Pareto optimal matchings of students to courses in the presence of prerequisites

Katarína Cechlárová, Bettina Klaus, and David F. Manlove

Abstract

We consider the problem of allocating applicants to courses, where each applicant has a subset of acceptable courses that she ranks in strict order of preference. Each applicant and course has a *capacity*, indicating the maximum number of courses and applicants they can be assigned to, respectively. We thus essentially have a many-to-many bipartite matching problem with one-sided preferences, which has applications to the assignment of students to optional courses at a university.

We consider additive preferences and lexicographic preferences as two means of extending preferences over individual courses to preferences over bundles of courses. We additionally focus on the case that courses have prerequisite constraints: we will mainly treat these constraints as compulsory, but we also allow alternative prerequisites. We further study the case where courses may be corequisites.

For these extensions to the basic problem, we present the following algorithmic results, which are mainly concerned with the computation of Pareto optimal matchings (POMs). Firstly, we consider compulsory prerequisites. For additive preferences, we show that the problem of finding a POM is NP-hard. On the other hand, in the case of lexicographic preferences we give a polynomial-time algorithm for finding a POM, based on the well-known sequential mechanism. However we show that the problem of deciding whether a given matching is Pareto optimal is co-NP-complete. We further prove that finding a maximum cardinality (Pareto optimal) matching is NP-hard. Under alternative prerequisites, we show that finding a POM is NP-hard for either additive or lexicographic preferences. Finally we consider corequisites. We prove that, as in the case of compulsory prerequisites, finding a POM is NP-hard for additive preferences, though solvable in polynomial time for lexicographic preferences. In the latter case, the problem of finding a maximum cardinality POM is NP-hard and very difficult to approximate.

1 Introduction

Problems involving the allocation of indivisible goods to agents have gained a lot of attention in literature, since they model many real scenarios, including the allocation of pupils to study places [4], workers to positions [22], researchers to projects [29], tenants to houses [1] and students to courses [10], etc. We assume that agents on one side of the market (pupils, workers, researchers, tenants, students) have preferences over objects on the other side of the market (study places, positions, projects, courses, etc.) but not vice versa. In such a setting where the desires of agents are in general conflicting, economists regard Pareto optimality (or Pareto efficiency) as a basic, fundamental criterion to be satisfied by an allocation. This concept guarantees that no agent can be made better off without another agent becoming worse off. A popular and very intuitive approach to finding Pareto optimal matchings is represented by the class of *sequential allocation mechanisms* [24, 9, 8, 3].

In the one-to-one case (each agent receives at most one object, and each object can be assigned to at most one agent) this mechanism has been given several different names in the literature, including serial dictatorship [1, 28], queue allocation [34], Greedy-POM [2], sequential mechanism [8, 3] etc. Several authors independently proved that a matching is Pareto optimal if and only if can be obtained by the serial dictatorship mechanism (Svensson in 1994 [34], Abdulkadiroğlu and Sönmez in 1998 [1], Abraham et al. in 2004 [2], and Brams and King in 2005 [9]).

In general many-to-many matching problems (agents can receive more than one object, and objects can be assigned to more than one agent), the sequential allocation mechanism works as follows: a central authority decides on an ordering of agents (often called a *policy*) that can contain multiple copies of an agent (up to her capacity). According to the chosen policy, an agent who has her turn chooses her most preferred object among those that still have a free slot. This approach was used in [3, 8], where the properties of the obtained allocation with respect to the chosen policy and strategic issues are studied.

The serial dictatorship mechanism is a special case of the sequential allocation mechanism where the policy contains each agent exactly once, and when agent a is dealt with, she chooses her entire most-preferred bundle of objects. The difficulty with serial dictatorship is that it can output a matching that is highly unfair. For example, it is easy to see that if there are two agents a_1 and a_2, each with the same preference list over objects, and each with the same capacity, which is equal to the length of their preference list, and each object has capacity 1, the mechanism will give all acceptable objects to a_i and no objects to a_{3-i} when the policy indicates that a_i should choose first ($i \in \{1, 2\}$).

In this paper we shall concentrate on one real-life application of this allocation problem that arises in education, and so our terminology will involve *applicants* (students) for the agents and *courses* for the objects. In most universities students have some freedom in their choice of courses, but at the same time they are bound by the rules of the particular university. A detailed description of the rules of the allocation process and the analysis of the behaviour of students at Harvard Business School, based on real data, is provided by Budish and Cantillon [10]. They assume that students have a linear ordering of individual courses and their preferences over bundles of courses are responsive to these orderings. The emphasis in [10] was on strategic questions. The empirical results confirmed the theoretical findings that, loosely speaking, dictatorships (where students choose one at a time their entire most preferred available bundle) are the only mechanisms that are strategy-proof and ex-post Pareto efficient.

Another field experiment in course allocation is described by Diebold et al. [14]. The authors compared the properties of allocations obtained by the sequential allocation mechanism where the policy is determined by the arrival time of students (i.e., first-come first-served) and by two modifications of the Gale-Shapley student-optimal mechanism, i.e., they assumed that courses may also have preferences or priorities over students. Moreover, they only considered the case when each student can be assigned to at most one course.

In reality, a student can attend more courses, but not all possible bundles are feasible for her. Cechlárová et al. [12] considered explicit notions of feasibility for bundles of courses. For these feasibility concepts, a given bundle can be checked for feasibility for a given applicant in time polynomial in the number of courses. Such an algorithm may check for example if no two courses in the bundle are scheduled at the same time, or if the student has enough budget to pay the fees for all the courses in the bundle, etc. Cechlárová et al. [12] showed that a sufficient condition for a general sequential allocation mechanism to output a Pareto optimal matching is that feasible bundles of courses form families that are closed with respect to subsets, and preferences of students over bundles are lexicographic. They also showed that under these assumptions a converse result holds, i.e., each Pareto optimal matching can be obtained by the sequential allocation mechanism for a suitable policy.

Prerequisites and corequisites

In this paper we deal with prerequisite and corequisite constraints. Prerequisite constraints model the situation where a student may be allowed to subscribe to a course c only if she subscribes to a set C' of other course(s). The courses in C' are usually called *prerequisite courses*, or *prerequisites*, for c. For example, at a School of Mathematics, an Optimal Control Theory course may have as its prerequisites a course on Differential Equations as well as a course in Linear Algebra; a prerequisite for a Differential Equations course could be a Calculus course, etc. On the other hand corequisite constraints model the situation where a student takes course c_1 if and only if she takes course c_2. These courses are referred to as *corequisite courses*, or *corequisites.* For example, a corequisite constraint may act on two courses: one that is a theoretical programming course and the other that is a series of corresponding programming lab sessions.

We consider three different models involving prerequisite and corequisite constraints. The first model involves compulsory prerequisites, but we allow the possibility that different students may have different prerequisite constraints. For example, for a doctoral study program in Economics, an economics graduate may have as a prerequisite a mathematical course and, on the other hand, a mathematics graduate may have as a prerequisite a course on microeconomics, etc. The second model concerns *alternative prerequisites.* Here it is assumed that certain courses require that a student subscribes to at least one of a set of other courses. For example, a course in mathematical modelling may require that a student attends one of a range of courses that deal with a specific mathematical modelling software package, such as Maple, MATLAB or Mathematica. Finally, the third model considers corequisites. Here we assume that constraints on corequisite courses are identical for all applicants.

As we assume that applicants express their preferences only over individual courses, a suitable extension of these preferences to preferences over bundles of courses has to be chosen. Among the most popular extensions are *responsive* preferences [31]. That is, an applicant has responsive preferences over bundles of courses if bundle C' is preferred to bundle C'' whenever C' is obtained from C'' by replacing some course in C'' by a more preferred course not contained in C''. Responsiveness is a very mild requirement and responsive preferences form a very wide and variable class. Therefore we shall restrict our attention to two specific examples, namely *additive* [5, 8, 10] and *lexicographic* [16, 5, 33, 32, 35] preferences. Although lexicographic preferences can be modelled as additive preferences by choosing appropriate weights [8], we would like to avoid this approach as it requires very large numbers, moreover, assuming lexicographic preferences from the outset leads to more straightforward algorithms.

To the best of our knowledge, matchings with prerequisite constraints have not been studied yet from an algorithmic perspective. Some connections can be found in the literature on scheduling with precedence constraints [26], but, unlike in the scheduling domain, there is no common optimality criterion for all the agents, since their desires are often conflicting and all have to be taken into account.

