The College Admissions problem with lower and common quotas

Péter Biró, Tamás Fleiner, Robert W. Irving and David F. Manlove

Department of Computing Science
University of Glasgow
Glasgow G12 8QQ
UK

Technical Report
TR-2009-303
July 2009

University of Glasgow | Department of Computing Science

The College Admissions problem with lower and common quotas

Péter Biró[1,*,†], Tamás Fleiner[2,†], Robert W. Irving[1,*] and David F. Manlove[1,*]

[1] *Department of Computing Science, University of Glasgow, Glasgow G12 8QQ, UK.*
Email: {pbiro,rwi,davidm}@dcs.gla.ac.uk.

[2] *Department of Computer Science and Information Theory,*
Budapest University of Technology and Economics, H-1117, Magyar tudósok körútja 2,
Budapest, Hungary. Email: fleiner@cs.bme.hu.

Abstract

We study two generalised stable matching problems motivated by the current matching scheme used in the higher education sector in Hungary. The first problem is an extension of the College Admissions problem in which the colleges have lower quotas as well as the normal upper quotas. Here, we show that a stable matching may not exist and we prove that the problem of determining whether one does is NP-complete in general. The second problem is a different extension in which, as usual, individual colleges have upper quotas, but in addition, certain *bounded* subsets of colleges have common quotas smaller than the sum of their individual quotas. Again, we show that a stable matching may not exist and the related decision problem is NP-complete. On the other hand we prove that, when the bounded sets form a nested set system, a stable matching can be found by generalising, in non-trivial ways, both the applicant-oriented and college-oriented versions of the classical Gale-Shapley algorithm. Finally, we present an alternative view of this nested case using the concept of choice functions, and with the aid of a matroid model we establish some interesting structural results for this case.

1 Introduction

The *College Admissions* (or *Hospitals / Residents*) problem was introduced by Gale and Shapley [11]. They gave a linear time algorithm that always finds a stable matching. Roth [21] discovered that the very same method had already been implemented in 1952 in the National Resident Matching Program, the centralised matching scheme that coordinates junior doctor recruitment in the US. Since then, similar matching schemes have been organised in many countries to allocate graduating medical students to hospital posts (hence the alternative name for the problem), and these matching schemes are widely used for other professions as well. Gusfield and Irving [14] and Roth and Sotomayor [24] provide the classical results and background material for this problem.

Regarding the original context, the Gale-Shapley algorithm is also used in handling higher education admissions in a number of countries, including Spain [20], Turkey [4] and Hungary [5] (whilst a different method is used for medicine and related subjects in

*Supported by EPSRC grant EP/E011993/1.
†Supported by OTKA grant K69027.

Germany [6]). Moreover, the same kind of admission systems have been introduced for secondary schools in, amongst others, Boston [2], New York [1] and again Hungary [5].

We define the College Admissions problem (CA) as follows. We are given a bipartite graph $G(A \cup C, E)$, where $A = \{a_1, a_2, \dots, a_n\}$ is the set of applicants and $C = \{c_1, c_2, \dots, c_m\}$ is the set of colleges. If an applicant a_i applies to a college c_j and c_j finds a_i acceptable then $(a_i, c_j) \in E$. Every applicant has a strict order of preference over the colleges she applies to, and every college also has a strict order of preference over its acceptable applicants, so each applicant a_i and each college c_j has a *preference list*, denoted by $P(a_i)$ and $P(c_j)$, respectively. Furthermore, each college c_j has a positive integer *quota* $q(c_j)$, this being the maximum number of applicants that it can accept.

A *matching* is a set of edges in the graph, such that every applicant is matched to at most one college and no college has more assignees than its quota. For a matching M, if an applicant a_i is matched in M then let $M(a_i)$ denote the college to which a_i is assigned, and let $M(c_j)$ denote the set of applicants assigned to college c_j. A matching is *stable* if, for every acceptable applicant-college pair (a_i, c_j) that is not in the matching, either a_i is matched and she prefers $M(a_i)$ to c_j, or c_j has a full quota of applicants that it prefers to a_i. In this case, we say that the pair (a_i, c_j) is *dominated* by M. An applicant and a college form a *blocking pair* if it is not dominated by the matching.

Admission to higher education establishments in Hungary has been organised by a centralised matching scheme since 1985. The applicants submit a preference list over the particular courses of study they are interested in. In addition to the name of the institution and the field of study, they have to indicate whether they are applying for state financed or privately financed study[1], but here, we consider these entities simply as distinct *colleges* for simplicity.

Basically, it is the Gale-Shapley algorithm that produces a kind of stable matching every year by matching more than 100,000 students. However, there are at least three special features in the scheme that are also interesting in a theoretical sense.

The first feature, which was studied in [5], is the presence of *ties* in the system. The colleges rank their applicants according to their academic scores and/or their results in the entrance exams; the solution is based on the concept of so-called *stable score-limit*. This is equivalent to a stable matching if there are no ties, i.e., if two students cannot have the same score at the same college.

The second feature, which is the subject of Section 2 of this paper, is the condition of *lower quotas*. In addition to upper quotas, which are part of the classical College Admissions problem, here, every college may have a lower quota as well, that is the minimum number of students that must be admitted if the college is to be open. We will show that a stable matching may not exist in this case; moreover, the problem of deciding whether a stable matching exists is NP-complete in general. We also mention some relaxed versions of this problem, and we describe the current heuristics used in the Hungarian application. Here we note that in a recent paper [15], Hamada *et al.* studied a related problem with similar motivations. The difference between their model and ours is that they assume that each applicant has a complete preference list (i.e., the underlying bipartite graph is complete) and they also require the matching to satisfy all lower quotas (i.e., no college can be closed in their model). They proved that the problem of finding a matching with the minimum number of blocking pairs under these conditions is NP-hard (moreover it is not approximable within $n^{1-\varepsilon}$ for any positive constant ε).

[1] An applicant may rank first a state financed course of study in economics at university A, then secondly another state financed course of study in economics at university B, and thirdly a privately financed course of study in economics at the first university A. So the fee status is included in the preferences of the applicants in this way.

The third feature, studied in Sections 3, 4 and 5, is the problem of *common quotas*. In this case, in addition to the individual quotas of the colleges, particular sets of colleges can have common quotas smaller than the sum of their individual quotas. This means that the number of students assigned to colleges in a given set cannot exceed the common quota for that set. A solution is said to be stable if, for every acceptable applicant-college pair (a_i, c_j), if a_i is not admitted to c_j then either a_i is admitted to a preferred college, or c_j has a full quota of better applicants, or there is a set of colleges C_k such that $c_j \in C_k$ and the common quota of C_k is filled by better applicants than a_i. (This implies that all of the colleges in the set C_k judge the students on the same basis – i.e., have preference lists that are consistent.) In Section 3, we show that a stable matching may not exist under these conditions, and that it is NP-complete to determine whether one does. On the other hand, we show in Section 4 that, for nested set systems, the problem becomes solvable in polynomial time. We describe applicant-oriented and college-oriented Gale-Shapley type algorithms that find stable matchings, which we prove to be optimal / pessimal for the applicants and colleges, respectively. Finally, in Section 5, we study the problem from the perspective of *choice functions* and present some results on the structure of stable matchings that follow from an appropriate matroid model.

2 Lower quotas

In Hungary, higher education institutions can declare lower quotas for each of their particular areas of study. If the number of assigned students is less than this quota for a particular area then the course has to be cancelled for that year. In general, this creates an interval that ensures a reasonable number of students every year in order to make the course viable. For some specialist areas of study this lower quota may be very small. In this section we consider the complexity of the College Admissions problem with lower quotas. We prove that for a given instance of this problem, a stable matching need not exist and deciding whether one does exist is NP-complete.

2.1 Problem definition

Suppose we are given an instance of the College Admissions problem in which, in addition to its (upper) quota, each college $c_j \in C$ has a *lower quota* $l(c_j)$. In the presence of lower quotas, to avoid ambiguity we refer to $q(c_j)$ as defined previously as c_j's *upper quota*, also denoting $q(c_j)$ by $u(c_j)$. We assume that $l(c_j) \leq u(c_j)$ for all $c_j \in C$.

A *matching* in this context requires that every college c_j satisfies $|M(c_j)| = 0$ or $l(c_j) \leq |M(c_j)| \leq u(c_j)$. We say that c_j is *closed* if $|M(c_j)| = 0$, and *open* otherwise. If $|M(c_j)| < u(c_j)$ then we say that c_j is *undersubscribed*, otherwise, if $|M(c_j)| = u(c_j)$ then we say that c_j is *full*.

A matching is *stable* if the following two conditions are satisfied:

a) (*no blocking pair*) there is no open college c_j and applicant a_i such that $(a_i, c_j) \in E$, c_j is either undersubscribed or prefers a_i to a member of $M(c_j)$, and a_i is either unmatched or prefers c_j to $M(a_i)$;

b) (*no blocking coalition*) there is no closed college c_j (*blocking college*) and a set of $l(c_j)$ applicants, each of whom is either unmatched (and finds c_j acceptable) or prefers c_j to her assigned college.

Let CA-LQ denote the problem of deciding whether an instance of the College Admissions problem with lower quotas admits a stable matching.

For a given instance of CA, each college admits the same number of applicants in all stable matchings, by the so-called "Rural Hospitals" theorem, first proved by Gale and Sotomayor [12]. Therefore, if every college achieves its lower quota for an instance of CA, then the set of stable matchings remains the same in the corresponding CA-LQ instance as well. To show this, it is enough to see that a college that achieves its lower quota in a stable matching for the CA instance would be a blocking college, if closed, in any matching that has no blocking pair for the corresponding CA-LQ instance. However, if some college does not achieve its lower quota in the stable matchings for the CA instance then a stable matching may not exist for the CA-LQ instance, as we show by the following example.

2.2 An unsolvable instance

Example 1

We have two applicants a_1, a_2 and two colleges c_1, c_2, whose preference lists are given below. In this example, and henceforth throughout this section of the paper, in the preference list of a given college c_j, the integers following the first and second colons are $l(c_j)$ and $u(c_j)$ respectively.

$$
\begin{array}{ll}
P(a_1) : c_1 \; c_2 & \qquad P(c_1) : 2 : 2 : a_1 \; a_2 \\
P(a_2) : c_2 \; c_1 & \qquad P(c_2) : 1 : 1 : a_1 \; a_2
\end{array}
$$

Here, one college must be closed, since we have only two applicants and the sum of the lower quotas is three. If c_2 is closed and $M(c_1) = \{a_1, a_2\}$ then (a_2, c_2) is a blocking coalition. Suppose now that c_1 is closed. If $M(c_2) = \{a_2\}$ then (a_1, c_2) is a blocking pair. Otherwise, if $M(c_2) = \{a_1\}$ then c_1 is a blocking college with $\{a_1, a_2\}$.

2.3 Complexity results

We now prove that CA-LQ is NP-complete. To do so, we reduce from the NP-complete problem COM SMTI [18]. This problem is defined as follows: we are given an instance I of the stable marriage problem [11] with n men and n women, where preference lists may include ties and may be incomplete (i.e., a given person may not find all members of the opposite sex to be acceptable). In this context a matching is a set of mutually acceptable (man,woman) pairs such that no person appears in more than one pair. A matching M is *stable* if there is no mutually acceptable (man,woman) pair, each of whom is unmatched or prefers the other to his/her partner in M. The question is whether I admits a *complete* stable matching, i.e., a stable matching of size n. COM SMTI is NP-complete, even if each man's list is strictly ordered, and each woman w_j's list is either strictly ordered or is a tie comprising two men, at least one of whom ranks w_j in first place [18, Theorem 2].

Theorem 1. CA-LQ *is NP-complete, even if no upper quota exceeds 3.*

Proof. The problem is clearly in NP. To show NP-hardness, we reduce from the NP-complete restriction of COM SMTI as described above. Let I be an instance of this problem, in which $U = \{m_1, \ldots, m_n\}$ is the set of men, and $W = \{w_1, \ldots, w_n\}$ is the set of women. Let $W_0 \subseteq W$ be the set of women whose preference list in I is a tie of length 2. We create an instance I' of CA-LQ as follows. Every man in I corresponds to an applicant in I', and every woman in $W \backslash W_0$ corresponds to a college with lower and upper quota 1 in I'; the preference lists of these entities in I' are initially identical to the corresponding preference lists in I. Now let $w_j \in W_0$ and suppose that the two men $m_{j,1}$ and $m_{j,2}$ are tied in w_j's list in I. In I' we create two colleges w_j^1 and w_j^2, and two additional applicants a_j^1 and a_j^2.

The preference lists of these applicants and colleges, and the lower and upper quotas of these colleges, are as follows:

$$P(a_j^1) : w_j^2 \ w_j^1 \qquad P(w_j^1) : 3 : 3 : a_j^1 \ m_{j,1} \ a_j^2$$
$$P(a_j^2) : w_j^1 \ w_j^2 \qquad P(w_j^2) : 3 : 3 : a_j^2 \ m_{j,2} \ a_j^1$$

Finally, we replace w_j by w_j^1 in $m_{j,1}$'s list in I', and similarly we replace w_j by w_j^2 in $m_{j,2}$'s list in I'. We claim that I has a complete stable matching if and only if I' has a stable matching.

For, suppose that I has a complete stable matching M. Create a matching M' in I' as follows. Initially let $M' = M$. Suppose $(m_i, w_j) \in M'$ for some woman $w_j \in W_0$, where $m_i = m_{j,r}$ for some $r \in \{1, 2\}$. Replace (m_i, w_j) in M' by (a_j^1, w_j^r), (a_j^2, w_j^r) and $(m_{j,r}, w_j^r)$. It may be verified that M' is a stable matching in I'.

Conversely suppose that M' is a stable matching in I'. Create a matching M in I as follows. Initially let $M = M'$. Suppose that $(m_{j,r}, w_j^r) \in M'$ for some $r \in \{1, 2\}$, where $m_{j,r} = m_i$ for some $m_i \in U$. Then $\{(a_j^1, w_j^r), (a_j^2, w_j^r)\} \subseteq M$ as $l(w_j^r) = 3$. Replace (a_j^1, w_j^r), (a_j^2, w_j^r), $(m_{j,r}, w_j^r)$ in M by (m_i, w_j). It is clear that M is a matching in I, since the lower quotas imply that at most one of w_j^1, w_j^2 can be open, for each $w_j \in W_0$. Moreover, the stability of M' in I' implies that, for each $w_j \in W_0$, exactly one of w_j^1, w_j^2 is open, since at least one of $m_{j,1}$, $m_{j,2}$ ranks w_j in first place in I. Hence it follows that M is a complete stable matching in I. □

2.4 Heuristics and relaxed problems

At the beginning of Section 2, we noted that lower quotas are specified by higher education institutions in Hungary for the areas of study that they offer. However, the results of the previous subsection indicate that the problem of finding a stable matching (if one exists) in such a context is hard. In this subsection we briefly describe heuristics that are used in Hungary to cope with the complexity of this problem. These heuristics themselves motivate some variants of CA-LQ, whose complexity we also consider here.

