr{C}_u, q_u^-(C) \leq q_u^+(C)$, and $\forall v \in \mathcal{S} \uplus \mathcal{C}, q^+(v) \geq 0$.

and for each course $u \in \mathcal{C}$, a classification $\mathscr{C}_u \subseteq 2^{E(u)}$, with upper quotas $q_u^+ : \mathscr{C}_u \to \mathbb{N}$, such that the sets in $\mathscr{C}_u$ are pairwise disjoint.
Output : A matching $M \subseteq E$, such that

1. M is feasible for the above input.
2. M is class-stable.

Our natural attempt to solve this problem is to follow the techniques used in [3]. For the above question, this gives us the following modification to the Gale-Shapley algorithm:

1. The students propose to courses, in decreasing order of preference, as long as they are undersubscribed, or until they exhaust their preference list.
2. When a course u receives a proposal from a student v belonging to class C^v, it proceeds as follows:
   - 2.1. If u is undersubscribed and has less than $q_v^+(C^v)$ students in the class C^v, u accepts.
   - 2.2. If u has $q_v^+(C^v)$ students in the class C^v, it compares v to the least preferred such student, and keeps the more preferred one.
   - 2.3. If u has less than $q_v^+(C^v)$ students in C^v but is full overall, then it proceeds like Gale-Shapley, i.e. it compares u to the overall least preferred existing student, keeping the more preferred one.

Each edge is used for at most one proposal, so this algorithm terminates. Let the matching obtained be M. We now prove class-stability. The key idea is that the second condition above cannot worsen the worst partner of a course, either overall or within a class; and the third condition above can worsen its worst partner within a class, but not overall. We elaborate below.

Theorem 35. *M is class-stable.*

Proof. Suppose there is a feasible blocking pair with respect to M. Let this be (u, v).
By the blocking condition for v and the algorithm above, v must have proposed to u and got rejected. Suppose this rejection was due to the overall quota, i.e. due to condition 2.3. in the above algorithm. Then u was full at the time of rejection, and the number of partners of u never decreases since it only swaps partners. So by the blocking condition for u, $\exists v' \in M(u), v' <_u v$. But v was the least preferred partner at the time of rejection, and u was full. We observe that once u is full, since every subsequent rejection is in favour of a better partner, its least preferred partner can only improve. So if the least preferred student in $M(u)$ is v'', then $v' \geq_u v'' >_u v$, which is a contradiction.
Hence the rejection of v was due to the class quota, i.e. due to condition 2.2. in the above algorithm. So at the time of rejection, u had $q_v^+(C^v)$ students in C^v. Consider the rejections after this that affect the partners of u in C^v. If all these rejections are due to condition 2.2., then C^v remains full for u, and rejections in C^v are in favour of a better student in C^v itself, so similarly to the earlier case, we arrive at a contradiction. Hence consider the first rejection in C^v, after the rejection of v, that occurs due to condition 2.3., and let this rejected student be v''. At this time, v'' was the least preferred partner for u overall, and hence within C^v. By our choice of the first such rejection, until this rejection the least preferred partner for u in C^v would not have worsened, so $v'' >_u v$. But then, by the argument we made for the first case, $v' >_u v'' >_u v$, which is again a contradiction. ■

Since the above algorithm has at most one proposal per edge and at most two comparisons per proposal, it runs in $\mathcal{O}(m)$ time.

Chapter 5

Popularity in SCLQ

Popularity in Students-Courses with One-Sided Lower Quotas

Input : A bipartite graph $G = (\mathcal{S} \uplus \mathcal{C}, E)$, and for each $v \in \mathcal{S} \uplus \mathcal{C}$:

1. ordered preference lists $\textbf{pref}(v) = (E(v), <_v)$
2. upper and lower quotas $q^+, q^- : \mathcal{S} \uplus \mathcal{C} \to \mathbb{N}$, such that $\forall v \in \mathcal{S} \uplus \mathcal{C}, q^-(v) \leq q^+(v)$, and $q^-(\mathcal{C}) = \{0\}$

Output : A subset $M \subseteq E$ such that :

1. $\forall v \in \mathcal{S} \uplus \mathcal{C}, q^-(v) \leq |M(v)| \leq q^+(v)$
2. M is popular among all matchings having the previous property.