We would however like to draw the reader's attention to the works of Guerin et al. [19] and Dodson et al. [15], who analyze a version of a course selection problem in greater depth, using also probabilistic methods. Their work is a part of a larger research programme that involves advising college students about what courses to study and when, taking into account not only the required course prerequisites, but also the students' course histories and obtained grades. Based on this information, the authors try to estimate a student's ability to take multiple courses concurrently, with the goal to optimise the student's total expected utility and her chances of moving successfully toward graduation. Guerin et al. and Dodson et al. consider also temporal factors, meaning that a student can only take a certain course in the current semester if she has completed the necessary prerequisites during previous semesters. By contrast, here we assume that students choose all their courses as well as their necessary prerequisites simultaneously, and we concentrate on computational problems connected with producing a matching that fulfils a global welfare criterion.

Our contribution

As mentioned above, we will formally introduce three possible models of course allocation involving prerequisites or corequisites. In the first case the prerequisites are antisymmetric and compulsory (i.e., a constraint might ensure that an applicant can subscribe to course c_1 only if she also subscribes to course c_2, but she can attend c_2 without attending c_1). For additive preferences we show that computing a Pareto optimal matching is an NP-hard problem. In the case of lexicographic preferences we illustrate that the simple sequential allocation mechanism and its natural modification may output a matching that either violates prerequisites or Pareto optimality. Therefore we stipulate that on her turn, an applicant chooses her most preferred course together with all the necessary prerequisites. Still, it is impossible to obtain each possible Pareto optimal matching in this way. It is also unlikely that an efficient algorithm will be able to produce all Pareto optimal matchings, since the problem of checking whether a given matching admits a Pareto improvement is NP-complete. Considering structural properties of Pareto optimal matchings, it is known that finding a Pareto optimal matching with minimum cardinality is NP-hard, even in the very restricted one-to-one model (naturally without prerequisites) [2], but here we show that the problem of finding such a matching with maximum cardinality is also NP-hard.

The second model involving alternative prerequisites (i.e., where a constraint takes the form that an applicant can attend course c_1 only if she also attends either course c_2 or course c_3) seems to be computationally the most challenging case. We show that although a Pareto optimal matching always exists, it cannot be computed efficiently unless P=NP, both under additive as well as lexicographic preferences of applicants.

For the third case with corequisites (i.e., an applicant can attend a course if and only if she attends all its corequisites) we propose another modification of the sequential allocation mechanism for finding Pareto optimal matchings. If the corequisites for all the applicants are the same, the model is closely related to matchings with sizes [6] or many-to-many matchings with price-budget constraints [12], and we strengthen the existing results by showing that the problem of finding a maximum size Pareto optimal matching is not approximable within $N^{1-\varepsilon}$, for any $\varepsilon > 0$, unless P=NP, where N is the total capacity of the applicants.

The organisation of the paper is as follows. In Section 2 we give formal definitions of the problem models and define relevant notation and terminology. Sections 3, 4 and 5 deal separately with the three different models, involving compulsory prerequisites, alternative prerequisites and corequisites respectively. Finally, Section 6 concludes with some open problems and possible directions for future research. All proofs and examples appear in Appendices A and B respectively.

2 Definitions and notation

2.1 Basic Course Allocation problem

An instance of the *Course Allocation problem* (CA) involves a set $A = \{a_1, a_2, \dots, a_{n_1}\}$ of *applicants* and a set $C = \{c_1, c_2, \dots, c_{n_2}\}$ of *courses.* Each course $c_j \in C$ has a capacity $q(c_j)$ that denotes the maximum number of applicants that can be assigned to c_j. Similarly each applicant $a_i \in A$ has a capacity $q(a_i)$ denoting the maximum number of courses that she can attend. The vector $\mathbf{q}$ denotes applicants' and courses' capacities. Moreover a_i has a strict linear order (preference list) $P(a_i)$ over a subset of C. We shall represent a_i's preferences $P(a_i)$ as a simple ordered list of courses, from the most preferred to the least preferred one. With some abuse of notation, we shall say that a course c_j is *acceptable* to applicant a_i if $c_j \in P(a_i)$, otherwise c_j is *unacceptable* to a_i. $\mathcal{P}$ denotes the n_1-tuple of applicants' preferences.Thus altogether, the tuple $I = (A, C, \mathbf{q}, \mathcal{P})$ constitutes an instance of CA.

An *assignment* M is a subset of $A \times C$. The set of applicants assigned to a course $c_j \in C$ will be denoted by $M(c_j) = \{a_i \in A : (a_i, c_j) \in M\}$ and similarly, the *bundle* of courses assigned to an applicant a_i is $M(a_i) = \{c_j \in C : (a_i, c_j) \in M\}$. An assignment M is a *matching* if, for each $a_i \in A$, $M(a_i) \subseteq P(a_i)$ and $|M(a_i)| \leq q(a_i)$, and for each $c_j \in C$, $|M(c_j)| \leq q(c_j)$. In the presence of prerequisites and corequisites, additional *feasibility* constraints are to be satisfied by a matching, which will be defined below.

An applicant $a_i \in A$ has *additive preferences* over bundles of courses if she has a utility $u_{a_i}(c_j)$ for each course $c_j \in C$, and she prefers a bundle of courses $C_1 \subseteq C$ to another bundle $C_2 \subseteq C$ if and only if $\sum_{c_j \in C_1} u_{a_i}(c_j) > \sum_{c_j \in C_2} u_{a_i}(c_j)$.

Applicant a_i compares bundles of courses *lexicographically* if, given two different bundles $C_1 \subseteq C$ and $C_2 \subseteq C$ she prefers C_1 to C_2 if and only if her most preferred course in the symmetric difference $C_1 \oplus C_2$ belongs to C_1. Notice that the lexicographic ordering on bundles of courses generated by a strict preference order $P(a_i)$ is also strict.

Applicant a_i prefers matching M' to matching M if she prefers $M'(a_i)$ to $M(a_i)$. We say that a matching M' *(Pareto) dominates* a matching M if at least one applicant prefers M' to M and no applicant prefers M to M'.

A *Pareto optimal matching* (or *POM* for short), is a matching that is not (Pareto) dominated by any other matching. As the dominance relation is a partial order over $\mathcal{M}$, the set of all matchings in I, and $\mathcal{M}$ is finite, a POM exists for each instance of CA.

2.2 Compulsory prerequisites

We now define the first extension of CA involving compulsory prerequisites. Suppose that for each applicant $a_i \in A$, there is a partial order $\rightarrow_{a_i}$ on the set of courses C (i.e., $\rightarrow_{a_i}$ is a reflexive, antisymmetric and transitive binary relation) representing the prerequisites of applicant a_i. It is easy to see that this partial order can be fully specified if for each $c_j \in C$ its *immediate* prerequisites for a_i are given.We now define the *feasibility* of a bundle of courses relative to the constraints on compulsory prerequisites.

Definition 1. *A bundle of courses $C' \subseteq C$ is* feasible *for an applicant $a_i \in A$ if the following three conditions are fulfilled:*

(i) $C' \subseteq P(a_i)$;

(ii) $|C'| \leq q(a_i)$;

(iii) C' *fulfils a_i's prerequisites, i.e., for each $c_j, c_k \in C$, if $c_j \in C'$ and $c_j \rightarrow_{a_i} c_k$ then $c_k \in C'$.*

A subset C' of a partially-ordered set C fulfilling condition (*iii*) is called a *down-set* (see [13]). We shall denote by $\overset{\rightarrow_{a_i}}{c_j}$ the inclusion minimal down-set of C (with respect to a_i's prerequisites) that contains course c_j.

For technical reasons, we assume that $\overset{\rightarrow_{a_i}}{c_j} \subseteq P(a_i)$ for each $a_i \in A$ and each $c_j \in P(a_i)$. If this is not the case then we can easily modify the preference list of applicant a_i either by deleting a course c_j if $P(a_i)$ does not contain all the courses in $\overset{\rightarrow_{a_i}}{c_j}$, or we can append the missing courses to the end of $P(a_i)$.

An instance I of the *Course Allocation problem with (compulsory) PRequisites* (CAPR) comprises a tuple $I = (A, C, \mathbf{q}, \mathcal{P}, \rightarrow)$, where $(A, C, \mathbf{q}, \mathcal{P})$ is an instance of CA and $\rightarrow$ is the n_1-tuple of prerequisite partial orders $\rightarrow_{a_i}$ for each applicant $a_i \in A$. In an instance of CAPR, a *matching* M is as defined in the CA case, together with the additional property that, for each applicant $a_i \in A$, $M(a_i)$ is feasible for a_i.

In an instance of CAPR where the prerequisites are the same for all applicants, we may drop the applicant subscript when referring to the prerequisite partial order $\rightarrow$.