In the heuristics currently in use in Hungary, a generalised version of the Gale-Shapley algorithm is used several times. After each run, one college is closed and the Gale-Shapley algorithm is called again for the reduced instance until all the remaining colleges achieve their lower quotas. The college to be closed is selected from those colleges that have not reached their lower quota. Moreover, the chosen college is one for which the ratio of the number of assigned applicants to the lower quota is minimum.

Although this natural heuristic gives a matching that is stable for the remaining open colleges, there may still be a blocking college. Moreover, there are some instances where this heuristic might not lead to a stable solution, even when one exists, as the following example shows:

Example 2

We have three applicants, a_1, a_2, a_3, and two colleges, c_1, c_2. The preference lists of these applicants and colleges, and the lower and upper quotas of these colleges, are as follows:

$$P(a_1) : c_1 \qquad P(a_2) : c_2 \ c_1 \qquad P(a_3) : c_2$$
$$P(c_1) : 2 : 2 : a_1 \ a_2 \qquad P(c_2) : 3 : 3 : a_2 \ a_3$$

After running the Gale-Shapley algorithm for the first time in the heuristic, the resulting matching is $M_1 = \{(a_1, c_1), (a_2, c_2), (a_3, c_2)\}$. Neither c_1 nor c_2 has as many applicants as its lower quota. Now, the program closes c_1, because its ratio, $\frac{1}{2}$ is less than the ratio of c_2, which is $\frac{2}{3}$. So c_1 is removed from the instance. In the second round the program finds the matching $M_2 = \{(a_2, c_2), (a_3, c_2)\}$, in which c_2 still does not achieve its lower quota, so the program closes c_2 also, and returns an empty matching. However, it is obvious that the matching $\{(a_1, c_1), (a_2, c_1)\}$ is stable.

Pairwise stable matching with lower quotas

In a relaxed variant of the problem, we may omit the second stability condition and forget about blocking colleges. We refer to the resulting (weaker) stability criterion as *pairwise stability*. A pairwise stable solution always exists; a natural question is to maximise the number of assigned applicants. Let MAX-PS-CA-LQ denote the problem of finding a maximum size pairwise stable matching for an instance of CA-LQ.

This concept has at least two weak points. The first is that MAX-PS-CA-LQ is also NP-hard. The second is that this solution can be unfair. To deal with the first point, we define MAX-PS-CA-LQ-D to be the decision version of MAX-PS-CA-LQ. We will show that MAX-PS-CA-LQ-D is NP-complete by reducing from the NP-complete problem Exact Cover by 3-sets (X3C) [13]. This latter problem is defined as follows: we are given a set $X = \{x_1, \ldots, x_n\}$ of *elements*, and a set $C = \{c_1, \ldots, c_m\}$ of *clauses*, where $c_i \subseteq X$ and $|c_i| = 3$ for each $c_i \in C$, and $n = 3q$ for some $q \in \mathbb{Z}^+$. The question is whether C has an *exact cover* for X, i.e., whether there is a set $C' \subseteq C$ such that C' is a partition of X. This problem is NP-complete even if each element belongs to either two or three clauses [8], therefore we will show that MAX-PS-CA-LQ-D is NP-complete even if all the vertices have degree two or three.

Theorem 2. MAX-PS-CA-LQ-D *is NP-complete, even if each lower and upper quota is equal to 3.*

Proof. We reduce from X3C, as defined above. In fact our transformation is similar to that described in Cornuéjols [7, page 186]. We repeat the construction for completeness, as it will be extended in a subsequent proof. Hence let $\langle X, C\rangle$ be an instance I of X3C, where X and C are as defined above. We construct an instance I' of MAX-PS-CA-LQ-D as follows. The sets of applicants and colleges in I' are X and C respectively. The preference list of each applicant $x_i \in X$ in I' is an arbitrary linear ordering of the clauses in C containing x_i. Similarly the preference list of each college $c_j \in C$ in I' is an arbitrary linear ordering of the elements in the clause c_j. Each college has lower and upper quotas equal to 3. We show that I has an exact cover if and only if I' has a pairwise stable matching of size at least n. This is not difficult since, as Cornuéjols [7] showed, I has an exact cover if and only if I' has a matching (w.r.t. the quotas) of size at least n, and every such matching in I' must be pairwise stable (no open college can form a blocking pair with any of its applicants since all three of them are admitted in a matching that satisfies the quota restrictions). □

The following example shows that, on the one hand, a pairwise stable solution can be arbitrarily larger than a stable solution, but still, the former solution can be regarded as unfair.

Example 3

We have ten applicants, a_1, $a_2, \ldots,$ a_{10}, and two colleges, c_1, c_2. The preference lists of these applicants and colleges, and the lower and upper quotas of these colleges, are as follows:

$P(a_1) : c_1\ c_2$	$P(c_1) :\ \ 1 : 1\ \ \ : a_1$
$P(a_i) : c_2\ (2 \leq i \leq 10)$	$P(c_2) :\ \ 10 : 10\ \ \ : a_1\ a_2 \ldots\ a_{10}$

Here, the only stable matching is $\{(a_1, c_1)\}$ of size 1. However, if we require pairwise stability only, then matching $\{(a_i, c_2) : i = \{1, \ldots, 10\}\}$ has size 10. However, this latter solution may not be considered fair, since a college (c_1) is closed that would have enough applicants in all stable matchings for the corresponding CA instance.

Stable matching with lower quotas for popular colleges

Let I be an instance of CA-LQ. Define a college c_i in I to be *popular* if $|M_0(c_i)| \geq l(c_i)$, where M_0 is a stable matching[2] in the instance I' of CA obtained from I by disregarding the lower quotas. College c_i is *unpopular* otherwise.

As a third variant of CA, in a given instance I of CA-LQ we can try to find a pairwise stable matching M in I, such that no popular college is closed in M. Such a matching M in I is called a *popular pairwise stable matching.*

A popular pairwise stable matching always exists in I, and in fact the heuristic described at the beginning of Section 2.4 will find such a matching. So here, again we can consider the problem of finding a maximum popular pairwise stable matching in I, denoted by MAX-POP-CA-LQ. This problem is a restricted version of MAX-PS-CA-LQ, since we have the extra requirement that no popular college can be closed. Also MAX-POP-CA-LQ is a more general variant of CA-LQ, because in the latter problem, unpopular colleges can be blocking.

In Example 1, a maximum popular pairwise stable matching is $\{(a_1, c_2)\}$, since c_2 is popular whilst c_1 is unpopular. In Example 2, both colleges are unpopular. Here, a maximum popular pairwise stable matching is $\{(a_1, c_1), (a_2, c_1)\}$, however the heuristic described at the beginning of Section 2.4 is not able to find it. In Example 3, c_1 is popular whilst c_2 is unpopular, so the solution is the matching $\{(a_1, c_1)\}$.

Let MAX-POP-CA-LQ-D be the decision version of MAX-POP-CA-LQ. We now show that MAX-POP-CA-LQ is NP-hard.

Theorem 3. MAX-POP-CA-LQ-D *is NP-complete, even if each lower and upper quota is equal to 3.*

Proof. We give a modification of the reduction shown in the proof of Theorem 2 (we henceforth assume the notation introduced in that proof). The applicants and their preference lists are initially as constructed in that proof. Corresponding to each clause $c_j \in C$, we now create two colleges c_j^1 and c_j^2. Suppose that $c_j = \{a_r, a_s, a_t\}$, where $r < s < t$. Each of c_j^1 and c_j^2 prefers a_r to a_s to a_t in I'. We now modify the applicants' preference lists in I' as follows: replace c_j by c_j^1 and c_j^2 (in that order) in each of the preference lists of a_r and a_s, and replace c_j by c_j^2 and c_j^1 (in that order) in the preference list of a_t. Let $u(c_j^k) = l(c_j^k) = 3$ for $k \in \{1, 2\}$. The construction ensures that no college in I' can have 3 applicants in a (Gale-Shapley) stable matching in the CA instance obtained by ignoring the lower quotas, so no college is popular. By this fact, we can use an argument similar to the one in the proof of Theorem 2 to show that I has an exact cover if and only if I' has a popular pairwise stable matching of size at least n. ☐

2.5 Further questions

We remark that all of the above three problems, namely CA-LQ, MAX-PS-CA-LQ and MAX-POP-CA-LQ, remain open for lower quotas at most 2.

[2] By the Rural Hospitals Theorem [12], each college has the same number of assignees in the stable matchings of a given CA instance.

3 Common quotas: hardness results

Until 2007, in the Hungarian matching scheme, there were separate quotas for state financed and privately financed courses of study in each field at each university, and the common quotas of fields applied to state financed places only. As from 2007, there has been a common quota for each field at each university, and still a common national quota for state financed places for each field of study.[3] This latter case may cause difficulties, because a stable matching may not exist in this setting, as is shown in Example 4 below. The main result in this section is that the problem of deciding whether a stable matching exists, given an instance of the College Admissions problem with common quotas, is NP-complete.

3.1 Problem definition

For an instance of CA, we return to using the symbol q to represent (upper) quotas, so we have $q : C \longrightarrow \mathbb{N}$. For $C \subseteq \mathcal{C} \subseteq 2^C$, let $\mathcal{C}$ be a *set system of colleges*, comprising the so-called *bounded* sets of colleges. We extend the domain of q to include every $C_k \in \mathcal{C}$; let $q(C_k)$ be the *common quota* of C_k.

A matching with respect to common quotas is a matching of the CA instance in which no bounded set of colleges has more assignees than its common quota. A matching M is *stable*, if for every acceptable applicant-college pair $(a_i, c_j) \notin M$, either a_i is matched and she prefers $M(a_i)$ to c_j, or c_j fills its quota with better applicants than a_i, or there is a bounded set of colleges C_k such that $c_j \in C_k$ and the common quota of C_k is filled by better applicants than a_i. To make the definition meaningful, there has to be a preference list for every bounded set of colleges, which we consider as a *master list*. That is, we suppose that the following two conditions hold (where $P(C_k)$ denotes the preference list of bounded set C_k):

- if $c_i \in C_k$ and $a_j \in P(c_i)$ then $a_j \in P(C_k)$, and conversely, if $a_j \in P(C_k)$ then there must be a college $c_i \in C_k$ such that $a_j \in P(c_i)$.

- if $c_i \in C_k$ then c_i prefers a_j to another applicant a_r if and only if C_k prefers a_j to a_r.

We note that it is a consequence of the second bullet point that there is some consistency regarding the preferences of bounded sets with common colleges, namely, if $C_k \cap C_l \neq \emptyset$ then an applicant a_j precedes another applicant a_r in the master list of C_k if and only if a_j precedes a_r in C_l's master list.

We denote the decision problem of determining whether a stable matching exists for an instance of CA with common quotas by CA-CQ.

3.2 An unsolvable instance and complexity results

First we give an example to show that a stable matching may not exist, and another example for which the sets of admitted applicants differ in two stable matchings. By using these examples as gadgets, we will prove that CA-CQ is NP-complete.

[3]For example, for CS studies at Budapest University of Technology and Economics (BME) there were 450 places for state-financed study and 50 places for privately financed study, whilst there was a common quota of, say, 3000 places for state financed CS studies in Hungary. Since 2007, there is only a common quota of 500 for CS studies at BME and still the same national quota for state financed CS studies.

Example 4.

We are given three applicants a_1, a_2, a_3, four colleges c_1, c_2, c_3, c_4 and two sets of colleges $\{c_1, c_2\}$, $\{c_2, c_3\}$ with common quotas. In this example, and from this point onwards, the integer after the first colon in the preference list of each college (and each set of colleges) denotes the (common) quota. The preference lists and the quotas are as follows :

$$\begin{array}{rcl} P(a_1) & : & c_1\ c_4 \\ P(a_2) & : & c_2 \\ P(a_3) & : & c_4\ c_3 \end{array} \qquad \begin{array}{rcl} P(c_1) & : & 1 : a_1 \\ P(c_2) & : & 1 : a_2 \\ P(c_3) & : & 1 : a_3 \\ P(\{c_1, c_2\}) & : & 1 : a_2\ a_1 \\ P(\{c_2, c_3\}) & : & 1 : a_3\ a_2 \\ P(c_4) & : & 1 : a_1\ a_3 \end{array}$$

Here, no stable matching exists. To see this, suppose for a contradiction that M is a stable matching. If a_1 is unmatched then (a_1, c_4) is blocking. Otherwise, if $M(a_1) = c_1$ then it must be the case that $(a_3, c_4) \in M$, but then $(a_2, c_2) \notin M$ because of the quota of $\{c_1, c_2\}$, so (a_2, c_2) is blocking. Finally, if $M(a_1) = c_4$ then $(a_3, c_3) \in M$, otherwise this pair would be blocking. But this implies $(a_2, c_2) \notin M$, so (a_1, c_1) is blocking, a contradiction.[4]

Finally we remark that if we remove a_2 from the instance, then it becomes solvable, since $\{(a_1, c_1), (a_3, c_4)\}$ is a stable matching. We will use this fact in the proof of Theorem 4 below.

Example 5.

We are given four applicants a_1, a_2, a_3, a_4, six colleges c_1, c_2, c_3, c_4, c_5, c_6 and four sets of colleges $\{c_1, c_2\}$, $\{c_2, c_3\}$, $\{c_4, c_5\}$, $\{c_5, c_6\}$ with common quotas. The preference lists and the quotas are as follows.

$$\begin{array}{rcl} P(a_1) & : & c_1 \\ P(a_2) & : & c_5\ c_2 \\ P(a_3) & : & c_3\ c_6 \\ P(a_4) & : & c_4 \end{array} \qquad \begin{array}{rcl} P(c_1) & : & 1 : a_1 \\ P(c_2) & : & 1 : a_2 \\ P(c_3) & : & 1 : a_3 \\ P(\{c_1, c_2\}) & : & 1 : a_2\ a_1 \\ P(\{c_2, c_3\}) & : & 1 : a_2\ a_3 \end{array} \qquad \begin{array}{rcl} P(c_4) & : & 1 : a_4 \\ P(c_5) & : & 1 : a_2 \\ P(c_6) & : & 1 : a_3 \\ P(\{c_4, c_5\}) & : & 1 : a_2\ a_4 \\ P(\{c_5, c_6\}) & : & 1 : a_3\ a_2 \end{array}$$

Here, there are precisely two distinct stable matchings, $\{(a_1, c_1), (a_2, c_5), (a_3, c_3)\}$ and $\{(a_2, c_2), (a_3, c_6), (a_4, c_4)\}$. To see this, it is enough to observe that both a_2 and a_3 must be matched in a stable matching, since either (a_2, c_2) or (a_3, c_6) would be blocking otherwise. Under this restriction, only the above two maximal matchings are possible, and it is easy to check that both of them are stable. This example shows that the sets of admitted applicants may differ for two stable matchings.