Since we closely follow [2], we try to simulate repeated proposals with priorities via copies and dummies. However, we note that in the SC setting, we need to prevent duplicate edges in the transformed instance, i.e. the allocation of the same course to different copies of the same student. As we have seen in Lemma 22, the higher priorities only match vertices up to their lower quotas. If a student's lower quota is "filled" in the transformed instance using multiple allocations of the same course, then the matching obtained in the original instance will not be feasible for that student. To prevent this behaviour, we classify student-copies according to their original student, placing a unit upper quota on every such class. We will then show that the slightly weaker notion of stability under classifications, which we call class-stability, gives us the desired solution.

We set up the transformed instance exactly as it is done for HRLQ in [2]. The underlying idea, once again, is that students undersubscribed in the first round of proposals will propose again, and then if still deficient, will propose some more times. As before, this is implemented using levels and dummies. So we have

$$\mu = \sum_{v \in \mathcal{S}} q^-(v) + 2$$

We create copies $v^0, \ldots, v^{\mu-1}$ for each student v. For each copy v^s:

$$q^+(v^s) = \begin{cases} q^+(v) & s \in \{0,1\} \\ q^-(v) & \text{otherwise} \end{cases}$$

For each $s \neq \mu - 1$, we add dummy courses $\mathcal{D}_{v^s} = \{d^i_{v^s} \mid i \in q^+(v^s)\}$, and call the collection of all dummies $\mathcal{D}$. The preference lists are as follows. For each student v:

$$\begin{aligned}
v^{\mu-1} &: d^1_{v^{\mu-2}} \ \ldots \ d^{q^-(v)}_{v^{\mu-2}} \ \mathbf{pref}(v) \\
\forall 2 < s < \mu - 1, v^s &: d^1_{v^{s-1}} \ \ldots \ d^{q^-(v)}_{v^{s-1}} \ \mathbf{pref}(v) \ d^1_{v^s} \ \ldots \ d^{q^-(v)}_{v^s} \\
v^2 &: d^{q^+(v)-q^-(v)+1}_{v^1} \ \ldots \ d^{q^+(v)}_{v^1} \ \mathbf{pref}(v) \ d_{v^2} \ \ldots \ d^{q^+(v)}_{v^2} \\
v^1 &: d^1_{v^0} \ \ldots \ d^{q^+(v)}_{v^0} \ \mathbf{pref}(v) \ d^1_{v^1} \ \ldots \ d^{q^+(v)}_{v^1} \\
v^0 &: \mathbf{pref}(v) \ d^1_{v^0} \ \ldots \ d^{q^+(v)}_{v^0}
\end{aligned}$$

and for each course u, where $\mathbf{pref}(u)^s$ is the s-level copy of $\mathbf{pref}(u)$:

$$u : \mathbf{pref}(u)^{\mu-1} \ \ldots \ \mathbf{pref}(u)^0$$

For the dummy courses, the preferences are:

$$\begin{aligned}
\forall i \in [q^+(v)], d^i_{v^0} &: v^0 \ v^1 \\
\forall i \in [q^+(v) - q^-(v)], d^i_{v^1} &: v^1 \\
\forall i \in [q^+(v)] \setminus [q^+(v) - q^-(v)], d^i_{v^1} &: v^1 \ v^2 \\
\forall i \in [q^-(v)], 2 \leq s < \mu - 1, d^i_{v^s} &: v^s \ v^{s+1}
\end{aligned}$$

Then, we add the following class quotas : where

$$\forall v \in \mathcal{S}, C^v = \{v^0, \ldots, v^{\mu-1}\}$$

each course u imposes the upper quota $q^+_u(C^v) = 1$ on the class C^v.