2.3 Alternative prerequisites

The second model is a variant of CAPR in which prerequisites need not be compulsory but are in general presented in the form of alternatives. Formally, each applicant a_i has a mapping $\mapsto_{a_i}: C \to 2^C$ with the following meaning: if $c_j \mapsto_{a_i} \{c_{i_1}, c_{i_2}, \ldots, c_{i_k}\}$, it must then hold that if a_i wants to attend course c_j then she has to attend at least one of the courses $c_{i_1}, c_{i_2}, \ldots, c_{i_k}$ too. We thus define a bundle of courses $C' \subseteq C$ to be *feasible* for a given applicant $a_i \in A$ if Conditions (i) and (ii) in Definition 1 are satisfied, and moreover, for any course $c_j \in C'$, if $c_j \mapsto_{a_i} \{c_{i_1}, c_{i_2}, \ldots, c_{i_k}\}$, then $c_{i_r} \in C'$ for some r $(1 \leq r \leq k)$.

An instance I of the *Course Allocation problem with Alternative PRequisites* (CAAPR) comprises a tuple $I = (A, C, \mathbf{q}, \mathcal{P}, \mapsto)$, where $(A, C, \mathbf{q}, \mathcal{P})$ is an instance of CA and $\mapsto$ is the n_1-tuple of mappings $\mapsto_{a_i}$ for each applicant $a_i \in A$. In an instance of CAAPR, a *matching* M is as defined in the CA case, together with the additional property that, for each applicant $a_i \in A$, $M(a_i)$ is feasible for a_i.

2.4 Corequisites

In the third and final model we assume that constraints on courses are given in the form of corequisites. We assume that corequisite constraints are not applicant-specific, and hence there is a single reflexive, symmetric and transitive relation $\leftrightarrow$ on C such that each applicant is allowed to subscribe to a course $c_j \in C$ only if she also subscribes to each $c_k \in C$ such that $c_j \leftrightarrow c_k$. Two courses $c_j, c_k \in C$ with $c_j \leftrightarrow c_k$ are said to be each other's *corequisites*. Relation $\leftrightarrow$ is an equivalence relation on C and it effectively partitions the set of courses into equivalence classes $C^1, C^2, \ldots, C^r$. Hence an applicant can subscribe either to all courses in one equivalence class or to none.

Formally, we define a bundle of courses $C' \subseteq C$ to be *feasible* for a given applicant $a_i \in A$ if Conditions (i) and (ii) in Definition 1 are satisfied, and moreover, for any two courses $c_j, c_k \in C$, if $c_j \leftrightarrow c_k$ then $c_j \in C'$ if and only if $c_k \in C'$. An instance I of the *Course Allocation problem with Corequisites* (CACR) comprises a tuple $I = (A, C, \mathbf{q}, \mathcal{P}, \leftrightarrow)$, where $(A, C, \mathbf{q}, \mathcal{P})$ is an instance of CA and $\leftrightarrow$ is as defined above. In an instance of CACR, a *matching* M is as defined in the CA case, together with the additional property that, for each applicant $a_i \in A$, $M(a_i)$ is feasible for a_i.

We remark that we do not consider corequisites as a special case of compulsory prerequisites, nor vice versa, for in the definition compulsory prerequisites, we stipulate that the order relation is antisymmetric, while for corequisites symmetry is required.

3 Compulsory prerequisites

In the presence of compulsory prerequisites, we consider the case of additive preferences in Section 3.1 and lexicographic preferences in Section 3.2. It turns out that the problem of finding a POM under additive preferences is NP-hard, as we show in Section 3.1. Thus the majority of our attention is focused on the case of lexicographic preferences in Section 3.2. In that section we mainly consider algorithmic questions associated with the problems of (i) finding a POM, (ii) testing a matching for Pareto optimality, and (iii) finding a POM of maximum size.

3.1 Additive preferences

Our first result shows that the assumption of additive preferences in CAPR makes it difficult to compute a POM.

Lemma 2. *The problem of finding a most-preferred feasible bundle of courses of a given applicant with additive preferences in* CAPR *is NP-hard.*

In the case of just one applicant a_1, a matching M is a POM if and only if a_1 is assigned in M her most-preferred feasible bundle of courses, otherwise M is dominated by assigning a_1 to this bundle. Hence the above lemma directly implies the following result.

Theorem 3. *Given an instance of* CAPR *with additive preferences, the problem of finding a POM is NP-hard.*

Given the above negative result, we do not pursue additive preferences any further in this section, and instead turn our attention to lexicographic preferences.

3.2 Lexicographic preferences

3.2.1 Finding a POM

In this section we explore variants of the sequential allocation mechanism, and show that one formulation allows us to find a POM in polynomial time. This mechanism, referred to as SM-CAPR, does however have some drawbacks: it is not truthful (see Section 6) and it is not able to compute all POMs in general.

In the context of course allocation when there are some dependencies among courses (for instance the constraints on prerequisites in CAPR) the standard sequential mechanism might output an allocation that does not fulfil some constraints on prerequisites. On the other hand, if we require that an applicant can only choose a course if she is already assigned all its prerequisites, the output may be a matching that is not Pareto optimal. This is illustrated by Example 14 in Appendix B.

Therefore we propose a variant of the sequential allocation mechanism, denoted by SM-CAPR, that can be regarded as being "in between" the serial dictatorship mechanism and the general sequential allocation mechanism. Suppose a policy σ is fixed; again one applicant can appear in σ several times, up to her capacity. Applicant a_i on her turn identifies her most-preferred course c_j that she has not yet considered, and that she is not already assigned to. If all courses c_k in $\overrightarrow{c_j}^{a_i}$ satisfy the property that either (i) c_k has a free place or (ii) c_k is already assigned to a_i, and the number of courses in $\overrightarrow{c_j}^{a_i}$ not already assigned to a_i does not exceed the remaining capacity of a_i, then a_i is assigned the bundle $\overrightarrow{c_j}^{a_i}$. If it is impossible to assign bundle $\overrightarrow{c_j}^{a_i}$ to a_i then a_i moves to the next course in her preference list until either she is assigned to some bundle or her preference list is exhausted. This completes a single turn for a_i.

For this approach we need an algorithm that decides whether applicant a_i can be assigned $\overrightarrow{c_j}^{a_i}$. A subsidiary method called Explore, explicitly given in Algorithm 1, takes as input a course c_j where $c_j \notin M(a_i)$ and $|M(c_j)| < q(c_j)$. Here it is assumed that the matching M and the CAPR instance I are global variables. We use a set S, maintained as a global variable, to collect together the courses in $\overrightarrow{c_j}^{a_i}$ that are not already assigned to a_i, assuming the whole bundle can be assigned to a_i. An array of Booleans called 'visited' is maintained as a global variable and used to prevent a course in $\overrightarrow{c_j}^{a_i}$ being visited more than once if there are several paths to it from predecessors. A further global variable, a Boolean 'feasible', initially `true`, remains `true` if each course in $\overrightarrow{c_j}^{a_i}$ either has room for a_i or is already assigned to a_i, and becomes `false` otherwise. Explore is a recursive algorithm that performs a modified depth-first search in the directed graph corresponding to $\overrightarrow{c_j}^{a_i}$.

The pseudocode for SM-CAPR is given in Algorithm 2. This algorithm constructs a POM M in a given CAPR instance I relative to a given policy σ, with the aid of Explore.

Algorithm 1 Explore

Require: course c_j where $c_j \notin M(a_i)$ and $|M(c_j)| < u_j$ (CAPR instance I, matching M, applicant a_i, set S and Boolean feasible are global variables)

Ensure: if c_j and each of its prerequisites either (i) has room for a_i or (ii) is already assigned a_i then all courses not already assigned to a_i are added to S and feasible is unchanged, otherwise feasible is set to `false`

$S := S \cup \{c_j\}$;
if c_j is not a leaf node in $\overset{\rightarrow_{a_i}}{c_j}$ **then**
    for each $c_k \in \overset{\rightarrow_{a_i}}{c_j} \setminus \{c_j\}$ **do**
        if $c_k \notin M(a_i)$ **and not** visited(c_k) **then**
            if $|M(c_k)| < q(c_k)$ **then**
                visited(c_k) := `true`;
                Explore(c_k);
            else
                feasible := `false`;

It uses a Boolean 'isAssigned' to determine whether an applicant a_i has been assigned to some bundle of courses on her turn yet. When attempting to assign a_i to the bundle $\overset{\rightarrow_{a_i}}{c_j}$, it initialises the set S and the Boolean 'feasible' as described above in the context of Explore. It also sets the 'visited' Booleans for each course in $\overset{\rightarrow_{a_i}}{c_j}$ in preparation for the depth-first search carried out by Explore. Once Explore returns, we can assign a_i to the bundle $\overset{\rightarrow_{a_i}}{c_j}$ as long as 'feasible' is `true` and a_i has room for the courses in $\overset{\rightarrow_{a_i}}{c_j}$ that she is not already assigned to; if so, a_i is assigned to all such courses and 'isAssigned' is given the value `true`.