Theorem 4. CA-CQ *is NP-complete, even if every quota is 1 and no college is included in more than 2 bounded sets.*

Proof. The problem is clearly in NP. To show NP-hardness, we reduce from the NP-complete restriction of COM SMTI as described in Section 2.3. Let I be an instance of this problem, in which $U = \{m_1, \ldots, m_n\}$ is the set of men, and $W = \{w_1, \ldots, w_n\}$ is the set

[4] We note that this situation can be realised in the Hungarian application. Colleges c_1 and c_2 can correspond to privately financed and state financed versions of a particular field of study at some faculty with a common quota, whilst c_3 can correspond to state financed study in the same area (e.g. Swahili language) at another faculty, with a common national quota for c_2 and c_3 respectively. Finally, c_4 can be any other field of study in some different faculty.

of women. Let $W_0 \subseteq W$ be the set of women whose preference list in I is a tie of length 2. We create an instance I' of CA-CQ, with applicant set A and college set C, as follows.

First we create the so-called *proper part* of I'. Every man $m_i \in U$ in I gives rise to an applicant $a_i \in A$, and every woman w_j in $W \backslash W_0$ gives rise to a college c_j in C with quota 1; the preference lists of these agents in I' are derived directly from the corresponding preference lists in I, except that a woman in W_0 appearing in the preference list of a man m_i gives rise to a college in the preference list of applicant a_i as described below. Now let $w_j \in W_0$ and suppose that the two men $m_{j,1}$ and $m_{j,2}$ are tied in w_j's list in I. In I' we create a gadget corresponding to the instance of Example 5. We create two additional applicants b_j^1, b_j^2, six colleges c_j^1, c_j^2, c_j^3, c_j^4, c_j^5, c_j^6 where four sets of colleges $\{c_j^1, c_j^2\}$, $\{c_j^2, c_j^3\}$, $\{c_j^4, c_j^5\}$, $\{c_j^5, c_j^6\}$ have common quotas with the following preferences:

$$\begin{array}{lcl} P(b_j^1) & : & c_j^5 \; c_j^2 \\ P(b_j^2) & : & c_j^3 \; c_j^6 \end{array} \qquad \begin{array}{rcl} P(c_j^1) & : & 1 : a_{j,1} \\ P(c_j^2) & : & 1 : b_j^1 \\ P(c_j^3) & : & 1 : b_j^2 \\ P(\{c_j^1, c_j^2\}) & : & 1 : b_j^1 \; a_{j,1} \\ P(\{c_j^2, c_j^3\}) & : & 1 : b_j^1 \; b_j^2 \end{array} \qquad \begin{array}{rcl} P(c_j^4) & : & 1 : a_{j,2} \\ P(c_j^5) & : & 1 : b_j^1 \\ P(c_j^6) & : & 1 : b_j^2 \\ P(\{c_j^4, c_j^5\}) & : & 1 : b_j^1 \; a_{j,2} \\ P(\{c_j^5, c_j^6\}) & : & 1 : b_j^2 \; b_j^1 \end{array}$$

Let $A_1 = \{a_i : 1 \leq i \leq n\}$, $A_2 = \{b_j^1, b_j^2 : w_j \in W_0\}$. Finally, in the position where w_j appears in the list of $m_{j,1}$ we put c_j^1 in $a_{j,1}$'s list, and similarly the occurrence of w_j in the list of $m_{j,2}$ leads to c_j^4 in $a_{j,2}$'s list. We show that I has a complete stable matching M if and only if the proper part of I' has a stable matching M' in which every applicant $a_i \in A_1$ is matched to some college. This natural correspondence is the following. Whenever $(m_i, w_j) \in M$, with $w_j \in W \setminus W_0$, let $(a_i, c_j) \in M'$, and for every $w_j \in W_0$ let

$$(m_{j,1}, w_j) \in M \iff \{(a_{j,1}, c_j^1), (b_j^2, c_j^3), (b_j^1, c_j^5)\} \in M'$$

and

$$(m_{j,2}, w_j) \in M \iff \{(a_{j,2}, c_j^4), (b_j^2, c_j^6), (b_j^1, c_j^2)\} \in M'.$$

It is straightforward to verify that the completeness and stability of M implies the stability of M' and ensures that M' covers every applicant $a_i \in A_1$.

In the other direction, if M' is a stable matching that covers every applicant $a_i \in A_1$ then it must be the case that either $\{(a_{j,1}, c_j^1), (b_j^2, c_j^3), (b_j^1, c_j^5)\} \subseteq M'$, or $\{(a_{j,2}, c_j^4), (b_j^2, c_j^6), (b_j^1, c_j^2) \subseteq M'$ for every pair of applicants $a_{j,1}$ and $a_{j,2}$ such that $m_{j,1}$ and $m_{j,2}$ are tied in the list of w_j for some $w_j \in W_0$. This is because exactly one of the pairs $(a_{j,1}, c_j^1)$, $(a_{j,2}, c_j^4)$ must belong to M'. They cannot both belong to M' for stability reasons, and if neither of them were to belong to M' then, because both $(a_{j',1}, c_{j'}^1)$ and $(a_{j',2}, c_{j'}^4)$ cannot belong to M' for any other $w_{j'} \in W_0$, M' could not cover every $a_i \in A_1$, a contradiction. Therefore we can create M from M' in the opposite direction from that described, so the correspondence (between the matchings of I and I' with the required properties) is one-to-one, the stability of M' implies the stability of M and the assumption that M' covers every $a_i \in A_1$ implies the completeness of M.

To complete the reduction, we construct the *additional part* of I' as follows. For every $a_i \in A_1$ we add a gadget which is equivalent to the instance defined in Example 4. To be precise, for every $a_i \in A_1$, we create two applicants z_i^1, z_i^2 and four colleges d_i^1, d_i^2, d_i^3, d_i^4 with common quotas for the sets of colleges $\{d_i^1, d_i^2\}$ and $\{d_i^2, d_i^3\}$. The preference lists and the quotas are as follows.

$$
\begin{array}{rcl}
P(z_i^1) & : & d_i^1 \ d_i^4 \\
P(z_i^2) & : & d_i^4 \ d_i^3
\end{array}
\qquad\qquad
\begin{array}{rcl}
P(d_i^1) & : & 1 : z_i^1 \\
P(d_i^2) & : & 1 : a_i \\
P(d_i^3) & : & 1 : z_i^2 \\
P(\{d_i^1, d_i^2\}) & : & 1 : a_i \ z_i^1 \\
P(\{d_i^2, d_i^3\}) & : & 1 : z_i^2 \ a_i \\
P(d_i^4) & : & 1 : z_i^1 \ z_i^2
\end{array}
$$

Finally, we attach d_i^2 to the end of a_i's preference list. We claim that I has a complete stable matching if and only if I' has a stable matching. To prove this we observe that on the one hand, in any stable matching for I', every $a_i \in A_1$ must be matched to a college in the proper part of I', since otherwise the corresponding additional gadget would contain a blocking pair as described in Example 4. On the other hand, if every $a_i \in A_1$ is matched to a college in the proper part of I' then this matching can be extended to the additional part, ensuring stability, by adding $\{(z_i^1, d_i^1), (z_i^2, d_i^4)\}$ to the matching in each gadget, as we remarked in Example 2. □

4 Common quotas: nested set systems

We say that the set system $\mathcal{C}$ is *nested* if, for every pair S, S' of sets in $\mathcal{C}$ such that $S \cap S' \neq \emptyset$, we have either $S \subseteq S'$ or $S' \subseteq S$. In this section we consider a special case of CA-CQ in which the bounded sets form a nested set system.

The Student-Project Allocation problem (SPA), studied by Abraham *et al.* [3], is such a special case of CA-CQ. In SPA, students seek to be matched to projects, and each project is proposed by a single supervisor. In addition to a quota for each project (typically 1), each supervisor has a quota, so the projects proposed by a given supervisor form a bounded set. It is immediate that the bounded sets form a nested set system in this case.

We note that the bounded sets formed a nested set system in the Hungarian application until 2007, because each faculty had separate quotas for state financed and privately financed studies and the common quota of any particular field of study applied to state financed places only. In this section we give two polynomial-time algorithms for CA-CQ with nested set systems. The first algorithm is *applicant-oriented* and the second is *college-oriented*; these produce stable matchings that are, in the first case, unequivocally optimal for the applicants, and in the second case, in a more limited sense that will be explained in Section 4.3, optimal for the colleges.

4.1 Definitions

We suppose that each set consisting of a single college is bounded (by its quota), and we assume that each bounded set S of colleges of cardinality greater than 1 has a quota that is less than the sum of the quotas of any sets in $\mathcal{C}$ that form a partition of S (otherwise S would be redundant as a bounded set.) A *group* is a maximal bounded set, i.e., a bounded set that is contained in no other. So the assumption that bounded sets are nested implies that the groups form a partition of the set of colleges. We denote by $G(c)$ the group containing the college c. If $S \subset S'$ and there exists no bounded set S^* satisfying $S \subset S^* \subset S'$ then S is a *child* of S' and S' is the *parent* of S, the terminology reflecting the tree-structured hierarchies of bounded sets.

Let $S \in \mathcal{C}$. For a given matching M, $M(S)$ denotes the set of assignees of the colleges in S. Recall that $M(a)$ denotes the assigned college of applicant a ($M(a)$ is null if a is unassigned). Relative to M, a bounded set S is *full* if the number of applicants assigned to colleges in that set, $|M(S)|$, is equal to the quota of the set, $q(S)$, otherwise it is *undersubscribed*. A college is *free* if no set containing it is full, and is *constrained* otherwise. So a free college can admit at least one additional applicant without violating any quota restrictions. For a constrained college c let the *critical set of* c relative to M, denoted by $cs_M(c)$, be the innermost, i.e., minimal with respect to $\subseteq$, full bounded set that contains c. A constrained college can admit an additional applicant only if another applicant is rejected by some college in its critical set. The concepts *free*, *constrained* and *critical set* are defined also, in the obvious way, for any bounded set.

As observed in Section 3.1, it is implicit in the definition of stability that colleges in a given group have consistent preferences over applicants. Hence we assume that associated with each group is a strictly ordered master preference list containing all the applicants who are acceptable to at least one college of that group. The preference list of a college c is inherited from the master preference list of $G(c)$.

4.2 Applicant-oriented algorithm

In the applicant-oriented algorithm each applicant applies to the colleges on her preference list, in turn, as in the classical Gale-Shapley algorithm. In a *step* of the algorithm, when applicant a applies to college c, the outcome depends initially on whether c is free. If so, c (provisionally) accepts a. Otherwise, a must compete for a place at c. The applicant who may be displaced by a is not necessarily currently assigned to c. Let S be the critical set of c in M. If a is preferred, on the master list of $G(c)$, to the least preferred assignee b of some college d in S then b is rejected by d, enabling a to be accepted by c. Otherwise c rejects a. As usual, this process continues until every applicant is either assigned, or has been rejected by every acceptable college.

A pseudocode version of the algorithm appears in Figure 1.

```
while (there is an applicant a who is unmatched
        and who has not applied to all of her acceptable colleges) {
    c = the first college on a's preference list to which she has not applied;
    if (c is free)
        assign a to c;
    else {
        let S be the critical set of c;
        if (a is preferred, on the master list of G(c), to the
                worst applicant b who is assigned to a college in S) {
            let d be the college to which b is currently assigned;
            remove (b, d) from the matching;    // i.e. b is rejected by d
            add (a, c) to the matching;
        }
        else
            a is rejected by c;
    }
}
```

Figure 1: Applicant-oriented algorithm

Lemma 5. *(i) If a college c becomes constrained at some point during execution of the applicant-oriented algorithm, then it is never again free.*
(ii) If M is a matching in which c is constrained, and M' is the matching obtained after the subsequent step of the applicant-oriented algorithm, then the least preferred assignee of $cs_{M'}(c)$ cannot be worse than the least preferred assignee of $cs_M(c)$.

Proof. (i) At any point when c has a critical set S, the only way that S can become undersubscribed is as a result of the rejection by some college in S of an existing assignee a. If this leaves S undersubscribed, it must be because some superset S' of S is full, and the rejection of a was because a college in $S' \setminus S$ gained a new more highly ranked assignee. So c remains constrained (by S').
(ii) We first observe that if $S \subset S'$ then it is immediate that the worst assignee of S is at least as highly ranked as the worst assignee of S' (in the appropriate master list).

Suppose that, in the relevant step of the algorithm, applicant b applies to college d. If b is rejected then the matching remains the same, so that the result holds trivially. On the other hand, if d accepts the application of b, let S be the critical set of c in matching M, i.e. $cs_M(c) = S$, and let $S' = cs_{M'}(c)$. We denote $cs_M(d)$ by T, when d is constrained, and consider four cases.
Case (a) d was free or $T \cap S = \emptyset$. In this case it is easy to see that $S = S'$ and $M(S) = M(S')$, so the result follows at once.
Case (b) $T \subset S$. Then $c \notin T$, otherwise the fact that T is full would contradict the assumption that $S = cs_M(c)$. Hence $cs_{M'}(c) = S$. The transformation from M to M' affects only colleges in T, so $M(S) \setminus M(T) = M'(S) \setminus M'(T)$. Furthermore the lowest-ranked applicant in $M'(T)$ must be ranked more highly than the lowest ranked applicant in $M(T)$. It follows that the lowest-ranked applicant in $M'(S)$ must be ranked at least as high as the lowest ranked applicant in $M(S)$.
Case (c) $T = S$. Here, we have $|M(S)| = |M'(S)| = q(S)$ and the worst applicant in $M'(S)$ is ranked more highly than the worst applicant in $M(S)$. In this case, it is possible that $S' \subset S$, but then, by our first observation, the worst applicant in $M'(S')$ is ranked at least as highly as the worst applicant in $M'(S)$, and therefore, is ranked more highly than the worst applicant in $M(S)$.
Case (d) $S \subset T$. If the rejected applicant was not in $M(S)$ then it is immediate that $S = S'$ and $M(S) = M(S')$. Otherwise, it must be the case that $|M'(S)| < |M(S)|$ and also $|M'(S^*)| < |M(S^*)|$ for every S^* where $S \subset S^* \subset T$ and $d \notin S^*$. Furthermore, for every S^* such that $S \subset S^* \subset T$ and $d \in S^*$, $|M'(S^*)| = |M(S^*)| < q(S^*)$ implying $S' = T$. Since S' was the critical set of d, and its worst assignee, who was also the worst assignee of S, was rejected, it follows that the worst applicant in $M'(S')$ is ranked more highly than the worst applicant in $M(S)$. □

It is straightforward to verify that both parts of Lemma 5 are true, not only for a single college c, but for any bounded set S of colleges.