Let a class-stable matching on this instance be M'. We obtain the solution to the original instance as follows :

$$M = \{(u, v) \mid (u, v^s) \in M'\}$$

Now we must prove that M is indeed maximum-cardinality popular among feasible matchings in the original instance. Our class upper quotas ensure feasibility by avoiding the overcounting we mentioned earlier. For the remaining proof, we shall closely follow the results proved for HR2LQ above, setting lower quotas for courses to zero.
The earlier results hold here if:

- they do not depend on the unit capacity of residents
- they hold under class-stability, which is weaker than stability.

We notice that the main results, Theorems 23, 27 and 28 do not directly appeal to the above conditions. So they will hold here if we can prove the lemmas that lead to those theorems. We shall use the same definitions of V^j, U_j and *active* vertices.
We observe that certain kinds of blocking pairs that arise in our proofs are always feasible blocking pairs.

1. Where the course in the pair is a dummy. This is because dummies don't classify student-copies.

2. Where the blocking condition compares two copies of the same student. This is because then they are in the same class, which satisfies the additional constraint for a feasible blocking pair.

3. Where the pair (v^s, u^t) is such that $v \notin M(u)$, because then $M'(u^t) \cap C^v$ is empty.

Now we work through the desired results.

- Lemma 15 does not depend on residents or stability, hence it directly holds here.

- Lemma 17 does not depend on the unit capacity of residents, and we observe that the only blocking pairs that arise are as in observations 1 and 2.

- Corollary 18 holds by Lemma 17.

- Lemma 20 depends on a blocking pair of the form (v^{s-1}, u^t). But by hypothesis, $u \notin M(v)$. So this is a feasible blocking pair by observation 3 above.

- Corollary 21 holds by Lemma 20 and Corollary 18.

- Parts 1,2, and 3 of Lemma 22 hold by our above observations. Part 4 earlier required $V = \mathcal{H}$. We notice that the only difference between $\mathcal{H}$ and $\mathcal{R}$ was the upper quota at priority 1. Here $\mathcal{S}$ is given the full upper quota at priority 1, like $\mathcal{H}$ earlier. Hence, again by our above observations, Part 4 holds for $V = \mathcal{S}$, which suffices for our subsequent proofs.

For the remaining results, since we reason within $M \oplus N$, any blocking pairs (v^s, u^t) that arise are such that $v \notin M(u)$. So again this is a feasible blocking pair by observation 3 above.

Thus our algorithm gives the desired matching. We calculate its running time.

Theorem 36. *The above algorithm runs in $\mathcal{O}(m^2)$ time.*

Proof. By Section 2.4, this algorithm takes $\mathcal{O}(m')$ time, where m' is the number of edges in the transformed instance. Only copies of students are made, so each edge is copied μ times, and trivially infeasible instances can be ignored as mentioned in Section 2.4. Hence $m' = \mu m$, and we get

$$\begin{aligned}\mu m &= (\sum_{v \in \mathcal{S}} q^{-}(v) + 2)m \\ &\leq (\sum_{v \in \mathcal{C}} q^{+}(v) + 2)m \\ &\leq (\sum_{v \in \mathcal{C}} |E(v)| + 1)m \\ &= \mathcal{O}(m^2)\end{aligned}$$

■

Chapter 6

Further Generalisation

In our attempts to extend the above algorithms to Popularity in SC2LQ, i.e. both-sided lower quotas in Student-Courses, we have come across some interesting questions.

The natural approach towards popularity in SC2LQ would be to repeat the technique that we used for HR2LQ, i.e. to give higher priorities to deficient vertices on both sides, which we would simulate using copies and dummies. As before, we need to ensure that for an edge (u, v) in the original instance, a copy u^s in the transformed instance matches with at most one copy of v, and vice-versa. As observed for SCLQ, the upper quota condition proved in Lemma 15 does not suffice to ensure this when copies are made in a many-to-many setting. For our solution in SCLQ, we used classifications imposed by courses on the student-copies, which suggests that using class quotas on both sides could be useful for extending it to SC2LQ. We shall formalise this method below, and exhibit why it does not suffice, along with an interesting property of the hurdle that arises. We shall also suggest two possible approaches to overcome this obstacle.