Notice that serial dictatorship will be obtained as a special case of SM-CAPR if all the copies of one applicant form a substring (i.e., a contiguous subsequence) of the policy.

To derive the complexity bound of the algorithm, let us first observe that the complexity of Explore(c_j) is $O(D_{a_i})$ when executed relative to applicant a_i, where $D_{a_i} = \left|\overset{\rightarrow_{a_i}}{c_j}\right|$. This in turn is $O(R_{a_i})$ where R_{a_i} is the number of courses in $\rightarrow_{a_i}$, which is obviously $O(n_2)$ (recall that $|C| = n_2$). Thus the complexity of SM-CAPR is $O(Ln_2)$, where L is the total length of the applicants' preference lists.

Theorem 4. *Algorithm* SM-CAPR *produces a POM for a given instance I of* CAPR *and for a given policy σ in I. The complexity of the algorithm is $O(Ln_2)$, where L is the total length of the applicants' preference lists and n_2 is the number of courses.*

The complexity of the algorithm can be no better than $O(Ln^2)$ in the worst case, as Example 15 in Appendix B shows. Example 16 in Appendix B indicates that in general, SM-CAPR is not capable of generating all POMs for a given CAPR instance. Theorem 3 shows that finding a POM in the presence of additive preferences is NP-hard. Example 17 in Appendix B shows where SM-CAPR can fail to find a POM in this context.

3.2.2 Testing for Pareto optimality

In the previous subsection we gave a polynomial-time algorithm for constructing a POM in an instance of CAPR. It is also reasonable to expect that an alternative approach could involve starting with an arbitrary matching, and for as long as the current matching M is dominated, replace M by any matching that dominates it. However, the difficulty with this method is that the problem of determining whether a matching is Pareto optimal is computationally hard, as we demonstrate by our next result. This hardness result also shows that there is unlikely to be a "nice" (polynomial-time checkable) characterisation of

Algorithm 2 SM-CAPR

Require: CAPR instance I and a policy σ
Ensure: return M, a POM in I

```
M := ∅;
for each applicant a_i ∈ σ in turn do
    isAssigned := false;
    while a_i has not exhausted her preference list and not isAssigned do
        c_j := next course in a_i's list;
        feasible := true;
        if c_j ∉ M(a_i) and |M(c_j)| < q(c_j) then
            S := ∅;
            for each c_k ∈ →c_j^{a_i} do
                visited(c_k) := false;
            visited(c_j) := true;
            Explore(c_j);
            if feasible and |M(a_i)| + |S| ≤ q(a_i) then
                for each c_k ∈ S do
                    M := M ∪ {(a_i, c_k)};
                isAssigned := true;
return M;
```

a POM, in contrast to the case where there are no prerequisites [12]. We firstly define the following problem:

Name: DOM CAPR
Instance: an instance I of CAPR and a matching M in I
Question: is there a matching M' that dominates M in I?

Theorem 5. DOM CAPR *is NP-complete even if each course has capacity 1 and has at most one immediate prerequisite for each applicant.*

We remark that the variant of DOM CAPR for additive preferences is also NP-complete by Theorem 5, since lexicographic preferences can be viewed as a special case of additive preferences by creating utilities that steeply decrease in line with applicants' preferences – see [8] for more details.

3.2.3 Finding large POMs

Example 16 shows that an instance of CAPR may admit POMs of different cardinalities, where the *cardinality* of a POM refers to the number of occupied course slots. It is known that finding a POM with minimum cardinality is an NP-hard problem even for HA, the House Allocation problem (i.e., the restriction of CA in which each applicant and each course has capacity 1) [2, Theorem 2]. However, by contrast to the case for HA [2, Theorem 1] and CA [12, Theorem 6]), the problem of finding a maximum cardinality POM in the CAPR context is NP-hard, as we demonstrate next via two different proofs. Our first proof of this result shows that hardness holds even if the matching is not required to be Pareto optimal.

We firstly define some problems. Let MAX CAPR and MAX POM CAPR denote the problems of finding a maximum cardinality matching and a maximum cardinality POM respectively, given an instance of CAPR. Let MAX CAPR-D denote the decision version of MAX CAPR: given an instance I of CAPR and an integer K, decide whether I admits a matching of cardinality at least K. We obtain MAX POM CAPR-D from MAX POM CAPR similarly.

Theorem 6. MAX CAPR-D *is NP-complete, even if each applicant has capacity 4 and each course has capacity 1, and the prerequisites are the same for all applicants. Moreover* MAX POM CAPR-D *is NP-hard for the same restrictions.*

In the case when a matching is required to be Pareto optimal, NP-hardness holds for stronger conditions on prerequisites than those assumed in Theorem 6.

Theorem 7. MAX POM CAPR-D *is NP-hard, even if each course has at most one prerequisite, and the prerequisites are the same for all applicants.*

Example 18 in Appendix B shows that the difference in cardinalities between POMs may be arbitrary, and that SM-CAPR is not in general a constant-factor approximation algorithm for MAX POM CAPR.

4 Alternative prerequisites

In this section we turn our attention to CAAPR, the analogue of CAPR in which prerequisites need not be compulsory but may be presented as alternatives. We will show that, in contrast to the case for CAPR, finding a POM is NP-hard, under either additive or lexicographic preferences.

Recall that as CAPR is a special case of CAAPR, Lemma 2 implies that finding a most-preferred bundle of courses under additive preferences is NP-hard. Now we prove the same result for lexicographic preferences.

Lemma 8. *Given an instance I of* CAAPR *and an applicant a_i in I, the problem of deciding whether I admits a matching in which a_i receives her most-preferred course is NP-complete.*

Corollary 9. *The problem of finding the most-preferred feasible bundle of courses of a given applicant with lexicographic preferences in* CAAPR *is NP-hard.*

As noted in Section 3.1, in the case of just one applicant a_1, a matching M is a POM if and only if a_1 is assigned in M her most-preferred feasible bundle of courses. Hence Lemma 2 and Corollary 9 directly imply the following assertion.

Theorem 10. *Given an instance of* CAAPR *the problem of finding a POM is NP-hard. The result holds under either additive or lexicographic preferences.*

We remark that, since CAPR is a special case of CAAPR, Theorem 5 implies that the problem of deciding whether a given matching M is a POM is co-NP-complete for lexicographic preferences (and also for additive preferences by the remark following Theorem 5).

5 Corequisites

In this section we focus on CACR, the extension of CA involving corequisite courses. As in the case of CAPR, we will show that finding a POM in the presence of additive preferences is NP-hard. Thus the majority of our attention is concerned with lexicographic preferences. In this case we show how to modify the sequential mechanism in order to obtain a polynomial-time algorithm for finding a POM in the CACR case. Moreover we show that in CACR, the problem of finding a maximum cardinality POM is very difficult to approximate.

We begin with additive preferences. A simple modification of the proof of Lemma 2 (ensuring that, for each i $(1 \leq i \leq n)$, $c_i \leftrightarrow d_i^r$ for each r $(1 \leq r \leq w_i - 1)$) gives the following analogue of Theorem 3.

Theorem 11. *Given an instance of* CACR *with additive preferences, the problem of finding a POM is NP-hard.*

In view of Theorem 11, in the remainder of this section we assume that preferences are lexicographic. In this case we can find a POM efficiently by dealing with the corequisites as follows. Let I be an instance of CACR, let σ be a policy in I, and assume the notation defined in Section 2.4. We lose no generality by supposing that each applicant either finds all the courses in one equivalence class C^k acceptable, or none of them. Replace all the courses in C^k by a single *supercourse* c^k, whose capacity is equal to the sum of the capacities of the courses in C^k. For any applicant $a_i \in A$ who finds all courses in C^k acceptable, remove all such courses from a_i's list and replace them by c^k; since preferences are lexicographic, the position of c^k in the modified preference list of a_i is the position of the most-preferred course of C^k in her original list. Let I' denote the CACR instance obtained from I by using this transformation.

The sequential mechanism for CACR can be executed on I' as follows. The mechanism works according to a given policy σ in stages. In one stage, the applicant a_i who next has her turn according to σ, chooses her most-preferred supercourse c^k to which she has not yet applied. Applicant a_i is assigned to c^k if two conditions are fulfilled: (i) the number of courses assigned to a_i so far plus the cardinality of C^k does not exceed $q(a_i)$, and (ii) each course $c_j \in C^k$ still has a free slot. If this is not possible, at the same stage a_i applies to her next supercourse until she is either assigned some supercourse or her preference list is exhausted. Once the whole process terminates, let M' be the assignment of applicants to supercourses in I' and construct the following assignment M in I from M':

$$M = \{(a_i, c_j) : a_i \in A \wedge c_j \in C^k \wedge (a_i, c^k) \in M'\}.$$

Let SM-CACR denote the mechanism that constructs M from I and σ. With a suitable choice of data structures this algorithm can be implemented to run in $O(L+n_2)$ time, where L is the total length of the applicants' preference lists and n_2 is the number of courses.