Theorem 6. *(i) The matching M found by the applicant-oriented algorithm is stable.*
(ii) In M, each applicant has the best assignment possible in any stable matching.

Proof. (i) Suppose that M is not stable, and that (a, c) is a blocking pair. Then a must have been rejected by c during the execution of the algorithm. So at that point, c was constrained and the worst assignee b of its critical set must be ranked higher than a on the master list of $G(c)$. By Lemma 5, c remains constrained in the subsequent steps of the algorithm and the worst assignee of its critical set can never be ranked lower than b, and hence not lower than a, a contradiction.
(ii) Now suppose that $(a, c) \in M$, and that there is another stable matching M' with $(a, c') \in M'$, where a prefers c' to c. So (a, c') is a *stable pair*, i.e., a pair that is matched in some stable matching, and c' must have rejected a during the execution of the algorithm. Assume, without loss of generality, that this was the first rejection involving a stable pair.

Let X be the matching at the point in the algorithm just before a college c^* received an application from an applicant a^* that resulted in the rejection of applicant a by college

c'. Let $S = cs_X(c^*)$ and let $X^* = X \cup (a^*, c^*)$, i.e. X^* is the matching X augmented with the additional pair (a^*, c^*). We will show that there exists an applicant b and a college d such that

- $d = X^*(b)$, and
- $b \notin M'(d)$, and
- either d is free in M' or $S \subseteq cs_{M'}(d)$.

To see this, first we note that $|M'(S)| \leq q(S) = |X(S)|$. Moreover, $|X^*(S)| = q(S)+1$, since the quota of S is exceeded after adding the new pair, and $|X^*(S_i)| \leq q(S_i)$ for each $S_i \subset S$, since $S = cs_X(c^*)$, so that S is the innermost set for which the quota is exceeded. This implies that one of the children of S, say S_2, satisfies $|M'(S_2)| < |X^*(S_2)| \leq q(S_2)$. By the same reasoning, we can construct a sequence of sets $S = S_1 \supset S_2 \supset \cdots \supset S_k = \{d\}$ such that for every i, S_{i+1} is a child of S_i and S_i satisfies $|M'(S_i)| < |X^*(S_i)| \leq q(S_i)$. Finally, let b be any applicant from $X^*(d) \backslash M'(d)$. By the construction, it is straightforward to verify that b and d satisfy the required conditions.

Now, we are in a position to complete the proof. We first note that b must be more highly ranked than a, for otherwise d would have rejected b instead of c' rejecting a. Hence, if b is unmatched in M' or prefers d to $M'(b)$ then, since $S \subseteq cs_{M'}(d)$, it follows that (b, d) is a blocking pair for M'. On the other hand, if b prefers $M'(b)$ to d then b must have been rejected by $M'(b)$ earlier than a was rejected by c', a contradiction. □

Corollary 7. *All executions of the applicant-oriented algorithm yield the same stable matching.*

Complexity analysis

We now establish that the complexity of the applicant-oriented algorithm is $O(kL + pn)$ where L is the number of acceptable (applicant, college) pairs, k is the maximum level of nesting of bounded sets, n is the number of applicants and p is the number of bounded sets. The first term comes from the fact that the main loop of the algorithm is executed $O(L)$ times, and, as we establish below, with the exception of updating worst assignees, every step within the loop can be implemented to run in $O(k)$ time. The second term represents the total amount of time for updating worst assignees.

We represent the bounded sets by a forest structure F in which each node corresponds to a bounded set. Each individual college is a leaf node, and the parent of a node representing the bounded set C is the node representing the parent of C. For each individual college c and bounded set C create a boolean value, accessible in constant time, that records whether c belongs to C. We assume that the number of colleges is $O(n)$, and that the data is provided in a form that allows these structures to be set up in $O(pn)$ time.

For each college, maintain a pointer to the leaf node representing it. In each node of F, representing, say, the bounded set C, we store

- a pointer to its parent, to allow us to move up the tree;
- the (common) quota of C;
- the number of applicants currently assigned to colleges in C;
- a pointer to the preference list of the group containing C, and a *worst assignee* pointer, which references the last entry in that list representing an applicant who is currently assigned to a college in C.

For efficiency reasons we adopt a 'lazy' approach to updating the worst assignee pointer. We ensure that it is accurate whenever the bounded set in question is full, since it is only in these circumstances that the pointer is needed. It is initialised to point to a dummy entry at the end of the relevant master list.

To determine whether a college c is constrained, and if so to find its critical set, it is enough to follow the path of parent pointers from the leaf node representing c until a node is reached for which the number of assignees is equal to the quota, or until the root of the tree is reached without this condition being satisfied. This can be done in $O(k)$ time.

When a pair (a, c) is added to, or deleted from, the matching, follow the path from the leaf representing c to the root, incrementing or decrementing the number of assignees at each node, as appropriate. This can be achieved in $O(k)$ time.

If the bounded set C represented by a node becomes full as a result of incrementing the number of assignees, the worst assignee pointer is updated. By the analogue of Lemma 5(ii) for an arbitrary bounded set, the worst assignee of C must be at least as good as the last time C was full, so this update can be accomplished by scanning backwards in the appropriate master preference list, from that former worst assignee, until we reach an applicant assigned to a college in C. Because of the initalisation of the worst assignee pointer, this works also in the case where C becomes full for the first time. For each node in the forest, each entry in the appropriate master list is visited at most once during such scans, so the total time for all such updates is the sum, taken over all trees in the forest, of the number of nodes times the length of the master list, which is $O(pn)$.

4.3 College-oriented algorithm

The college-oriented algorithm is a little more subtle. As we would expect, the basis is a sequence of offers from colleges to applicants. Each college c makes offers to applicants in the order in which they appear in the master list of $G(c)$, and never makes an offer to the same applicant twice. As always, when an applicant receives an offer, she (provisionally) accepts it if she is unmatched at that point, otherwise she accepts it if and only if it is a better offer than the one she currently holds.

At each step of the algorithm an offer is initiated by an undersubscribed group G that contains a free college with an *untried* applicant (i.e., an applicant to whom it has not yet offered a place). Let a be the highest ranking applicant in the master list of G who is an untried applicant for some free college in G. Among these free colleges, let c be the one most preferred by a; then c offers a place to a at that point. We say that the pair (c, a) constitutes the *leading offer* at that point.

A pseudocode version of the algorithm appears in Figure 2.

To establish the correctness of this algorithm, we again refer to the hierarchical forest of bounded sets, identifying each bounded set with the node representing it.

The following technical lemma is used in the proof of Theorem 9.

Lemma 8. *If, at some step during the execution of the algorithm, a college d in group G offers a post to an applicant b, and (a) some other superior applicant a is not offered a post by a college c in G where a is not marked as a 'tried' applicant for c, or (b) b is not offered a post by a preferred college c where b is not marked as a 'tried' applicant for c, then c must be constrained, and its critical set at that point is not on the path from d to the root of the tree.*

Proof. The fact that c must be constrained is straightforward from the definition of the leading offer. Furthermore, since d is a free college, there cannot be any full bounded set on the path from d to the root of the tree (so no critical set either). □

```
while (some group G of colleges is undersubscribed,
        and contains a free college with an untried applicant) {
    let (c, a) constitute the leading offer at this point;
        mark a as a 'tried' applicant for c;
    if (a is unassigned)
        add (a, c) to the matching;
    else if (a prefers c to her current assigned college d) {
        remove (a, d) from the matching;     // d is rejected by a
        add (a, c) to the matching;
    }
    else
        a rejects c;
}
```

Figure 2: College-oriented algorithm

Now we can establish the key properties of the college-oriented algorithm.

Theorem 9. *(i) The matching M produced by the college-oriented algorithm is stable.*
(ii) If an applicant a is matched to a college c in M then there is no stable matching in which a is unmatched, or matched to a college lower on her preference list, i.e., M is the applicant-pessimal stable matching.
(iii) If a college c is matched to applicants $a_1, \ldots, a_k$ in M, then there is no applicant more highly ranked than any of $a_1, \ldots, a_k$ who is matched to c in any stable matching.

Proof. (i) Suppose that M is not stable, and that it is blocked by an applicant a and college c. Then c cannot have offered a place to a, because applicants can only improve as the algorithm progresses, so this would contradict the assumption that (a, c) is a blocking pair for M. It follows that c must have a critical set, say S, on termination of the algorithm.

If (a, c) is a blocking pair then there is some college d in S and some applicant b who is assigned to d in M such that either b is inferior to a, or b is equal to a and a prefers c to d. If b is inferior to a then it might be the case that d is equal to c. However, in this case it is easy to see the contradiction, because when c offered a place to b, c must have offered to the more preferred applicant a and hence a must be eventually matched to a college at least as good as c. Suppose now that d is not equal to c and b is the worst assignee of S (where b could be equal to a). Let $\mathcal{P}$ denote the path from the node representing d to the root of the tree containing it. It is immediate that the node representing S lies on $\mathcal{P}$. Further let $\mathcal{Q}$ denote the path from the node representing c to the node representing S.

At the step when d offered a place to b, it follows from Lemma 8 that c's critical set must have been a descendant node of S not on $\mathcal{P}$, say T. Let M^* denote the matching at the point just before d offered a place to b. Since d is free in M^*, it follows that every node on $\mathcal{P}$ is undersubscribed, implying in particular that $|M^*(S)| < q(S)$. On the other hand, T is the critical set of c in M^*, which implies $|M^*(T)| = q(T)$. Further let $M' = M \setminus \{(b, d)\}$. The fact that S is the critical set of c in M implies that every node below S on $\mathcal{Q}$, including T, is undersubscribed in M and therefore also in M', namely $q(T) > |M'(T)|$.

From the above observations we have $|M'(S)| = q(S) - 1 \geq |M^*(S)|$. Let us consider the children of S. It must be the case that either
a1) there exists a child T_1 of S such that $T \subset T_1 \subset S$ and $|M^*(T_1)| \leq |M'(T_1)|$, or
b1) there exists a child S_1 of S such that $T \not\subseteq S_1 \subset S$ and $|M^*(S_1)| < |M'(S_1)|$.
This is because if $|M^*(T_1)| > |M'(T_1)|$ holds for T_1, where T_1 is the child of S on the

path from S to T, then there must be some other child of S, say S_1 (not on the path from S to T), for which $|M^*(S_1)| < |M'(S_1)|$. Note that T_1 cannot be equal to T, since $|M^*(T)| = q(T) > |M'(T)|$.

In case b1) we can construct a sequence of sets $S \supset S_1 \supset S_2 \supset \cdots \supset S_k = d^*$ such that for every i, S_{i+1} is a child of S_i and S_i satisfies $|M^*(S_i)| < |M'(S_i)|$. This means that d^* is a free college in M^* and d^* has an assignee b^* in $M'(S) \setminus M^*(S)$ who must be better than b (since b was the worst assignee of S in M), leading to a contradiction since d^* should have offered a place to b^* at the point where d offered a place to b.

In case a1) we use the same argument for T_1 as we did for S. It must be the case that either
a2) there exists a child T_2 of T_1 such that $T \subset T_2 \subset T_1$ and $|M^*(T_2)| \le |M'(T_2)|$, or
b2) there exists a child S_1 of T_1 such that $T \not\subset S_1 \subset T_1$ and $|M^*(S_1)| < |M'(S_1)|$.
As before, T_2 cannot be equal to T.

Again, in case b2) we can construct a sequence of sets $S \supset T_1 \supset S_1 \supset S_2 \supset \cdots \supset S_k = d^*$ leading to a similar contradiction. On the other hand, in case a2) we continue with the same reasoning for T_2 leading to subcases a3) and b3), and so on.

Continuing in this way, after at most j steps, where j is the length of the path $\mathcal{Q}$, we are led to an inevitable contradiction, since case a) cannot arise at the jth step.

(ii) Suppose that there is a stable matching M' in which a is unmatched, or is matched to a college lower than c on her preference list. Suppose, without loss of generality, that, when c offered a post to a, this was the first time that an applicant received an offer that was superior to her status in some stable matching. Call such an offer a *superior* offer. In order to avoid (a, c) being a blocking pair for M', the critical set of c in M', say $S = cs_{M'}(c)$, must be full in M' with applicants ranked more highly than a. Suppose that just before c offered a place to a we had a matching M^*. Since c was free at that moment, we have $|M^*(S)| < |M'(S)| = q(S)$. This implies that one of the children of S, say S_2, satisfies $|M^*(S_2)| < |M'(S_2)| \le q(S_2)$. By the same reasoning, we can construct a sequence of sets $S = S_1 \supset S_2 \supset \cdots \supset S_k = d$ such that for every i, S_{i+1} is a child of S_i and S_i satisfies $|M^*(S_i)| < |M'(S_i)| \le q(S_i)$. Finally, let b be any applicant from $M'(d) \setminus M^*(d)$. It follows from the construction that d is free in M^* and that b is more highly ranked than a. So when c offered a place to a, b must have been matched to a college e in M^* that she prefers to d. Therefore b had already received an offer from a college (e) that is preferred to the college (d) she is matched to in the stable matching M', which contradicts the assumption that c's offer to a was the first superior offer.

(iii) Suppose that (a, c) is in M, and that $(b, c) \notin M$ is in some stable matching M', with b more highly ranked than a. Then, by part (ii), b must prefer c to $M(b)$ or b is unmatched in M, and it is immediate that (b, c) is a blocking pair for M, a contradiction.

□

The following is an immediate corollary of Theorem 9(ii).