6.1 Both-sided Classifications

Class-stability under both-sided laminar classifications (2LCSM) in the SC setting has been studied in [10]. This problem can be stated as follows :

> Both-sided Laminar Classified Stability in Students-Courses
> **Input :** A bipartite graph $G = (\mathcal{S} \uplus \mathcal{C}, E)$, and for each $v \in \mathcal{S} \uplus \mathcal{C}$:
>
> 1. ordered preference lists $\mathbf{pref}(v) = (E(v), <_v)$
> 2. upper quotas $q^+ : \mathcal{S} \uplus \mathcal{C} \to \mathbb{N}$
>
> and for each course $u \in \mathcal{S} \uplus \mathcal{C}$, a classification $\mathscr{C}_u \subseteq 2^{E(u)}$, with upper and lower quotas $q_u^+, q_u^- : \mathscr{C}_u \to \mathbb{N}$, such that for each $C \in \mathscr{C}_u, q_u^-(C) \leq q_u^+(C)$.

Similar to what we stated for LCSM in Section 2.3, the blocking condition under 2LCSM is complicated, but in the absence of lower quotas, we can simply use a modification to the blocking pair.

Definition 37. *$(v, u) \in \mathcal{S} \times \mathcal{C}$ is said to be a* feasible blocking pair *with respect to a matching M if it is a blocking pair, and the following conditions hold:*

- *For any class $C_u^v \in \mathscr{C}_u$ containing v, either $|M(u) \cap C_u^v| < q_u^+(C_u^v)$ or $\exists v' \in M(u) \cap C_u^v, v' <_u v$.*
- *For any class $C_v^u \in \mathscr{C}_v$ containing u, either $|M(v) \cap C_v^u| < q_v^+(C_v^u)$ or $\exists u' \in M(v) \cap C_v^u, u' <_v u$*

As before, a *class-stable* matching does not contain any feasible blocking pairs, and a matching respecting all quotas is said to be *feasible*. This allows us to formalise a version of our problem where there are no lower quotas. Also, once again, we require only partition classes to formalise our application. Hence, a special case of Both-sided Partition Classified Stability (2PCSM) suffices for us. We state this version below.

Both-sided Partition Classified Stability in Students-Courses
Input : A bipartite graph $G = (\mathcal{S} \uplus \mathcal{C}, E)$, and for each $v \in \mathcal{S} \uplus \mathcal{C}$:

1. ordered preference lists $\mathbf{pref}(v) = (E(v), <_v)$
2. upper quotas $q^+ : \mathcal{S} \uplus \mathcal{C} \to \mathbb{N}$

and for each course $u \in \mathcal{S} \uplus \mathcal{C}$, a classification $\mathscr{C}_u \subseteq 2^{E(u)}$, with upper quotas $q_u^+ : \mathscr{C}_u \to \mathbb{N}$, such that the sets in $\mathscr{C}_u$ are pairwise disjoint.
Output : A matching $M \subseteq E$, such that

1. M is feasible for the above input.
2. M is class-stable.

The approach towards 2LCSM in [10] is to construct an ordered matroid for each side, such that the independent sets arise from the class quotas, while the order arises from the preference orders of all the vertices on the respective side. They show that a kernel of the two matroids gives a class-stable matching. In our solution to SCLQ, we avoided the worst-case running time of [3], of which [10] is a generalisation, by employing a simple adaptation for our special case. We attempted to extend this adaptation here, but were unable to suitably capture classifications on the proposing side. Hence, an approach to SC2LQ using classifications can potentially give a poor running time in the worst case. A more pressing issue, however, is that it may not be correct at all.

Application to Popularity in SC2LQ
Consider an SC2LQ instance, and build a transformed instance in the same manner as we did for

HR2LQ in Chapter 4. We extend the classification scheme used for SCLQ to both sides – classes with unit upper quota are formed from the copies that share the same original vertex. As before, M' is a class-stable matching in the transformed instance. Now consider an edge (u, v) of the original instance – classifications do not prevent (u^s, v^t) and (u^{s+1}, v^{t+1}) from appearing simultaneously in M'. This is because the class quotas imposed by a vertex are independent of those imposed by other vertices – here, however, we need each copy of a vertex to "know" the status of other copies, e.g. we need v^s to avoid u^t *if* (v^{s+1}, u^{t+1}) has been chosen, and so on.