Theorem 12. *Algorithm* SM-CACR *produces a POM for a given instance I of* CACR *and for a given policy σ in I. The complexity of the algorithm is $O(L+n_2)$, where L is the total length of the applicants' preference lists and n_2 is the number of courses.*

In the CACR model as defined in Section 2.4, corequisite constraints are common to all applicants. In this setting, and after the modification described prior to Theorem 12, in which courses are merged into supercourses, CACR becomes equivalent to CAP, the extension of CA with price-budget constraints described in [12]. For an instance I of CAP, it is known that for each POM M in I, there exists a policy σ such that executing SM-CACR relative to σ produces M [12, Theorem 3]. Example 19 in Appendix B presents an observation about the behaviour of SM-CACR if we extend it to the variant of CACR in which corequisite constraints are specific to individual applicants.

Given an instance of CAP, the problem of finding a maximum cardinality POM is NP-hard [12, Theorem 7]. Using the connection between CACR and CAP described in the previous paragraph, the same is therefore true for MAX POM CACR, the problem of finding a maximum cardinality POM, given an instance of CACR. We now strengthen this result by showing that MAX POM CACR is very difficult to approximate.

Theorem 13. MAX POM CACR *is NP-hard and not approximable within a factor of $N^{1-\varepsilon}$, for any $\varepsilon > 0$, unless P=NP, where N is the total capacity of the applicants.*

6 Open problems and directions for future research

We would like to conclude with several open problems and directions for future research.

1. **Refining the boundary between efficiently solvable and hard problems.** In the proofs of the NP-hardness and inapproximability results in this paper we had some applicants whose preference lists were not complete and/or whose capacity was not bounded by a constant. Will the hardness results still be valid if there are no unacceptable courses and all capacities are bounded? These problems also call for a more detailed multivariate complexity analysis. It might be interesting to determine whether restricting some other parameters, e.g., the lengths of preference lists, may make the problems tractable.
2. **Indifferences in the preference lists.** In this paper, we assumed that all the preferences are strict. If preference lists contain ties, sequential mechanisms have to be carefully modified to ensure Pareto optimality. Polynomial-time algorithms for finding a Pareto optimal matching in the presence of ties have been given in the context of HAT (the extension of HA where preference lists may include ties) by Krysta et al. [25] and in its many-to-many generalisation CAT (the extension of CA where preference lists may include ties) by Cechlárová et al. [11]. However as far as we are aware, it remains open to extend these algorithms to the cases of CAPR and CACR where preference lists may include ties.
3. **Strategic issues.** By a standard argument, one can ensure that the sequential mechanism that lets each applicant on her turn choose her entire most-preferred bundle of courses (i.e., the serial dictatorship mechanism) is strategy-proof even in the case of prerequisites. However, serial dictatorship may be very unfair, as the first dictator may grab all the courses and leave nothing for the rest of the applicants. Let us draw the reader's attention to several economic papers that highlight the special position of serial dictatorship among the mechanisms for allocation of multiple indivisible goods: serial dictatorship is the only allocation rule that is Pareto efficient, strategy-proof and fulfils some additional properties, namely non-bossiness and citizen sovereignty [30], and population monotonicity or consistency [23]. We were not able to obtain a similar characterization of serial dictatorship for CAPR.

   As far as the general sequential mechanism is concerned, a recent result by Hosseini and Larson [20] shows that no sequential mechanism that allows interleaving policies (i.e., in which an a_i is allowed to pick courses more than once, and between two turns of a_i another applicant has the right to pick a course) is strategy-proof, even in the simpler case without any prerequisites. It immediately follows that SM-CAPR is not strategy-proof. However, it is not known whether a successful manipulation can be computed efficiently. Further, we have shown that in CAPR, not all POMs can be obtained by a sequential mechanism. We leave it open as to whether a strategy-proof and Pareto optimal mechanism other than serial dictatorship exists in CAPR.

Acknowledgements

This work was supported by VEGA grants 1/0344/14 and 1/0142/15 from the Slovak Scientific Grant Agency (Cechlárová), by the Swiss National Science Foundation SNFS (Klaus), by grant EP/K010042/1 from the Engineering and Physical Sciences Research Council (Manlove), and by a Short-Term Scientific Mission from COST Action IC1205 on Computational Social Choice (Manlove). Part of this work was carried out whilst the authors visited Corvinus University of Budapest, and whilst the third author visited P.J. Šafárik University. This research was initiated during Dagstuhl Seminar 15241 on Computational Social Choice: Theory and Applications [7]. We would like to thank the following Dagstuhl participants who were involved in initial discussions regarding the results in this paper: Haris Aziz, Péter Biró, Jiehua Chen and Nicholas Mattei.

References

[1] A. Abdulkadiroğlu and T. Sönmez. Random serial dictatorship and the core from random endowments in house allocation problems. *Econometrica*, 66(3):689–701, 1998.

[2] D.J. Abraham, K. Cechlárová, D.F. Manlove, and K. Mehlhorn. Pareto optimality in house allocation problems. In *Proceedings of ISAAC '04: the 15th Annual International Symposium on Algorithms and Computation*, volume 3341 of *Lecture Notes in Computer Science*, pages 3–15. Springer, 2004.

[3] H. Aziz, T. Walsh, and L. Xia. Possible and necessary allocations via sequential mechanisms. In *Proceedings of IJCAI 2015: the 24th International Joint Conference on Artificial Intelligence*, pages 468–474. AAAI Press, 2015.

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

[5] S. Barberà, W. Bossert, and P.K. Pattanaik. Ranking sets of objects. In S. Barberà, P.J. Hammond, and C. Seidl, editors, *Handbook of Utility Theory*, volume 2, chapter 17, pages 893–977. Kluwer Academic Publishers, 2004.

[6] P. Biró and E. McDermid. Matching with sizes (or scheduling with processing set restrictions). *Discrete Applied Mathematics*, 164(1):61–67, 2014.

[7] C. Boutilier, B. Dorn, N. Maudet, and V. Merlin. Computational Social Choice: Theory and Applications (Dagstuhl Seminar 15241). *Dagstuhl Reports*, 5(6):1–27, 2016.

[8] S. Bouveret and J. Lang. Elicitation-free protocol for allocating indivisible goods. In *Proceedings of IJCAI 2011: the 22nd International Joint Conference on Artificial Intelligence*, pages 73–78. AAAI Press, 2011.

[9] S.J. Brams and D.L. King. Efficient fair division: Help the worst off or avoid envy? *Rationality and Society*, 17(4):387–421, 2005.

[10] E. Budish and E. Cantillon. The multi-unit assignment problem: Theory and evidence from course allocation at Harvard. *American Economic Review*, 102(5):2237–2271, 2012.

[11] K. Cechlárová, P. Eirinakis, T. Fleiner, D. Magos, D.F. Manlove, I. Mourtos, E. Ocel'áková, and B. Rastegari. Pareto optimal matchings in many-to-many markets with ties. In *Proceedings of SAGT 2015: the 8th International Symposium on Algorithmic Game Theory*, pages 27–39. Springer, 2015.

[12] K. Cechlárová, P. Eirinakis, T. Fleiner, D. Magos, I. Mourtos, and E. Potpinková. Pareto optimality in many-to-many matching problems. *Discrete Optimization*, 14:160–169, 2014.

[13] B.A. Davey and H.A. Priestly. *Introduction to lattices and order*. Cambridge University Press, 1990.

[14] F. Diebold, H. Aziz, M. Bichler, F. Matthes, and A. Schneider. Course allocation via stable matching. *Business and Information Systems Engineering*, 6(2):97–110, 2014.

[15] T. Dodson, N. Mattei, J.T. Guerin, and J. Goldsmith. An English-language argumentation interface for explanation generation with Markov decision processes in the domain of academic advising. *ACM Transactions on Interactive Intelligent Systems*, 3(3):18, 2013.

[16] P.C. Fishburn. Axioms for lexicographic preferences. *Review of Economic Studies*, 42(3):415–419, 1975.

[17] M.R. Garey and D.S. Johnson. *Computers and Intractability.* Freeman, San Francisco, CA., 1979.

[18] M.R. Garey, D.S. Johnson, and L. Stockmeyer. Some simplified NP-complete graph problems. *Theoretical Computer Science*, 1:237–267, 1976.