Corollary 10. *All executions of the college-oriented algorithm yield the same stable matching.*

Note that Theorem 9(iii) does not claim that, in matching M, each college has the 'best' set of assigned applicants that it can have in any stable matching. This need not be true. For example, a college can be assigned a particular set A of applicants in the matching found by the college-oriented algorithm, and a superset of A in some other stable matching. This also indicates that, in contrast to the classical case, the number of applicants assigned to a college may vary between one stable matching and another. These points are illustrated in the following example. (Despite this fact, we will use the term "college-optimal" for the matching obtained by a general kind of college-oriented algorithm in Section 5, however the expression "applicant-pessimal" is probably more accurate in this case.)

Example 6.

We have five colleges $c_1, \ldots, c_5$ and ten applicants $a_1, \ldots, a_{10}$. The bounded sets, in addition to the individual colleges, are $\{c_1, c_2\}$, $\{c_1, c_2, c_3\}$ and $\{c_4, c_5\}$, so that the latter two are the groups, which we refer to as Group A and Group B respectively. The preferences of the applicants, quotas of the bounded sets, and master lists of the groups are as shown:

$P(a_1) : c_1\ c_5$	$q(c_1) : 3$
$P(a_2) : c_1\ c_4$	$q(c_2) : 3$
$P(a_3) : c_2\ c_1$	$q(c_3) : 3$
$P(a_4) : c_2\ c_4$	$q(c_4) : 2$
$P(a_5) : c_5\ c_3$	$q(c_5) : 2$
$P(a_6) : c_3\ c_4$	$q(c_1, c_2) : 5$
$P(a_7) : c_1\ c_4\ c_5$	$q(c_1, c_2, c_3) : 7$
$P(a_8) : c_2\ c_1$	$q(c_4, c_5) : 3$
$P(a_9) : c_4\ c_1$	$P(c_1, c_2, c_3) : a_8\ a_9\ a_1\ a_2\ a_3\ a_4\ a_5\ a_6\ a_7\ a_{10}$
$P(a_{10}) : c_2\ c_4\ c_5$	$P(c_4, c_5) : a_4\ a_7\ a_9\ a_{10}\ a_5\ a_1\ a_2\ a_6$

An execution of the Applicant-oriented algorithm is as follows. Here, we use the notation $a \longrightarrow c$ to stand for 'applicant a applies to college c'.

$a_1 \longrightarrow c_1$; accepted.
$a_2 \longrightarrow c_1$; accepted.
$a_3 \longrightarrow c_2$; accepted.
$a_4 \longrightarrow c_2$; accepted.
$a_5 \longrightarrow c_5$; accepted.
$a_6 \longrightarrow c_3$; accepted.
$a_7 \longrightarrow c_1$; accepted.
$a_8 \longrightarrow c_2$; $\{c_1, c_2\}$ is full; c_1 rejects a_7; c_2 accepts a_8.
$a_7 \longrightarrow c_4$; accepted.
$a_9 \longrightarrow c_4$; accepted.
$a_{10} \longrightarrow c_2$; c_2 is full; c_2 rejects a_{10}.
$a_{10} \longrightarrow c_4$; c_4 is full; c_4 rejects a_{10}.
$a_{10} \longrightarrow c_5$; $\{c_4, c_5\}$ is full; c_5 rejects a_5; c_5 accepts a_{10}.
$a_5 \longrightarrow c_3$; accepted.

The applicant-optimal stable matching results, namely

$$M_1 = \{(a_1, c_1), (a_2, c_1), (a_3, c_2), (a_4, c_2), (a_5, c_3), (a_6, c_3), (a_7, c_4), (a_8, c_2), (a_9, c_4), (a_{10}, c_5).$$

An execution of the College-oriented algorithm is as follows. Here, we use the notation $c \longrightarrow a$ to stand for 'the leading offer is from college c to applicant a'.

Group A undersubscribed; $c_2 \longrightarrow a_8$; accepted.
Group A undersubscribed; $c_1 \longrightarrow a_8$; rejected.
Group A undersubscribed; $c_1 \longrightarrow a_9$; accepted.
Group A undersubscribed; $c_1 \longrightarrow a_1$; accepted.
Group A undersubscribed; $c_1 \longrightarrow a_2$; accepted.
Group A undersubscribed; $c_2 \longrightarrow a_3$; accepted.
Group A undersubscribed; $c_3 \longrightarrow a_5$; accepted.
Group A undersubscribed; $c_3 \longrightarrow a_6$; accepted.
Group B undersubscribed; $c_4 \longrightarrow a_4$; accepted.
Group B undersubscribed; $c_4 \longrightarrow a_7$; accepted.
Group B undersubscribed; $c_5 \longrightarrow a_7$; rejected.
Group B undersubscribed; $c_5 \longrightarrow a_{10}$; accepted.

The applicant-pessimal stable matching results, namely

$$M_2 = \{(a_1, c_1), (a_2, c_1), (a_3, c_2), (a_4, c_4), (a_5, c_3), (a_6, c_3), (a_7, c_4), (a_8, c_2), (a_9, c_1), (a_{10}, c_5)$$

Note that colleges c_1 and c_2 have different numbers of assigned applicants in the two matchings. College c_2, in particular, has more assignees in M_1 than in M_2, though the additional assignee, a_4, is lower ranked than the others, as must be the case according to Theorem 9(iii).

Complexity analysis

The number of loop iterations is $O(L)$, where L is the number of acceptable applicant-college pairs, since no college makes more than one offer to the same applicant. At each loop iteration, we have to decide which college makes an offer to which applicant. This can be decided by a traversal of the forest F of bounded sets. During this traversal, when we encounter a node representing a bounded set that is full, we need not enter any of its subtrees, since no free college can thereby be reached. When we reach a leaf node that represents a free college c, we can locate its next untried applicant a (if any), and determine whether an offer from c to a is a better candidate for leading offer than any previously seen, all in constant time, with suitable data structures. Also, as in the applicant-oriented algorithm, whenever a pair (a, c) is added to or removed from the matching, we update the count of assignees in each node that is an ancestor of the leaf node representing c. Hence each loop iteration can be completed in $O(m)$ time, the time for the forest traversal, giving an overall time bound of $O(mL)$ for the college-oriented algorithm.

We saw earlier that the complexity of the applicant-oriented algorithm is $O(kL + pn)$, where k is the maximum level of nesting of bounded sets, n is the number of applicants and p is the number of bounded sets. Since $k = O(m)$, $p = O(m)$ and $n = O(L)$, it follows that the complexity bound that we have derived for the applicant-oriented algorithm is at least as good as that for the college-oriented algorithm.

5 CA-CQ with nested set systems: choice functions and a matroid model

From Example 6, it is clear that certain well-known properties of the classical College Admissions problem do not carry over directly to the case of CA-CQ with nested set systems; for example, that each college should have the same number of assigned applicants in all stable matchings. However, it turns out that there are analogues of many of the structural aspects of the classical problem. To investigate these, we adopt in this section a more general model based on choice functions and matroids. Recall that $G = (A \cup C, E)$ denotes the bipartite graph with applicants and colleges as colour classes, and acceptable pairs forming the set E of edges. In this section we work with the edges rather than the vertices of G, so for this reason, for a set E' of edges and set V' of vertices (and for vertex v), $E'(V')$ (and $E'(v)$) will denote the set of those edges of E' that are incident with some vertex of V' (and with vertex v, respectively). In particular, if M is a stable matching and C_i is a bounded set, $M(C_i)$ stands for all those edges with an end vertex that is a college in C_i. (Hitherto, by $M(C_i)$ we meant the corresponding set of applicants.)

5.1 Choice functions

We define two specific choice functions on the edge set E of G. A *choice function* on ground set E is a mapping Q that assigns to each subset X of E a subset $Q(X)$ of X. (Note that in the Economics literature choice functions are often defined in a much more restrictive way, that is, Q is a *preference induced choice function on E* if there is a (well-) order $\prec$ on all subsets of E such that $Q(X)$ is the $\prec$-minimal subset of X.) A subset X of E is called *Q-independent* if $Q(X) = X$.

The choice function of the applicants is denoted by Q_A and for subset X of E, $Q_A(X)$ is the set of those edges in X that the applicants would choose if they could select freely from X, ignoring all quotas on the college side. Formally, every applicant a that has at least one edge (application) in X chooses her most preferred edge from X, and $Q_A(X)$ is the set of edges selected in this way. An *applicant-independent* set of edges is a set X of edges for which $Q_A(X) = X$.

For the choice function Q_C of colleges, $Q_C(X)$ denotes the set of edges that the colleges would accept if they could freely select from X and could accept several edges from the same applicant. More precisely, subset $Q_C(X)$ of X is determined by the following algorithm. Order the bounded sets as $\mathcal{C} = \{C_1, C_2, \ldots, C_m\}$ such that if $C_i \subseteq C_j$ then $i \le j$ holds. (In other words, take a linear extension of the partial order on $\mathcal{C}$ given by set inclusion.) Let $X_0 := X$, and for $i = 1, 2, \ldots m$ let X_i denote the set obtained from X_{i-1} after applying the quota of C_i. So $X_{i-1}(C_i)$ is the set of edges of X_{i-1} that are incident with some college in C_i. Let X'_i denote the set of those edges of $X_{i-1}(C_i)$ that are not amongst the best $q(C_i)$ edges of this set. Now let $X_i := X_{i-1} \setminus X'_i$ and let $Q_C(X) := X_m$. To construct X'_i, in addition to the quota $q(C_i)$, we also need the preference order on $X_{i-1}(C_i)$. Although we have a preference order for C_i on the applicants (that is, for the vertices), here we need one on the edges. The difference is that $X_{i-1}(C_i)$ may contain several edges from the same applicant-vertex. We use the following preference order: if two edges in $X_{i-1}(C_i)$ correspond to different applicants then the original preference order of C_i is used. To compare two edges from the same applicant, we use the applicant's preference order. Therefore we have the following partial order on the edges, denoted by $\preceq_E$: an edge $(a, c) \in E(G)$ is preferred to $(b, d) \in E(G)$ if either applicant a is preferred to applicant b by the bounded set containing both c and d or if $a = b$ and college c is preferred to college d by applicant a. We call subset X of E *college-independent* if $Q_C(X) = X$. Note that in this case each bounded set C_i contains at most $q(C_i)$ edges from X.

Clearly, if X is a set of edges then $Q_A(X) = X$ if and only if each applicant has at most one edge in X, and $Q_C(X) = X$ holds if and only if X satisfies all quotas of $\mathcal{C}$. Consequently, subset M of E is a matching if and only if M is both applicant-independent and college-independent. Let edge $(a, c) = e$ of E be an edge from applicant a to college c. We say that subset X of E *dominates edge e from the applicant side* if $e \notin Q_A(X \cup \{e\})$. This means that X contains an edge between some applicant-vertex a and college-vertex c' such that a prefers c' to c. Subset X of E *dominates edge e from the college side* if $e \notin Q_C(X \cup \{e\})$. This means that there is a bounded set C_i such that $c \in C_i$ and

$Q_C(X \cup \{e\})$ contains $q(C_i)$ different edges, each of which is better than e according to the preference order of C_i. We say that subset X of E *dominates subset* Y *of* E *from the college (applicant) side* and denote this fact by $X \preceq_C Y$ $(X \preceq_A Y)$ if each element of $Y \setminus X$ is dominated by X from the college (applicant) side. Subset X *dominates* e if X dominates e from the applicant side or from the college side (or from both). If each edge of Y is dominated by X then we say that X *dominates* Y. The above definitions are also valid for any arbitrary choice function Q on E, in particular subset X of E Q*-dominates* subset Y of E (in notation $X \preceq_Q Y$) if X Q-dominates each element y of $Y \setminus X$, that is, if $y \notin Q(X \cup \{y\})$.

From these definitions, it is not difficult to see that a matching M is stable if and only if the set of those edges that M dominates is $E \setminus M$. That is, each edge outside M is dominated by M, but M does not dominate any edge of M, hence M is both applicant and college-independent. Edge e *blocks* matching M if e is not dominated by M. More generally, if G is a bipartite graph with edge set E and colour classes C and A, and we are given choice functions Q_C and Q_A on E, then subset M of E is a *matching* if $M = Q_C(M) = Q_A(M)$ holds, that is, if M is both Q_A-independent and Q_C-independent. Matching M is *stable* if M dominates all edges of $E \setminus M$, that is, for any $e \notin M$ we have $e \notin Q_C(M \cup \{e\})$ or $e \notin Q_A(M \cup \{e\})$.

We show that our choice functions Q_A and Q_C have two important properties. A choice function Q on E is *comonotone* if the mapping of unchosen elements $X \mapsto X \setminus Q(X)$ is monotone, that is, if $X \subseteq Y \subseteq E$ implies $X \setminus Q(X) \subseteq Y \setminus Q(Y)$, or, in other words $\overline{Q}(X) \subseteq \overline{Q}(Y)$, where for choice function Q, $\overline{Q}(X) := X \setminus Q(X)$ denotes the unselected elements. Roughly speaking, if some option x is ignored then it will still be ignored from a greater choice set. Choice function Q is *increasing* if $X \subseteq Y \subseteq E$ implies $|Q(X)| \le |Q(Y)|$.

To justify the above properties for Q_A and Q_C, we prove that choice function Q_C is closely related to matroids. (The interested reader is referred to the second volume of the book of Schrijver [25] that contains everything we need about matroids.) We recall that subset E' of edges is college-independent if each bounded set C_i contains at most $q(C_i)$ edges from E'. Let $\mathcal{I}_C \subseteq 2^E$ denote the system of college-independent sets of edges.

Theorem 11. *Set system $\mathcal{I}_C$ forms the independent set of a matroid on set E of edges, that is $\mathcal{M}_C = (E, \mathcal{I}_C)$ is a matroid.*

Proof. We use induction on the number m of bounded sets. If $m = 1$ then we have only one bounded set, and as each college is a bounded set by itself, $(E, \mathcal{I}_C)$ is a uniform matroid of rank $q(C_1)$.

For $m > 1$, let $C^1, C^2, \ldots, C^k$ be the groups, that is the inclusionwise maximal bounded sets. Let $\mathcal{I}_{C^i}$ denote the system of those college-independent edge sets that only contain edges with college vertices from C^i. If $k > 1$ then $\mathcal{I}_{C^i}$ forms the independent set of a matroid by the induction hypothesis. This means that $(E, \mathcal{I}_C)$ is the direct sum of these matroids, hence it is a matroid, as we claimed.