We make an interesting observation about this particular kind of duplicates, which might be useful in further work. If duplicates of the form $(v^s, u^t), (v^{s+1}, u^{t+1})$ occur, then either $s = 0$ or $t = 0$. In fact, if we consider a stable M' rather than a class-stable M', then $s = t = 0$. We shall give a sketch of this result below.

Lemma 38. *Let $(v^s, u^t) \in M'$, $s > 0, t > 0$. Then $(v^{s-1}, u^{t-1}) \in M'$.*

Proof. By Lemma 17, $\exists d_v \in M'(v^{s-1}) \cap \mathcal{D}_{v^{s-1}}$ and $d_u \in M'(u^{t-1}) \cap \mathcal{D}_{u^{t-1}}$. But $u^{t-1} >_{v^{s-1}} d_v$, and $v^{s-1} >_{u^{t-1}} d_u$, so if $(v^{s-1}, u^{t-1}) \notin M'$, it would block M'. We will show that it would also be a feasible blocking pair.
By Lemma 17, for $s' < s-1, t' < t-1$, $M'(v^{s'}), M'(u^{t'}) \subseteq \mathcal{D}$. Also if $v^{s+1} \in M'(u^{t-1})$ then (v^{s+1}, u^t) would form a feasible blocking pair, so $M'(u^{t-1}) \cap C_{u^{t-1}}^{v^{s-1}} = \emptyset$. We can argue similarly for the other side. ■

Corollary 39. *If $(v^s, u^t), (v^{s'}, u^{t'}) \in M'$, $s < s'$, then either $s = 0$ or $t = 0$. If M' is stable, then $s = t = 0$.*

Proof. Suppose not, then $s > 0$. If $t > t'$, then by Lemma 17, $(v^{s+1}, u^{t'})$ would be a feasible blocking pair. By class constraints, $t \neq t'$, so $t' > t$. But by Lemma 38, $(v^{s-1}, u^{t-1}) \in M'$, which violates Corollary 18. So $s = 0$.
If $t > 0$, then by Lemma 17, (v^0, u^0) would be a blocking pair. ■

6.2 Possible Approaches to Popularity in SC2LQ

One possible approach to avoid these "parallel" duplicate edges is to extend the notion of classifications to vertex subsets. We allow classes $\mathscr{C}_{V'}$ for $V' \in 2^V, V \in \{\mathcal{S}, \mathcal{C}\}$, such that for $C \in \mathscr{C}_{V'}$, $M'(V') \cap C$ would now have upper and lower quotas $q^+_{V'}(C), q^-_{V'}(C)$. This coincides with the existing notion of classifications in the case where every V' with non-trivial class quotas is singleton. This generalisation would allow us the following : for an edge (u, v) in an SC2LQ instance, define V'_v to be the set of all copies of v in the corresponding transformed instance, and similarly define U'_u. Impose upper quotas $q^+_{V'_v}(U'_u) = 1, q^+_{U'_u}(V'_v) = 1$, and set their lower quotas to 0. This captures our required condition, i.e. exactly one copy of the edge (u, v) can now appear in M'. If the matroid criteria in [10] are modified to capture these generalised classifications, then we would obtain an algorithm for popularity in SC2LQ. We can expect the time complexity of generalised class-stability to depend on the size and/or number of the sets that impose classes. It is possible to rephrase our required constraints to involve a larger number of smaller sets : because of Corollary 18, we can consider pairs of levels rather than all levels at once. Given an edge (u, v), for each level $s \neq 0$, we define $V'_{v^{s-1}} = \{v^{s-1}, v^s\}$, such that each of these sets imposes unit upper quota on the class U'_u of all copies of u. We similarly define sets of the form U'_{u^t} which impose unit upper quota on V'_v.