[19] J.T. Guerin, J. Hanna, L. Ferland, N. Mattei, and J. Goldsmith. The academic advising planning domain. In *Proceedings of WS-IPC-12: the 3rd Workshop on the International Planning Competition*, pages 1–5, 2012.

[20] H. Hosseini and K. Larson. Strategyproof quota mechanisms for multiple assignment problems. Technical Report 1507.07064, Computing Research Repository, Cornell University Library, 2015. Available from `http://arxiv.org/abs/1507.07064`.

[21] R.M. Karp. Reducibility among combinatorial problems. In R.E. Miller and J.W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.

[22] A.S. Kelso, Jr. and V.P. Crawford. Job matching, coalition formation and gross substitutes. *Econometrica*, 50:1483–1504, 1982.

[23] B. Klaus and E. Miyagawa. Strategy-proofness, solidarity, and consistency for multiple assignment problems. *International Journal of Game Theory*, 30:421–435, 2001.

[24] D.A. Kohler and R. Chandrasekaran. A class of sequential games. *Operations Research*, 19(2):270–277, 1971.

[25] P. Krysta, D.F. Manlove, B. Rastegari, and J. Zhang. Size versus truthfulness in the House Allocation problem. In *Proceedings of EC 2014: the 15th ACM Conference on Economics and Computation*, pages 453–470. ACM, 2014.

[26] J. K. Lenstra and A.H.G. Rinnooy Kan. Complexity of scheduling under precedence constraints. *Operations Research*, 26(1):22–35, 1978.

[27] D. Maier and J.A. Storer. A note on the complexity of the superstring problem. Technical Report 233, Princeton University, Department of Electrical Engineering and Computer Science, Princeton, NJ, October 1977.

[28] M. Manea. Serial dictatorship and Pareto optimality. *Games and Economic Behavior*, 61:316–330, 2007.

[29] D. Monte and N. Tumennasan. Matching with quorums. *Economics Letters*, 120:14–17, 2013.

[30] S. Pápai. Strategyproof and nonbossy multiple assignments. *Journal of Public Economic Theory*, 3(3):257–271, 2001.

[31] A.E. Roth. The college admissions problem is not equivalent to the marriage problem. *Journal of Economic Theory*, 36:277–288, 1985.

[32] D. Saban and J. Sethuraman. A note on object allocation under lexicographic preferences. *Journal of Mathematical Economics*, 50:283–289, 2014.

[33] L.J. Schulman and V.V. Vazirani. Allocation of divisible goods under lexicographic preferences. Technical Report 1206.4366, Computing Research Repository, Cornell University Library, 2012. Available from `http://arxiv.org/abs/1206.4366`.

[34] L.-G. Svensson. Queue allocation of indivisible goods. *Social Choice and Welfare*, 11(4):323–330, 1994.

[35] T. Todo, H. Sun, and M. Yokoo. Strategyproof exchange with multiple private endowments. In *Proceedings of AAAI 2014: the 28th AAAI Conference on Artificial Intelligence*, pages 805–811, 2014.

A Appendix: Proofs

Proof of Lemma 2. We transform from the KNAPSACK problem, which is defined as follows. An instance I of KNAPSACK comprises a set of integers $w_1, w_2, \ldots, w_n, p_1, p_2, \ldots, p_n, W, P$. The problem is to decide whether there exists a set $K \subseteq \{1, 2 \ldots, n\}$ such that $\sum_{i \in K} w_i \leq W$ and $\sum_{i \in K} p_i \geq P$. KNAPSACK is NP-complete [21].

Let us construct an instance J of CAPR as follows. There is a single applicant a_1 such that $q(a_1) = W$. The set of courses is $C \cup D$, where $C = \{c_1, c_2, \ldots, c_n\}$ and $D = \cup_{i=1}^{n} \{d_i^1, d_i^2, \ldots, d_i^{w_i - 1}\}$. For each course c_i ($1 \leq i \leq n$), its utility for applicant a_1 is equal to p_i. Moreover c_i has $w_i - 1$ prerequisites $d_i^1, d_i^2, \ldots, d_i^{w_i - 1}$. For each i ($1 \leq i \leq n$) and j ($1 \leq j \leq w_i - 1$), d_i^j has utility ε_i^j for a_1, where the ε values are all positive, distinct and add up to less than 1. They are selected simply to ensure that a_1's preferences over individual courses are strict. It is easy to see that in J, a_1 has a feasible bundle with utility at least P if and only if I is a yes-instance of KNAPSACK. □

Proof of Theorem 4. It is straightforward to verify that the assignment M produced by SM-CAPR is a matching in I. Suppose for a contradiction that M is not a POM. Then there exists a matching M' that dominates M. Let A' be the set of applicants who prefer their assignment in M' to their assignment in M. For each $a_j \in A'$, denote by s_j the first stage of SM-CAPR when a_j did not get a course, say c_{i_j}, that is assigned to her in M'. Stage s_j must have initiated a call to $\mathsf{Explore}(c_{i_j})$ with applicant a_j, for c_{i_j} cannot be a prerequisite for any course assigned to a_j before stage s_j. Clearly c_{i_j} was not assigned to a_j in M, and c_{i_j} is the most-preferred course in $M'(a_j) \backslash M(a_j)$. Let $a_k = \arg\min_{a_j \in A'} \{s_j\}$.

As $c_{i_k} \in M'(a_k)$, all the prerequisites of c_{i_k} also belong to $M'(a_k)$. Since s_k is the first stage in which an applicant received a course in M' but not in M, all the courses assigned in M to any applicant a_j in previous stages also belong to $M'(a_j)$, for otherwise M' does not dominate M. Thus it was not the case that applicant a_k failed to receive course c_{i_k} in M at stage s_k because a_k did not have room for c_{i_k} and all of its prerequisites not already assigned to her in M. Rather, applicant a_k failed to receive course c_{i_k} in M at stage s_k because at least one of the prerequisite courses of c_{i_k}, say c_r, was already full in M before stage s_k. It follows from our previous remark that in M', all the places in c_r are occupied by applicants other than a_k. Thus c_{i_k} cannot be assigned to a_k in M' after all, a contradiction. □

Proof of Theorem 5. Clearly DOM CAPR is in NP. To show NP-hardness, we reduce from EXACT 3-COVER, which is NP-complete [21].

Name: EXACT 3-COVER
Instance: a set $X = \{x_1, x_2, \ldots, x_{3n}\}$ and a set $\mathcal{T} = \{T_1, T_2, \ldots, T_m\}$ such that for each i ($1 \leq i \leq m$), $T_i \subseteq X$ and $|T_i| = 3$.
Question: is there a subset $\mathcal{T}'$ of $\mathcal{T}$ such that $T_i \cap T_j = \emptyset$ for each $T_i, T_j \in \mathcal{T}'$ and $\cup_{T_i \in \mathcal{T}'} T_i = X$?

Let I be an instance of EXACT 3-COVER as defined above. For each $T_i \in \mathcal{T}$, let us denote the elements that belong to T_i by $x_{i_1}, x_{i_2}, x_{i_3}$. Obviously, we lose no generality by assuming that $m \geq n$.

We construct an instance J of DOM CAPR based on I in the following way. The set of applicants is $A = \{a_1, a_2, \dots, a_{m+1}\}$. The capacities are $q(a_i) = 4$ $(1 \leq i \leq m)$ and $q(a_{m+1}) = 2n+m$. The set of courses is $C = \mathcal{T} \cup X \cup Y \cup W$, where $\mathcal{T} = \{T_1, T_2, \dots, T_m\}$, $X = \{x_1, x_2, \dots, x_{3n}\}$, $Y = \{y_1, y_2, \dots, y_m\}$ and $W = \{w_1, w_2, \dots, w_{m-n}\}$. (Some of the courses in J derived from the elements and sets in I are denoted by identical symbols, but no ambiguity should arise.) The preferences of the applicants are:

$$\begin{array}{rl} P(a_i): & T_i, [W], y_i, x_{i_1}, x_{i_2}, x_{i_3} \quad (1 \leq i \leq m) \\ P(a_{m+1}): & y_1, [X], [W], y_2, \dots, y_m \end{array}$$

where the symbols $[W]$ and $[X]$ denote all the courses in W and X, respectively, in an arbitrary strict order. Recall that $\{x_{i_1}, x_{i_2}, x_{i_3}\} \subseteq X$ $(1 \leq i \leq m)$. The prerequisites of applicants are:

$$\begin{array}{rll} a_i: & T_i \rightarrow_{a_i} x_{i_1} \rightarrow_{a_i} x_{i_2} \rightarrow_{a_i} x_{i_3} & (1 \leq i \leq m) \\ a_{m+1}: & y_1 \rightarrow_{a_{m+1}} y_2 \rightarrow_{a_{m+1}} \cdots \rightarrow_{a_{m+1}} y_m & \end{array}$$

Define the following matching:

$$M = \{(a_i, y_i) : 1 \leq i \leq m\} \cup \{(a_{m+1}, x_j) : 1 \leq j \leq 3n)\} \cup \{(a_{m+1}, w_k) : 1 \leq k \leq m-n)\}.$$

We claim that I admits an exact cover if and only if M is dominated in J.