On the other hand, if $k = 1$, that is, if $C^1 = C_m$ is the unique group, then let $\mathcal{I}'$ denote the set of those edges that are college-independent for the set $C_1, C_2, \ldots, C_{m-1}$ of bounded sets. By the induction hypothesis, $(E, \mathcal{I}')$ is a matroid, hence its $q(C_m)$-truncation $(E, \mathcal{I}_C)$ is also a matroid. $\square$

Note that the proof of Theorem 11 also implies that "applicant-independent" subsets of edges (i.e., those subsets of the edges that contain at most one application from each applicant) form a matroid. The reason is that these subsets can be regarded as college independent where the bounded sets are the singletons and each quota is 1.

Observe that choice function Q_C always selects a college-independent set of edges. Moreover, for any set X of edges, $Q_C(X)$ has the property that it is an inclusionwise maximal college-independent subset of X: for each unselected element x of $X \setminus Q_C(X)$ subset $\{x\} \cup Q_C(X)$ is not college-independent, because there is some bounded C_i such that $Q_C(X)$ contains $q(C_i)$ edges to C_i with the property that all of these $q(C_i)$ elements are preferred to x.

This observation has two consequences. On the one hand, $|Q_C(X)| = r_{\mathcal{M}_C}(X)$ is the $\mathcal{M}_C$-rank of X for any X. As $\mathcal{M}_C$ is a matroid, $r_{\mathcal{M}_C}(X) \le r_{\mathcal{M}_C}(Y)$ whenever $X \subseteq Y$. This implies that Q_C (and hence Q_A) is increasing.

On the other hand, the above observation means that $Q_C(X)$ can be constructed by the greedy algorithm in the following way. Fix a linear order $e_1, e_2, \ldots$ of E in such a way that if two edges, e_i and e_j belong to the same bounded set then $i < j$ if and only if edge e_i is preferred to e_j (according to the partial order $\preceq_E$). Let $E_i := \{e_j : j \le i\}$ and $X_i := X \cap E_i$. The greedy algorithm constructs each $Q_C(X_i)$ in such a way that $Q_C(X_0) := \emptyset$ and

$$Q_C(X_{i+1}) := \begin{cases} Q_C(X_i) \cup \{e_{i+1}\} & \text{if } Q_C(X_i) \cup \{e_{i+1}\} \text{ is independent} \\ Q_C(X_i) & \text{otherwise.} \end{cases}$$

To prove comonotonicity of Q_C, we have to show that if for edge x of X we have $x \notin Q_C(X)$ then $x \notin Q_C(Y)$ holds whenever $X \subseteq Y$. But this follows immediately from the greedy algorithm: an edge x is thrown away by the greedy algorithm during the construction of $Q_C(Z)$ if and only if x is spanned in $\mathcal{M}_C$ by those elements of Z that precede x. So if x is not selected from X then x will not be selected from Y either, and this means that Q_C (and hence Q_A also) is comonotone.

Below we give a direct proof of the comonotonicity and the increasing property of our choice functions without using the matroid characterisation.

To prove that the choice functions Q_A of applicants and Q_C of colleges are comonotone and increasing, we exhibit a special property of Q_C that implies this. To specify this property, we say that subset Y of the partially ordered set $(E, \preceq)$ is *better* than subset X of E if there is an injective map $f : X \to Y$ such that

$$f(x) \preceq x \text{ for all } x \in X, \tag{1}$$

that is, for each element x of X there is a different element $f(x)$ of Y that is at least as preferred as x. (Here preference order $\preceq$ represents a ranking, so $\preceq$-smaller is preferred to $\preceq$-greater.) The disjoint union of better sets remains better according to the following lemma.

Lemma 12. *If $Y_1, Y_2, \ldots, Y_k$ are disjoint sets and Y_i is better than X_i for $i = 1, 2, \ldots, k$ then $\bigcup_{i=1}^k Y_i$ is better than $\bigcup_{i=1}^k X_i$.*

Proof. We can congregate functions f_i on X_i with property (1) into one injective function $f : \bigcup_{i=1}^k X_i \to \bigcup_{i=1}^k Y_i$ that also has property (1). □

The next lemma shows how to decide whether one set is better than another.

Lemma 13. *Let subsets $X = \{x_1, x_2, \ldots, x_k\}$ and $Y = \{y_1, y_2, \ldots, y_l\}$ of the linearly ordered set E be listed in increasing order. Then Y is better than X if and only if $f(x_i) := y_i$ defines a mapping with property* (1)*.*

Proof. Sufficiency is trivial. For the necessity, if Y is better than X then there is a mapping f' with property (1) that maps the smallest i elements of X into i different elements of Y such that none of them is greater than the ith element of X. So x_i cannot be smaller than y_i, hence f has property (1). □

At this point we can claim the promised property of Q_C. A choice function Q on partially ordered set $(E, \preceq)$ is *improving* if $Q(Y)$ is better than $Q(X)$ whenever $X \subseteq Y$ holds.

Lemma 14. *Choice function Q_C on E is improving for the partial order $\preceq_E$.*

Note that it follows from Lemma 14 that function f must have the property that for each edge x of X there is a bounded set C_i that contains the colleges of both edges x and $f(x)$.

Proof. We apply induction on $|\mathcal{C}|$. For $|\mathcal{C}| = 1$ the lemma is straightforward. Assume now that Q_C is improving whenever $|\mathcal{C}| < n$ and suppose $|\mathcal{C}| = n$ and $X \subseteq Y$. Let $C^1, C^2, \dots, C^k$ denote the groups, i.e., the inclusionwise maximal elements of $\mathcal{C}$. If $k > 1$ then the choice function $Q_C|_{C^i}$ restricted to C^i has fewer bounded sets, hence is improving for all $i = 1, 2, \dots, k$. In particular, $Q_C|_{C^i}(Y \cap C^i)$ is better than $Q_C|_{C^i}(X \cap C^i)$. Observe that $Q_C(Z) = \bigcup_{i=1}^k Q_C|_{C^i}(Z \cap C^i)$ holds for any set Z of edges, and the right hand side is a disjoint union. Hence by Lemma 12 $Q_C(Y) = \bigcup_{i=1}^k Q_C|_{C^i}(Y \cap C^i)$ is better than $Q_C(X) = \bigcup_{i=1}^k Q_C|_{C^i}(X \cap C^i)$.

If $k = 1$ that is, if $C^1 = C$ then let Q'_C be the choice function that we get from Q_C by disregarding the quota of C. As Q'_C has $n - 1$ bounded sets, Q'_C is improving by the induction hypothesis. Let $Q'_C(X) = \{x_1, x_2, \dots, x_k\}$ and $Q'_C(Y) = \{y_1, y_2, \dots, y_l\}$ be listed in increasing order. By Lemma 13, $Q'_C(x_i)$ cannot precede $Q'_C(y_i)$ in the preference order of bounded set C, so after applying the quota of C we get that $Q_C(Y)$ is better than $Q_C(X)$. $\square$

Lemma 15. *Choice functions Q_A of applicants and Q_C of colleges are comonotone and increasing.*

Proof. Choice function Q_A can be regarded as a special college-type choice function where $\mathcal{C}$ consists of singleton elements and $q(\{c\}) = 1$ for each element $\{c\} \in \mathcal{C}$. So it is enough to prove that any college-type choice function Q_C is increasing and comonotone. Let $X \subseteq Y$. By Lemma 14, Q_C is improving, hence there is an injection $f : Q_C(X) \to Q_C(Y)$. It follows that Q_C is increasing. To prove comonotonicity, we have to show that whenever element x of X is not in $Q_C(X)$ then $x \notin Q_C(Y)$ holds.

So assume that Q_C does not select x from X because $x \in X'_i$, that is, x is thrown away when we check the quota of C_i. But the restricted choice function $Q_C|_{C_i}$ is improving by Lemma 14, hence x cannot fit with $q(C_i)$ either in the construction of $Q_C(Y)$. $\square$

Comonotone and increasing choice functions have some interesting properties. The following three statements, namely Lemmas 16, 17 and Theorem 18 are well-known in lattice theory. We include their proofs for the sake of completeness.

Lemma 16. *If choice function Q on E is comonotone and increasing then*

$$Q(X) \subseteq Y \subseteq X \Rightarrow Q(X) = Q(Y) \text{ holds.} \tag{2}$$

Proof. If $Q(X) \subseteq Y \subseteq X$ then $Y \setminus Q(Y) \subseteq X \setminus Q(X)$ by comonotonicity, hence $Q(X) \subseteq Q(Y)$. The increasing property of Q implies $|Q(Y)| \leq |Q(X)|$, and with the previous relation this implies that $Q(X) = Q(Y)$. $\square$

An important consequence of Property (2) is that for any comonotone and path-independent choice function Q on E, relation $\preceq_Q$ is a partial order on Q-independent subsets of E. We need the following useful lemma to prove this.

Lemma 17. *If Q is a comonotone choice function with Property (2) on E and X and Y are subsets of E then $X \preceq_Q Y$ (that is, X dominates subset Y of E) if and only if $Q(X \cup Y) = Q(X)$.*

Proof. Assume first that $X \preceq_Q Y$. For any $y \in Y \setminus X$ we have that $y \in \overline{Q}(X \cup \{y\})$, hence by monotonicity of $\overline{Q}$ it follows that $(Y \setminus X \subseteq \overline{Q}(X \cup Y)$, that is, $Q(X \cup Y) \subseteq X \subseteq X \cup Y$. Now property (2) implies that $Q(X) = Q(X \cup Y)$.

Suppose now that $Q(X) = Q(X \cup Y)$ holds and let $y \in Y \setminus X$. Property (2) and $Q(X) \subseteq X \cup \{y\} \subseteq X \cup Y$ yields that $Q(X \cup \{y\}) = Q(X \cup Y) = Q(X)$, hence each element of $Y \setminus X$ is dominated by X, that is, $X \preceq_Q Y$. □

This observation allows us to prove that $\preceq_Q$ is a partial order on Q-independent sets.

Theorem 18. *If Q is a comonotone choice function with Property (2) on E then $\preceq_Q$ is a partial order on Q-independent subsets of E.*

Proof. For any subset X of E we have $Q(X) = Q(X \cup X)$. This means $X \preceq_Q X$ by Lemma 17, hence $\preceq_Q$ is reflexive. Assume that X and Y are Q-independent E such that $X \preceq_Q Y$ and $Y \preceq_Q X$ holds. Lemma 17 shows that $X = Q(X \cup Y) = Y$. This justifies the antisymmetry of $\preceq_Q$. For transitivity, assume that X, Y and Z are Q-independent and $X \preceq_Q Y \preceq_Q Z$. This means that $Q(X \cup Y) = X$ and $Q(Y \cup Z) = Y$ by Lemma 17 and the Q-independence. Monotonicity of $\overline{Q}$ implies that $Z \setminus Y \subseteq \overline{Q}(Y \cup Z)$ and $Y \setminus X \subseteq \overline{Q}(X \cup Y)$, hence $(Z \setminus Y) \cup (Y \setminus X) \subseteq \overline{Q}(X \cup Y \cup Z)$, that is, $Q(X \cup Y \cup Z) \subseteq X \subseteq X \cup Y \cup Z$. Now property (2) of Q yields that $Q(X \cup Y \cup Z) = Q(X) = X$, and from $Q(X \cup Y \cup Z) = X \subseteq X \cup Z \subseteq X \cup Y \cup Z$ it follows again by property (2) that $Q(X \cup Z) = Q(X \cup Y \cup Z) = X$, that is, $X \preceq_Q Z$. So $\preceq_Q$ is indeed a partial order on Q-independent sets. □

So far in this section we have proved that the CA-CQ problem with nested bounded sets can be formulated as a stable matching problem with increasing comonotone choice functions. It is not difficult to see that if Q is an increasing comonotone choice function then Q is *path-independent*, that is, for any X and Y in the domain of Q we have $Q(X \cup Y) = Q(Q(X) \cup Q(Y))$. Roth proved the following theorem for preference induced comonotone choice functions that are clearly path-independent [22, 23].

Theorem 19. *Assume that $G = (A \cup C, E)$ is a bipartite graph between sets A of applicants and C of colleges and edges of E represent applications. If choice functions Q_C of colleges and Q_A of applicants are both comonotone with property (2) then there is a stable matching, that is, a subset S of E such that $Q_A(S) = Q_C(S) = S$ and each edge $e \in E \setminus S$ is dominated by S.*

A possible proof of Theorem 19 is contained in [10], where a generalisation of the deferred acceptance algorithm is also described. It is slightly different from the one we need later, so here we describe a proof and an algorithm.

Proof. If choice function Q is comonotone then $\overline{Q}$ is monotone by the definition of comonotonicity. Let us define function F on subsets of E by $F(X) := E \setminus \overline{Q_A}(E \setminus \overline{Q_C}(X))$. First we prove a connection between stable matchings and fixed points of F.

Assume that S is a stable matching. Let $S_C := \{e \in E : e \in \overline{Q_C}(S \cup \{e\})$ denote the set of edges that S dominates from the college side. By the stability of S, each edge in $E \setminus (S \cup S_C)$ is dominated by S from the applicant side. The monotonicity of $\overline{Q_C}$ implies for $S' = S \cup S_C$ that $S_C \subseteq \overline{Q_C}(S')$, hence $Q_C(S') \subseteq S \subseteq S'$ holds. By the property (2) of Q_C this means that $Q_C(S') = Q_C(S) = S$, where the latter equation comes from the fact that S is a stable matching hence college-independent. In particular,

$\overline{Q_C}(S') = S_C$ holds. The above argument for choice function Q_A shows that $Q_A(S'') = S$, where $S'' = S \cup S_A$ and S_A is the set of edges that S dominates from the applicant side. As S is a stable matching, each element outside S' is dominated from the applicant side, hence $Q(S'') = S \subseteq E \setminus S_C \subseteq S \cup S_A = S''$, and property (2) of Q_A implies that $Q_A(E \setminus S_C) = Q_A(S'') = S$ thus $\overline{Q_A}(E \setminus S_C) = (E \setminus S_C) \setminus S = E \setminus (S_C \cup S) = E \setminus S'$. So

$$F(S') = F(S \cup S_C) = E \setminus \overline{Q_A}(E \setminus \overline{Q_C}(S')) = E \setminus \overline{Q_A}(E \setminus S_C) = E \setminus (E \setminus S') = S'$$

So each stable matching S determines a fixed point of F.