Another possibility is to avoid classifications altogether, and instead generalise the notion of *restricted edges*. Restricted edges are edges that are *forced* or *forbidden* from appearing in the solution. Since the constraints placed by vertices do not change, the definition of stability here is the original one, via blocking pairs. Cseh and Manlove have studied hardness and approximations of stability under restricted edges in [15]. We can extend those definitions to *restricted edge subsets*, such that the existing notion corresponds to all restricted subsets being singleton. Then, in the transformed instance for popularity in SC2LQ, we forbid any edge subset containing multiple copies of the same original edge, e.g. where $\bar{v} \neq v, \bar{u} \neq u$, the edge subset $\{(u^0, v^2), (u^1, v^3), (u^1, \bar{v}^1)\}$ would be forbidden, but $\{(u^0, v^2), (u^1, \bar{v}^1), (\bar{u}^0, \bar{v}^1)\}$ would not. Once again, because of Corrollary 18, we can expect that forbidding small subsets of the form $\{(v^s, u^t), (v^{s+1}, u^{t'})\}, t' \in \{t, t+1\}$, and similarly for the other side, would suffice to disallow duplicate edges. This generalisation would then give us an algorithm to either find or disprove the existence of popular matchings in SC2LQ. Corollary 39 allows us to further reduce the number of restricted subsets required, which we expect will improve the running time.

Bibliography

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

[2] Meghana Nasre and Prajakta Nimbhorkar. Popular matchings with lower quotas. In Satya V. Lokam and R. Ramanujam, editors, *37th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2017, December 11-15, 2017, Kanpur, India*, volume 93 of *LIPIcs*, pages 44:1–44:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017.

[3] Chien-Chung Huang. Classified stable matching. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '10, page 1235–1253, USA, 2010. Society for Industrial and Applied Mathematics.

[4] 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.

[5] Dan Gusfield and Robert W. Irving. *The Stable Marriage Problem: Structure and Algorithms.* MIT Press, Cambridge, MA, USA, 1989.

[6] Chien-Chung Huang and Telikepalli Kavitha. Popular matchings in the stable marriage problem. In Luca Aceto, Monika Henzinger, and Jirí Sgall, editors, *Automata, Languages and Programming - 38th International Colloquium, ICALP 2011, Zurich, Switzerland, July 4-8, 2011, Proceedings, Part I*, volume 6755 of *Lecture Notes in Computer Science*, pages 666–677. Springer, 2011.

[7] Florian Brandl and Telikepalli Kavitha. Popular matchings with multiple partners. In Satya V. Lokam and R. Ramanujam, editors, *37th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2017, December 11-15, 2017, Kanpur, India*, volume 93 of *LIPIcs*, pages 19:1–19:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017.

[8] Alvin E. Roth. Stability and polarization of interests in job matching. *Econometrica*, 52(1):47–57, 1984.

[9] David J. Abraham, Robert W. Irving, and David F. Manlove. The student-project allocation problem. In Toshihide Ibaraki, Naoki Katoh, and Hirotaka Ono, editors, *Algorithms and Computation*, pages 474–484, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[10] Tamás Fleiner and Naoyuki Kamiyama. A matroid approach to stable matchings with lower quotas. *Mathematics of Operations Research*, 41(2):734–744, 2016.

[11] Yu Yokoi. Envy-free matchings with lower quotas. *CoRR*, abs/1704.04888, 2017.

[12] Koki Hamada, Kazuo Iwama, and Shuichi Miyazaki. The hospitals/residents problem with lower quotas. *Algorithmica*, 74(1):440–465, January 2016.

[13] Ágnes Cseh and Telikepalli Kavitha. Popular edges and dominant matchings. In Quentin Louveaux and Martin Skutella, editors, *Integer Programming and Combinatorial Optimization - 18th International Conference, IPCO 2016, Liège, Belgium, June 1-3, 2016, Proceedings*, volume 9682 of *Lecture Notes in Computer Science*, pages 138–151. Springer, 2016.

[14] Yuri Faenza, Telikepalli Kavitha, Vladlena Powers, and Xingyu Zhang. Popular matchings and limits to tractability. *CoRR*, abs/1805.11473, 2018.

[15] Ágnes Cseh and David F. Manlove. Stable marriage and roommates problems with restricted edges: complexity and approximability. *CoRR*, abs/1412.0271, 2014.