For, suppose that $\{T_{j_1}, T_{j_2}, \dots, T_{j_n}\}$ is an exact cover in I. We construct a matching M' in J as follows. For each k $(1 \leq k \leq n)$, in M', assign a_{j_k} to T_{j_k} and to a_{j_k}'s three prerequisites of T_{j_k} that belong to X. Let $A' = \{a_{j_1}, a_{j_2}, \dots, a_{j_n}, a_{m+1}\}$ and let $A \backslash A' = \{a_{k_1}, a_{k_2}, \dots, a_{k_{m-n}}\}$. For each r $(1 \leq r \leq m-n)$, in M', assign a_{k_r} to w_r. Finally in M', assign a_{m+1} to every course in Y. It is straightforward to verify that M' dominates M in J.

Conversely, suppose there exists a matching M' that dominates M in J. Then at least one applicant must be better off in M' compared to M.

If a_{m+1} improves, she must obtain y_1 and so, due to her prerequisites, all the courses in Y. This means that each applicant a_i $(1 \leq i \leq m)$ must obtain a course that she prefers to y_i.

Each such applicant a_i can improve relative to M in two ways. Either she obtains in M' a course in W, or she obtains T_i. In the latter case then she must receive the corresponding courses $x_{i_1}, x_{i_2}, x_{i_3}$ in M'. In either of these two cases, since a_{m+1} cannot be worse off, she must obtain in M' the course y_1 and hence all courses in Y.

This means that in M' all the applicants must strictly improve compared to M. As there are only $n-m$ courses in W, there are exactly n applicant in $A \backslash \{a_{m+1}\}$ – let these applicants be $\{a_{j_1}, a_{j_2}, \dots, a_{j_n}\}$ – who obtain a course in $\mathcal{T}$ and its three prerequisites in X. As the capacity of each course is 1, it follows that $\{T_{j_k} : 1 \leq j \leq n\}$ is an exact cover in I. ☐

Proof of Theorem 6. Clearly MAX CAPR-D is in NP. To show NP-hardness, we reduce from IND SET-D in cubic graphs; here IND SET-D is the decision version of IND SET, the problem of finding a maximum independent set in a given graph. IND SET-D is NP-complete in cubic graphs [18, 27]. Let $\langle G, K \rangle$ be an instance of IND SET-D in cubic graphs, where $G = (V, E)$ is a cubic graph and K is a positive integer. Assume that $V = \{v_1, v_2, \dots, v_n\}$ and $E = \{e_1, e_2, \dots, e_m\}$. For a given vertex $v_i \in V$, let $E_i \subseteq E$ denote the set of edges incident to v_i in G. Clearly $|E_i| = 3$ as G is cubic.

We form an instance I of MAX CAPR-D as follows. Let A be the set of applicants and let $V \cup E$ be the set of courses, where $A = \{a_i : v_i \in V\}$ (we use the same notation for the vertices and edges in G as we do for the courses in I, but no ambiguity should arise.) Let the capacity of each applicant be 4 and the capacity of each course be 1. The preference list of each applicant is as follows:

$$a_i : v_i \;\; [E_i] \qquad (1 \leq i \leq n)$$

where the symbol $[E_i]$ denotes all members of E_i listed in arbitrary order. For each $v_i \in V$ and for each $e_j \in E_i$, define the prerequisite $v_i \to e_j$ for all applicants. We claim that G has an independent set of size at least K if and only if I has a matching of size at least $m+K$.

For, suppose that S is an independent set in G where $|S| \geq K$. Let $A' = \{a_i \in A : v_i \in S\}$. We form an assignment M in I as follows. For each applicant $a_i \in A'$, assign a_i to v_i plus the prerequisite courses in E_i. Then for each applicant $a_i \notin A'$, assign a_i to any remaining courses in E_i (if any). It is straightforward to verify that M is a matching in I. Also $|M| = m + |S| \geq m + K$, since every applicant $a_i \in A'$ obtains v_i and all prerequisite courses in E_i, and then the applicants in $A \backslash A'$ are collectively assigned to all remaining unmatched courses in E.

Conversely suppose that M is a matching in I such that $|M| \geq m + K$. Let S denote the courses in V that are matched in M, and suppose that $|S| < K$. Then since $|E| = m$ and all courses have capacity 1, $M \leq |S| + |E| \leq m + |S| < m + K$, a contradiction. Hence $|S| \geq K$. We now claim that S is an independent set in G. For, suppose that v_i and v_j are two adjacent vertices in G that are both in S. Then it is impossible for both a_i and a_j to meet their prerequisites on v_i and v_j in I, respectively, a contradiction.

To show the NP-hardness of MAX POM CAPR-D, we make the observation that in the above proof of NP-completeness of MAX CAPR-D, the matching M in I constructed from an independent set S in G is in fact Pareto optimal. To see this, let σ be an ordering of the applicants such that every applicant in A' precedes every applicant in $A \backslash A'$. Let M be the result of running Algorithm SM-CAPR relative to the ordering σ. It follows by Theorem 4 that M is a POM in I. The remainder of the proof of NP-completeness of MAX CAPR-D can then be used to show that MAX POM CAPR-D is NP-hard. □

Proof of Theorem 7. We firstly remark that, in view of Theorem 5, it is not known whether MAX POM CAPR-D belongs to NP. We show NP-hardness for this problem via a reduction from (2,2)-E3-SAT, which is defined as follows:

> *Name:* (2,2)-E3-SAT
> *Instance:* a Boolean formula B, where each clause in B has size three, and each variable occurs exactly twice as an unnegated literal and exactly twice as a negated literal in B.
> *Question:* is B satisfiable?

Let B be an instance of (2,2)-E3-SAT, where $V = \{v_1, v_2, \ldots, v_n\}$ is the set of variables and $C = \{c_1, c_2, \ldots, c_m\}$ is the set of clauses. We construct an instance I of MAX POM CAPR-D as follows. Let $X \cup Y$ be the set of courses, where $X = \{x_i^1, x_i^2, \bar{x}_i^1, \bar{x}_i^2 : 1 \leq i \leq n\}$ and $Y = \{y_i^1, y_i^2 : 1 \leq i \leq n\}$. The courses in X correspond to the first and second occurrences of v_i and $\bar{v}_i$ in B for each i $(1 \leq i \leq n)$. Let $A \cup G$ be the set of applicants, where $A = \{a_j : 1 \leq j \leq m\}$ and $G = \{g_i^1, g_i^2 : 1 \leq i \leq n\}$. Each course has capacity 1. Each applicant in A has capacity 1, whilst each applicant in G has capacity 2. For each i $(1 \leq i \leq n)$, define the prerequisite $y_i^1 \to y_i^2$, which is the same for all applicants. For each j $(1 \leq j \leq m)$ and for each s $(1 \leq s \leq 3)$, let $x(c_j^s)$ denote the X-course corresponding to the literal appearing at position s of clause c_j in B. For example if the second position of clause c_5 contains the second occurrence of literal $\bar{v}_i$, then $x(c_5^2) = \bar{x}_i^2$. The preference lists of the applicants are as follows:

$$\begin{array}{lll} P(a_j): & x(c_j^1), x(c_j^2), x(c_j^3) & (1 \le j \le m) \\ P(g_i^1): & y_i^1, y_i^2, x_i^1, x_i^2 & (1 \le i \le n) \\ P(g_i^2): & y_i^1, y_i^2, \bar{x}_i^1, \bar{x}_i^2 & (1 \le i \le n) \end{array}$$

We claim that B has a satisfying truth assignment if and only if I has a POM of size $m+4n$.

For, suppose that f is a satisfying truth assignment for B. We form a matching M in I as follows. For each i $(1 \le i \le n)$, if $f(v_i)$=`true` then add the pairs (g_i^1, y_i^1), (g_i^1, y_i^2), $(g_i^2, \bar{x}_i^1)$, $(g_i^2, \bar{x}_i^2)$ to M. On the other hand if $f(v_i)$=`false` then add the pairs (g_i^1, x_i^1), (g_i^1, x_i^2), (g_i^2, y_i^1), (g_i^2, y_i^2) to M. For each j $(1 \le j \le m)$, at least one literal in c_j is true under f. Let s be the minimum integer such that the literal at position s of c_j is true under f. Course $x(c_j^s)$ is still unmatched by construction; add $(a_j, x(c_j^s))$ to M. It may be verified that M is a POM of size $m + 4n$ in I.