Now assume that $F(S') = S'$ is a fixed point. By denoting $Q_C(S')$ by S we have $\overline{Q_C}(S') = S' \setminus S$. We show that S is a stable matching. By property (2) of Q_C, $Q_C(S') = S \subseteq S'$ implies that $Q_C(S) = Q_C(S') = S$ and for $x \in S' \setminus S$ we have $Q(S') \subseteq S \cup \{x\} \subseteq S'$, so $Q_C(S \cup \{x\}) = Q_C(S') = S$, that is S dominates each edge of $S' \setminus S$ from the college side.

From $F(S') = S'$ we get that

$$F(S') = S' = E \setminus \overline{Q_A}(E \setminus \overline{Q_C}(S')) = E \setminus \overline{Q_A}((E \setminus S') \cup S) \ ,$$

hence $E \setminus S' = \overline{Q_A}((E \setminus S') \cup S)$, or, in other words $S = Q_A((E \setminus S') \cup S)$. This means that for any element x of $E \setminus S'$ we have $Q_A((E \setminus S') \cup S) = S \subseteq S \cup \{x\} \subseteq (E \setminus S') \cup S$ so property (2) implies that $S = Q_A(S \cup \{x\})$. The conclusion is that S is a stable matching.

In what follows we prove that F is monotone. If $X \subseteq Y$ then $\overline{Q_C}(X) \subseteq \overline{Q_C}(Y)$ follows by the monotonicity of $\overline{Q_C}$, thus $E \setminus \overline{Q_C}(X) \supseteq E \setminus \overline{Q_C}(Y)$. The monotonicity of $\overline{Q_A}$ implies that $\overline{Q_A}(E \setminus \overline{Q_C}(X)) \supseteq \overline{Q_A}(E \setminus \overline{Q_C}(Y))$, hence $F(X) = E \setminus \overline{Q_A}(E \setminus \overline{Q_C}(X)) \supseteq E \setminus \overline{Q_A}(E \setminus \overline{Q_C}(Y)) = F(Y)$ holds, proving that F is indeed monotone.

This follows that $\emptyset \subseteq F(\emptyset) \subseteq F(F(\emptyset)) \subseteq \ldots$ is a chain of increasing sets. As ground-set E is finite, after some iterations we find a fixed point $S^0 = F(F(\ldots(F(\emptyset))\ldots))$ of F, such that $S_0 := Q_C(S^0)$ is a stable matching. □

Actually, the set S^0 we constructed at the end of the above proof is an inclusionwise minimal fixed point of F because if $F(S^*) = S^*$ is another fixed point then $\emptyset \subseteq S^*$, hence $F(\emptyset) \subseteq F(S^*) = S^*$, so $F(F(\emptyset)) \subseteq F(S^*) = S^*$, and so on. In other words, stable matching S_0 dominates all those edges from the student side that some stable matching may dominate from the student side, that is, each applicant receives the best assignment that she can have in a stable matching. In other words, $Q_C(S)$ is the so-called *applicant-optimal* stable matching in general. This is the solution that we constructed also with the applicant-oriented algorithm in Section 4.2 for the particular choice functions of our model CA-CQ with nested set systems.

Note that there is a college-optimal stable matching, as well. To construct it, we start the iteration of F with E, rather than with $\emptyset$. So we get a decreasing chain $F(E) \supseteq F(F(E)) \supseteq F(F(F(E)))\ldots$ of subsets of E that stabilises at a fixed point $S^1 = F(F(\ldots F(E)\ldots)) = F(F(\ldots F(F(S^1))\ldots)) = F(S^1)$. This S^1 is the inclusionwise maximal fixed point of F because if $F(S^*) = S^*$ is another fixed point of F then from $S^* \subseteq E$ by monotonicity it follows that $S^* = F(F(\ldots F(S^*)\ldots)) \subseteq F(F(\ldots F(E)\ldots)) = S^1$. That is, if a stable matching dominates some edge e from the college side then e is certainly dominated by stable matching $S_1 := Q_C(S^1)$ from the college side. We call S_1 the *college-optimal* stable matching. In particular, for the special choice functions of our model CA-CQ with nested set systems, this solution is the applicant-pessimal stable matching that the college-oriented algorithm finds in Section 4.3.

How do we construct the above applicant- and college-optimal stable matchings? It is simple: we iterate function F starting from E or from $\emptyset$. Let $E_0 := E$ and for $i = 0, 1, 2, \ldots$ let $E_{i+1} := F(E_i)$. As soon as $E_{i+1} = E_i$ we can determine college-optimal stable matching

$S_1 = Q_C(E_i)$. We know that $E_0 \supseteq E_1 \supseteq \ldots$ is a decreasing chain, so to construct E_{i+1} from E_i we have to find elements of $E_i \setminus E_{i+1}$ that we have to delete from E_i. From the formula $E_{i+1} := F(E_i) = E \setminus \overline{Q_A}(E \setminus \overline{Q_C}(E_i))$ it follows that $E_{i+1} = E_i \setminus \overline{Q_A}(E \setminus \overline{Q_C}(E_i))$, so at the $(i+1)$st iteration we delete $\overline{Q_A}(E \setminus \overline{Q_C}(E_i)) = \overline{Q_A}(Q_C(E_i) \cup (E \setminus E_i))$ from E_i.

Note that the iteration of F is not exactly the Gale-Shapley algorithm. The Gale-Shapley algorithm corresponds to the recursive definition $E'_{i+1} := E'_i \setminus \overline{Q_A}(Q_C(E'_i))$. However, from property (2) of Q_A and Q_C it is not difficult to prove that $E_i = E'_i$ holds for all i.

In what follows we give a matroid-free proof showing that the function we calculate by the greedy algorithm is indeed Q_C.

5.2 The greedy algorithm for the college choice function

To apply the generalised Gale-Shapley algorithm for the CA-CQ problem, we have to calculate choice function Q_C on several arguments. Though the definition of Q_C is algorithmic, using it for its calculation is not a best choice: there is a more efficient greedy algorithm that works as follows. Order set E of edges as $E = \{e_1, e_2, \ldots, e_m\}$ such that if e_i is preceding e_j in the preference order of some quota set C_k (either because the applicant is more preferred or because the two edges belong to the same applicant who prefers e_i) then we require that $i < j$ holds. There exists such a linear order, as the groups partition the set C of colleges (for this we need the nested property of $\mathcal{C}$ and that each individual college is a member of $\mathcal{C}$) and if we merge the preference orders (on the edges) of the groups then we get the property. The alternative algorithm to construct $Q_C(X)$ is a greedy one. Let $X^i := \{x_1, x_2, \ldots, x_i\}$ denote the first i elements of X according to the above linear order. For $i = 1, 2, \ldots$, in the ith step of the algorithm we find $Q^g_C(X^i)$ and for each bounded set $C_k \in \mathcal{C}$, we calculate all numbers $f(X^i, C_k) := |Q^g_C(X^i) \cap E(C_k)|$ of edges that are accepted from X^i to some college of C_k. (Recall that here $E(C_k) := \{e = (a, c) \in E : c \in C_k\}$ denotes the set of edges to colleges of C_k.) The ith step of the algorithm is the following.

If the ith element $x_i = (a, c)$ of X together with the elements of X^{i-1} chosen so far would violate some quota, i.e. if $f(X^{i-1}, C_k) = q(C_k)$ for some $c \in C_k \in \mathcal{C}$ then $Q^g_C(X^i) = Q^g_C(X^{i-1})$ and hence $f(X^i, C_k) = f(X^{i-1}, C_k)$ for all k. Otherwise, if no quota is violated, that is, if $f(X^{i-1}, C_k) < q(C_k)$ for all $c \in C_k \in \mathcal{C}$ then $Q^g_C(X^i) = Q^g_C(X^{i-1}) \cup \{x_i\}$. Saturations are $f(X^i, C_k) = f(X^{i-1}, C_k) + 1$ if $c \in C_k$ and $f(X^i, C_t) = f(X^{i-1}, C_t)$ if $c \notin C_t$. The greedy algorithm outputs $Q^g_C(X) = Q^g_C(X^l)$ where l is the size of X, hence $X = X^l$.

Lemma 20. *If Q_C is the choice function of a college then the above greedy algorithm calculates the correct value: $Q_C(X) = Q^g_C(X)$ for any subset X of E.*

Proof. To this end, it is enough to show that $Q_C(X^i) = Q^g_C(X^i)$ for $i = 1, 2, \ldots l$. Clearly, for $i = 1$, we chose x_1 in both algorithms if x_1 itself does not violate any of the quotas, so $Q^g_C(X^1) = Q_C(X^1)$. For the induction proof, assume that $Q^g_C(X^i) = Q_C(X^i)$, and consider $X^{i+1} = X^i \cup \{x_{i+1}\}$. From the description of the algorithms, it follows easily that $Q_C(X^i) = Q_C(X^{i+1}) \cap X^i$ and $Q^g_C(X^i) = Q^g_C(X^{i+1}) \cap X^i$, so we only have to prove that $x_{i+1} \in Q_C(X^{i+1})$ if and only if $x_{i+1} \in Q^g_C(X^{i+1})$.

Clearly, if $x_{i+1} \in Q_C(X^{i+1})$ then $(X^{i+1})'_k = \emptyset$ for all sets $C_k \in \mathcal{C}$ with $x_{i+1} \in E(C_k)$. Hence $Q_C(X^i) = Q_C(X^{i+1}) \setminus \{x_{i+1}\} = Q^g_C(X^{i+1}) \setminus \{x_{i+1}\}$ does not fill up any quota $q(C_k)$ for $x_{i+1} \in E(C_k)$. But this implies that $x_{i+1} \in Q^g_C(X^{i+1})$. On the other hand, if $x_{i+1} \notin Q_C(X^{i+1})$ then there is a member C_k of $\mathcal{C}$ such that $x_{i+1} \in (X^{i+1})'_k$, i.e. x_{i+1} does not fit in with quota $q(C_k)$. So either $Q_C(X^i) = Q^g_C(X^i)$ fills up the bounded set C_k, hence $x_{i+1} \notin Q^g_C(X^{i+1})$, or, if $|Q_C(X^i) \cap E(C_k)| < q(C_k)$ then this is because there is some other bounded set $C_t \in \mathcal{C}$ with $C_k \subset C_t$ such that C_t is full: $|Q_C(X^i) \cap E(C_t)| = q(C_t)$. As $x_{i+1} \in E(C_t)$ holds again, $x_{i+1} \notin Q^g_C(X^{i+1})$ follows. □

5.3 Structural results

Several interesting properties about the structure of stable matchings are known. Knuth [17] attributes to Conway the observation that stable marriages have a lattice structure: if each man picks the better assignment out of two stable matchings then another stable matching is created in which each woman gets the worse husband from the two. Using linear programming tools, Teo and Sethuraman [26] proved a generalisation of this, namely, that if k stable matchings are given and each man selects the lth best partner then a stable matching is created in which each woman receives her $(k+1-l)$th husband. Later this result was further generalised by Fleiner [9] for the many-to-many model and the proof was based on the lattice property. Even later, Klaus and Klijn [16] found the same proof for the many-to-one case.

Another interesting consequence of the lattice structure is the "Rural hospitals theorem" of Gale and Sotomayor [12] stating that if in the college admissions problem some college c does not fill up its quota in a stable matching then college c receives the same set of applicants in each stable matching. Our goal in this section is to extend the above properties to the CA-CQ problem. Note that the "natural" CA-CQ extension of the "Rural hospitals theorem" does not hold as we saw in Example 6. (There, college c_1 is assigned a number of students equal to its quota in one stable matching but a smaller number in another one. The example also shows that in contrast to the case of CA, in the CA-CQ problem it is not true that in the college-optimal stable matching each college gets the best set of students.)

Fleiner studied stable matching problems with comonotone set functions in [10]. Corollaries 26 and 27 of [10] yield the following result.

Theorem 21. *Assume that $G = (A \cup C, E)$ is a bipartite graph with colour classes C and A and that $\mathcal{M}_C$ and $\mathcal{M}_A$ are two matroids on E. Let Q_C and Q_A be increasing comonotone choice functions on E that are defined by the greedy algorithm on matroids $\mathcal{M}_C$ and $\mathcal{M}_A$ respectively, for some linear orders.*

(i) If $M_1, M_2, \ldots, M_k$ are stable matchings then $Q_C(\bigcup_{i=1}^k M_i)$ and $Q_A(\bigcup_{i=1}^k M_i)$ are also stable matchings.

(ii) For any two stable matchings M and M', $span_{\mathcal{M}_C}(M) = span_{\mathcal{M}_C}(M')$ and $span_{\mathcal{M}_A}(M) = span_{\mathcal{M}_A}(M')$ holds.

Obviously, Theorem 21 holds for the CA-CQ problem. But observe that it is more general than that. Theorem 21 is also true if (say) applicants can be assigned to several colleges up to certain personal quotas, and there can be nested bounded sets on the applicant side as well, just as for colleges. Theorem 19 implies for this case, too, that there exists a stable matching, and Theorem 21 concerns the structure of these stable matchings.

The first part of Theorem 21 implies our earlier observation on the applicant-optimal and college-optimal stable matchings, namely, if X is the union of all stable matchings then $M_C := Q_C(X)$ and $M_A := Q_A(X)$ are stable matchings. As M_C and M_A dominate any other stable matching from the college side and from the applicant side, respectively, it follows that M_C is the college optimal stable matching and M_A is the applicant-optimal one. This observation implies Theorem 6 (ii) and Theorem 9 (ii) and (iii).

But the first part of Theorem 21 has an even more interesting consequence, as this generalises the lattice property of stable marriages. If the colleges freely choose from the union of the edge sets of two stable matchings M_1 and M_2, then by Theorem 21 another stable matching is created that we denote by $M_1 \vee M_2$. Clearly, $M_1 \vee M_2$ dominates both M_1 and M_2 from the college side. Moreover, if a stable matching M dominates both M_1 and M_2 from the college side then M dominates $M_1 \cup M_2$ from the college side, hence M also dominates $M_1 \vee M_2$ from the college side. From this fact it follows that $M_1 \vee M_2$ is the least upper bound of M_1 and M_2 as $\preceq_C$ is a partial order on stable matchings by Theorem 18.

A similar proof shows that for stable matchings M_1 and M_2 stable matching $M_1 \wedge M_2 := Q_A(M_1 \cup M_2)$ is the least upper bound of M_1 and M_2 for partial order $\preceq_A$. The following lemma gives a relation the above two partial orders.

Lemma 22. *Assume that $G = (A \cup C, E)$ is a bipartite graph with colour classes A and C and that $\mathcal{M}_A$ and $\mathcal{M}_C$ are two matroids on E. Let Q_A and Q_C be comonotone choice functions with property (2) on E that are defined by the greedy algorithm on $\mathcal{M}_A$ and $\mathcal{M}_C$ respectively, for some linear orders. Then for any two stable matchings M_1 and M_2 relation $M_1 \preceq_A M_2$ is equivalent to $M_2 \preceq_C M_1$, or, in other words partial orders $\preceq_A$ and $\preceq_C$ are opposite on stable matchings.*

Proof. If M_1 and M_2 are stable matchings then each of them must dominate the other one. If M_1 dominates each element of $M_2 \setminus M_1$ from the applicant side then M_2 must dominate each element of $M_1 \setminus M_2$ from the college side, and vice verse. The lemma directly follows from this observation. □

Lemma 22 implies that partial order $\preceq_C$ forms a lattice on stable matchings with lattice operations $\vee$ and $\wedge$: we have seen that $M_1 \vee M_2$ is the $\preceq_C$-least upper bound and $M_1 \wedge M_2$ is the $\preceq_A$-least upper bound, hence by Lemma 22 $M_1 \wedge M_2$ is the $\preceq_C$-greatest lower bound of M_1 and M_2.

The lattice structure of stable matchings allows us to prove an extension of the result of Teo and Sethuraman [26]. Our proof is essentially the same as the one in [9] that has also been found by Klaus and Klijn [16] for the many-to-one case.

Theorem 23. *Assume that the* CA-CQ *problem is given by bipartite graph $G = (A \cup C, E)$ with colour classes A and C, a nested system $\mathcal{C}$ of bounded sets and quotas $q : \mathcal{C} \to \mathbb{N}$. Let $M_1, M_2, \ldots, M_n$ be arbitrary stable matchings and $1 \leq k \leq n$ an arbitrary integer. If each applicant chooses her k-th best assignment out of the n assignments provided by stable matchings M_i then the set M of these edges is a stable matching.*

Proof. Let each applicant a order the n stable matchings according to her preference as $M_a^1, M_a^2, \ldots, M_a^n$ such that a prefers M_a^1 the best and M_a^n the least. Let

$$M := \bigwedge_{a \in A} M_a = Q_A(\bigcup_{a \in A} M_a) \qquad \text{where} \qquad M_a := \bigvee_{i=1}^{k} M_a^i = Q_C(\bigcup_{i=1}^{k} M_a^i) \ .$$

Clearly, each of M and the M_a's are a stable matchings by Theorem 21 (i). Observe that M_a dominates each of $M_a^1, M_a^2, \ldots, M_a^k$ from the college side, hence M_a is dominated by these matchings from the applicant side. This means that in M_a each applicant a' of A receives her worst assignment out of her assignments in $M_a^1, M_a^2, \ldots, M_a^k$. In particular, applicant a receives her kth best assignment out of the ones given by $M_1, M_2, \ldots, M_n$. Any other applicant a' gets the kth best out of k assignments which is an assignment that is not better than the one represented by $M_{a'}^k$. As M is constructed by letting the applicants to choose from $\bigcup_{a \in A} M_a$, each applicant a will choose her kth best assignment represented by M_a^k. That is, M is the stable matching described in the theorem. □

The second part of Theorem 21 talks about the span on a subset of a matroid. Recall that an element e of some matroid $\mathcal{M}$ is spanned by subset E' of $\mathcal{M}$ if either $e \in E'$ or there is an independent subset E^* of E' such that any proper subset of $E^* \cup \{e\}$ is independent. This is equivalent to saying that the rank $r_{\mathcal{M}}(E')$ of E' equals the rank $r_{\mathcal{M}}(E' \cup \{e\})$, that is, in case of (say) $\mathcal{M}_C$, we have $|Q_C(E')| = |Q_C(E' \cup \{e\})|$. This latter formulation implies that if some stable matching M dominates e from the college side then $e \in span_{\mathcal{M}_C}(M)$. Recall that $E'(a)$, $E'(c)$ and $E'(C_i)$ denote the set of edges of E' that belong to applicant a, to college c and to some bounded set of colleges C_i, respectively.

What does it mean that some set E' of edges spans a certain edge e in the CA-CQ problem? Clearly, for matroid $\mathcal{M}_A$ it happens if and only if applicant-vertex a of edge e is incident with an edge (e or some other) of E', i.e., if $|E'(a)| \geq 1$. For $\mathcal{M}_C$ this means that either $e \in E'$ or there is a bounded set C_i containing c such that $|E'(C_i)| \geq q(C_i)$. Recall that bounded set C_i is full relative to matching M if $|M(C_i)| = q(C_i)$, and otherwise C_i is undersubscribed. Recall that a bounded set is free if it is not contained in a full bounded set. We call bounded set C_i *weakly free relative to* E' if no bounded set that properly contains C_i is full relative to E'. (In particular, a weakly free set relative to E' can be full relative to E'.) Bounded set C_i is *essential relative to* E' if C_i is weakly free and no bounded subset of C_i is full relative to E'. (In particular an essential bounded set cannot be full relative to E'.) We have the following generalisation of the "Rural hospitals theorem".

Theorem 24. *Let M be a stable matching in an instance of* CA-CQ. *If bounded set C_i is weakly free relative to M then for each stable matching M' we have $|M(C_i)| = |M'(C_i)|$. Moreover, if C_i is essential relative to M then $M(C_i) = M'(C_i)$.*

In other words, if a bounded set is not properly contained in a full bounded set then it is incident with the same number of edges in each stable matching. Moreover, if this bounded set does not contain a full bounded set then it is incident with the same set of edges in each stable matching.

Note that Theorem 24 remains true if applicants also have a college-type choice function as the proof only uses Theorem 21 and the structure of matroid $\mathcal{M}_C$.

Proof. We prove Theorem 24 by induction on i. (Recall that we fixed an order $C_1, C_2, \ldots$ of bounded sets such that a superset always has a greater index.) If C_1 is not weakly free relative to M then we have nothing to prove. If C_1 is free (that is weakly free and undersubscribed) relative to M then C_1 is essential relative to M as C_1 does not contain any other bounded sets. This means that $span_{\mathcal{M}_C}(M) \cap E(C_1) = M(C_1)$. So if M' is another stable matching then by Theorem 21,

$$M(C_1) = span_{\mathcal{M}_C}(M) \cap E(C_1) = span_{\mathcal{M}_C}(M') \cap E(C_1) = M'(C_1)$$

just as we claimed. Now suppose that C_1 is full relative to M and assume for a contradiction that $|M(C_1)| \neq |M'(C_1)|$. From $|M(C_1)| = q(C_1) \geq |M'(C_1)|$, it follows that $|M'(C_1)| < q(C_1)$, hence C_1 is undersubscribed in stable matching M'. But we just have proved that in this case C_1 cannot be full in any stable matching, a contradiction.

Now assume that Theorem 24 is true for each C_j with $j < i$ and we prove it for C_i. If C_i is an inclusionwise minimal bounded set then the proof is exactly the same as for C_1 above. Otherwise $C_i = C^1 \cup C^2 \cup \ldots \cup C^k$, where $C^1, C^2, \ldots, C^k$ are the inclusionwise maximal bounded subsets of C_i that are disjoint by the nested property. We may assume that C_i is weakly free relative to M as otherwise we have nothing to prove.

Suppose first that C_i is undersubscribed relative M. Clearly, each of $C^1, C^2, \ldots, C^k$ is weakly free relative to M, so Theorem 24 holds for them: $|M(C^t)| = |M'(C^t)|$ for any stable matching M'. But this means that

$$|M(C_i)| = \sum_{t=1}^{k} |M(C^t)| = \sum_{t=1}^{k} |M'(C^t)| = |M'(C_i)| \ .$$

If C_i is essential relative to M then all of $C^1, C^2, \dots C^k$ are also essential relative to M, hence

$$M(C_i) = \bigcup_{t=1}^{k} M(C^t) = \bigcup_{t=1}^{k} M'(C^t) = M'(C_i)$$

holds. This finishes the case when C_i is undersubscribed in M.

It remains to settle the case when C_i is full relative to M. If, indirectly C_i is undersubscribed in some other stable matching M' then the above argument also holds for M' instead of M. This yields in particular that C_i cannot be full relative to any stable matching, contradicting the existence of M. ☐

Theorem 24 is a genuine generalisation of the "Rural hospitals theorem" as in the absence of common quotas, each college c is a weakly free bounded set by itself, so each stable matching assigns the same number of applicants to c. Moreover, if college c does not fill up its quota in some stable matching then bounded set $\{c\}$ is essential hence c gets the same set of applicants in each stable matching.

Actually, in the college admissions problem each college is a group by itself. Interestingly, in the CA-CQ problem groups behave somewhat similarly to colleges in the College Admissions problem.

Corollary 25. *If C_i is a group in the* CA-CQ *problem then any stable matching assigns the same number of edges to C_i.*

Proof. A group is weakly free by definition, so the corollary follows from Theorem 24. ☐

5.4 Comparison between the direct and the matroid approaches

In Section 4, we constructed and studied two direct algorithms for solving the College Admissions problem with common quotas in the case of nested set systems, whilst in Section 5, we solved the same problem by a more general approach using choice functions and matroids. The reader might wonder why both descriptions are necessary. Here, we would like to answer this hypothetical question.

We proved the existence of a stable matching by each method, and we also showed that the two main variants of each method produce the applicant-optimal and the applicant-pessimal solutions, respectively. This latter fact ensures that the corresponding variants lead to the very same results. The reason for studying both methods, beside the obvious interest in having both a direct and a general argument, is the following. On the one hand, the direct algorithms have complexities which are not achievable by the general method. On the other hand, the structural results which are straightforward by the matroid model would be difficult to prove directly. A further advantage of the general approach is that, as we noted, similar results can be verified for such more general settings where the choice function on the the applicants' side is more complicated than the one in our model.

We note that the original algorithm of Gale and Shapley [11] and its variant studied by McVitie and Wilson [19] have a similar relation to each other as our general and direct method. Both produce the same results, but the $O(L)$ running time is achievable only with the latter variant, where the proposals are made one by one rather than simultaneously.

Acknowledgments

We would like to thank the Hungarian Higher Education National Institute and one of its representatives in particular, Gábor Varjasy, for providing us information about the higher education matching scheme, including a detailed description of their current algorithm and the real data of the 2008 matching run. Also, we would like to thank an anonymous referee for helpful comments.

References

[1] A. Abdulkadiroğlu, P. A. Pathak, and A. E. Roth. The New York City high school match. *American Economic Review, Papers and Proceedings*, 95(2):364–367, 2005.

[2] A. Abdulkadiroğlu, P. A. Pathak, A. E. Roth, and T. Sönmez. The Boston public school match. *American Economic Review, Papers and Proceedings*, 95(2):368–371, 2005.

[3] D.J. Abraham, R.W. Irving, and D.F. Manlove. Two algorithms for the student-project allocation problem. *Journal of Discrete Algorithms*, 5(1):79–91, 2007.

[4] M. Balinski and T. Sönmez. A tale of two mechanisms: student placement. *J. Econom. Theory*, 84(1):73–94, 1999.

[5] P. Biró. Student admissions in Hungary as Gale and Shapley envisaged. Technical report, no. TR-2008-291, University of Glasgow, Department of Computing Science, 2008.

[6] S. Braun, N. Dwenger, and D. Kübler. Telling the truth may not pay off: An empirical study of centralised university admissions in Germany. *SFB 649 Discussion Papers*, Humboldt University, Berlin, Germany, 2007.

[7] G. Cornuéjols. General factors of graphs. *J. Combin. Theory Ser. B*, 45(2):185–198, 1988.

[8] M. E. Dyer and A. M. Frieze. Planar 3DM is NP-complete. *J. Algorithms*, 7(2):174–184, 1986.

[9] T. Fleiner. Some results on stable matchings and fixed points. Technical report, EGRES report TR-2002-8, ISSN 1587-4451, December 2002. http://www.cs.elte.hu/egres.

[10] T. Fleiner. A fixed-point approach to stable matchings and some applications. *Math. Oper. Res.*, 28(1):103–126, 2003.

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

[12] D. Gale and M. Sotomayor. Some remarks on the stable matching problem. *Discrete Applied Mathematics*, 11:223–232, 1985.

[13] M. R. Garey and D. S. Johnson. *Computers and intractability.* W. H. Freeman and Co., San Francisco, Calif., 1979.

[14] D. Gusfield and R.W. Irving. *The Stable Marriage Problem: Structure and Algorithms.* MIT Press, 1989.

[15] K. Hamada, K. Iwama, and S. Miyazaki. The Hospitals/Residents problem with quota lower bounds. In *Proceedings of MATCH-UP 2008: Matching Under Preferences, satellite workshop of ICALP 2008*, pages 55–66. 2008.

[16] B. Klaus and F. Klijn. Median stable matching for college admissions. *Internat. J. Game Theory*, 34(1):1–11, 2006.

[17] D.E. Knuth. *Mariages Stables.* Les Presses de L'Université de Montréal, 1976.

[18] D.F. Manlove, R.W. Irving, K. Iwama, S. Miyazaki, and Y. Morita. Hard variants of stable marriage. *Theoretical Computer Science*, 276(1-2):261–279, 2002.

[19] D. G. McVitie and L. B. Wilson. Stable marriage assignment for unequal sets. *BIT*, 10:295–309, 1970.

[20] A. Romero-Medina. Implementation of stable solutions in a restricted matching market. *Review of Economic Design*, 3(2):137–147, 1998.

[21] A. E. Roth. The evolution of the labor market for medical interns and residents: a case study in game theory. *Journal of Political Economy*, 6(4):991–1016, 1984.

[22] A. E. Roth. Stability and polarization of interests in job matching. *Econometrica*, 52(6):47–57, 1984.

[23] A. E. Roth. Conflict and coincidence of interest in job matching: some new results and open questions. *Math. Oper. Res.*, 10(3):379–389, 1985.

[24] A.E. Roth and M.A.O. Sotomayor. *Two-sided matching: a study in game-theoretic modeling and analysis*, volume 18 of *Econometric Society Monographs*. Cambridge University Press, 1990.

[25] A. Schrijver. *Combinatorial optimization. Polyhedra and efficiency. Vol. B*, volume 24 of *Algorithms and Combinatorics*. Springer-Verlag, Berlin, 2003. Matroids, trees, stable sets, Chapters 39–69.

[26] C.-P. Teo and J. Sethuraman. The geometry of fractional stable matchings and its applications. *Math. Oper. Res.*, 23(4):874–891, 1998.