Conversely suppose that M is a POM in I of size $m + 4n$. Then the cardinality of M implies that every applicant is full in M. We firstly show that, for each i $(1 \le i \le n)$, either $\{(g_i^1, y_i^1), (g_i^1, y_i^2)\} \subseteq M$ or $\{(g_i^2, y_i^1), (g_i^2, y_i^2)\} \subseteq M$. Suppose this is not the case. As a consequence of the prerequisites, if $(g_i^r, y_i^1) \in M$ for some i $(1 \le i \le n)$ and $r \in \{1, 2\}$, then $(g_i^r, y_i^2) \in M$. Suppose now that $(g_i^r, y_i^2) \in M$ for some $r \in \{1, 2\}$, but $(g_i^r, y_i^1) \notin M$. Let M' be the matching obtained from M by removing any assignee of g_i^r worse than y_i^2 (if such an assignee exists) and adding (g_i^r, y_i^1) to M. Then M' dominates M, a contradiction. Now suppose that y_i^2 is unmatched in M. Let M' be the matching obtained from M by removing any assignee of g_i^1 worse than y_i^2 (if such an assignee exists) and adding (g_i^1, y_i^r) to M $(r \in \{1, 2\})$. Then M' dominates M, a contradiction. Thus the claim is established. It follows that for each i $(1 \le i \le n)$, either g_i^1 is matched in M to two members of X and g_i^2 is matched in M to two members of Y, or vice versa.

Now create a truth assignment f in B as follows. For each i $(1 \le i \le n)$, if $(g_i^1, y_i^1) \in M$, set $f(v_i)$=`true`, otherwise set $f(v_i)$=`false`. We claim that f is a satisfying truth assignment for B. For, let j $(1 \le j \le m)$ be given. Then $(a_j, x(c_j^s)) \in M$ for some s $(1 \le s \le 3)$. If $x(c_j^s) = x_i^r$ for some i $(1 \le i \le n)$ and r $(r \in \{1, 2\})$ then $f(v_i) = T$ by construction. Similarly if $x(c_j^s) = \bar{x}_i^r$ for some $(1 \le i \le n)$ and r $(r \in \{1, 2\})$ then $f(v_i) = F$ by construction. Hence f satisfies B. $\square$

Proof of Lemma 8. Clearly the problem belongs to NP. To show NP-hardness, we reduce from VC-D, the decision version of VC, which is the problem of finding a vertex cover of minimum size in a given graph. VC-D is NP-complete [17]. Let $\langle G, K \rangle$ be an instance of VC-D, where $G = (V, E)$ is a graph and K is a positive integer. Assume that $V = \{v_1, v_2, \ldots, v_n\}$ and $E = \{e_1, e_2, \ldots, e_m\}$. We construct an instance I of CAAPR as follows. Let the set of courses be $V \cup E \cup \{b\}$ (again, we use the same notation for vertices and edges in G as we do for courses in I, but no ambiguity should arise.) There is a single applicant a_1 with capacity $m + K + 1$ whose preference list is as follows:

$$P(a_1) : b, e_1, e_2, \ldots, e_m, v_1, v_2, \ldots, v_n.$$

Course b has a single compulsory prerequisite course e_1, and each course e_j $(2 \le j \le m-1)$ has a single compulsory prerequisite course e_{j+1}. Moreover, all the E-courses have (alternative) prerequisites; namely, for any j $(1 \le j \le m)$, if course e_j corresponds to the edge $e_j = \{v_i, v_k\}$ then $e_j \mapsto_{a_1} \{v_i, v_k\}$. We claim that G admits a vertex cover of size at most K if and only if I admits a matching in which a_1 is assigned course b.

For, suppose that G admits a vertex cover S where $|S| \le K$. Form a matching M by assigning a_1 to the bundle $B = \{b\} \cup E \cup S$. Then B is a feasible bundle of courses for a_1, and $b \in B$.

Conversely, suppose I admits a matching M in which a_1 is assigned a bundle containing course b. Then, due to the prerequisites, B must contain all E-courses and for each course in $e_j \in E$, B must contain some course in $v_i \in V$ that corresponds to a vertex incident to e_j. Let $S = B \cap V$. Clearly S is a vertex cover in G, and as $q(a) = m + K + 1$, it follows that $|S| \leq K$. $\square$

Proof of Theorem 12. It is straightforward to verify that the assignment M produced by SM-CACR is a matching in I. Suppose for a contradiction that M is not a POM in I. Denote by M' the matching in I' that SM-CACR constructs. Then there exists a matching M'' that dominates M' in I'. Let A' be the set of applicants who prefer, in I', their assignment in M'' to their assignment in M'. For each $a_j \in A'$, denote by s_j the first stage of SM-CACR when a_j did not get a supercourse, say c^{i_j}, that is assigned to her in M''. Clearly c^{i_j} was not assigned to a_j in M', and c^{i_j} is the most-preferred supercourse in $M''(a_j) \backslash M'(a_j)$. Let $a_k = \arg\min_{a_j \in A'}\{s_j\}$.

Clearly $c^{i_k} \in M''(a_k)$. Moreover since s_k is the first stage in which an applicant received a supercourse in M'' but not in M', all the supercourses assigned in M' to any applicant a_j in previous stages also belong to $M''(a_j)$, for otherwise M'' does not dominate M'. Thus it was not the case that applicant a_k failed to receive supercourse c^{i_k} in M' at stage s_k because a_k did not have room for c^{i_k} in M'. Rather, applicant a_k failed to receive course c^{i_k} in M' at stage s_k because at least one of the courses in C^{i_k}, say c_r, was already full in M' before stage s_k. It follows from our previous remark that in M'', all the places in c_r are occupied by applicants other than a_k. Thus c^{i_k} cannot be assigned to a_k in M'' after all, a contradiction. $\square$

Proof of Theorem 13. Let $\varepsilon > 0$ be given. Let B be an instance of (2,2)-E3-SAT (see the proof of Theorem 7), where $V = \{v_1, v_2, \ldots, v_n\}$ is the set of variables and $C = \{c_1, c_2, \ldots, c_m\}$ is the set of clauses. Let $\beta = \lceil \frac{2}{\varepsilon} \rceil$ and let $\alpha = n^\beta$.

We form an instance I of CACR as follows. Let $X \cup Y \cup Z$ be the set of courses, where $X = \{x_i^1, x_i^2, \bar{x}_i^1, \bar{x}_i^2 : 1 \leq i \leq n\}$, $Y = \{y_i^1, y_i^2 : 1 \leq i \leq n\}$ and $Z = \{z_1, z_2, \ldots, z_D\}$, where $D = 6n(\alpha - 1) + 1$. The courses in X correspond to the first and second occurrences of v_i and $\bar{v}_i$ in B for each i $(1 \leq i \leq n)$. Let $A \cup G \cup \{b, h\}$ be the set of applicants, where $A = \{a_j : 1 \leq j \leq m\}$ and $G = \{g_i^1, g_i^2 : 1 \leq i \leq n\}$. Each course has capacity 1. Each applicant in A has capacity 1, each applicant in G has capacity 2, h has capacity $2n - m$ and b has capacity D.

For each i $(1 \leq i \leq n)$, courses y_i^1 and y_i^2 are corequisites. Also all the courses in Z are corequisites. For each j $(1 \leq j \leq m)$ and for each s $(1 \leq s \leq 3)$, $x(c_j^s)$ is as defined in the proof of Theorem 7. The preference lists of the applicants are as follows:

$$
\begin{array}{rll}
P(a_j): & x(c_j^1), x(c_j^2), x(c_j^3) & (1 \leq j \leq m) \\
P(g_i^1): & y_i^1, y_i^2, x_i^1, x_i^2 & (1 \leq i \leq n) \\
P(g_i^2): & y_i^1, y_i^2, \bar{x}_i^1, \bar{x}_i^2 & (1 \leq i \leq n) \\
P(h): & [X] & \\
P(b): & [X], [Z] &
\end{array}
$$

In the preference lists of h and b, the symbols $[X]$ and $[Z]$ denote all members of X and Z listed in arbitrary strict order, respectively. In I the total capacity of the applicants, denoted by N, satisfies $N = D + 6n$. We claim that if B has a satisfying truth assignment then I has a POM of size $D + 6n$, whilst if B does not have a satisfying truth assignment then any POM in I has size at most $6n$.

For, suppose that f is a satisfying truth assignment for B. We form a matching M in I as follows. For each i $(1 \leq i \leq n)$, if $f(v_i)$=`true` then add the pairs (g_i^1, y_i^1), (g_i^1, y_i^2), $(g_i^2, \bar{x}_i^1)$, $(g_i^2, \bar{x}_i^2)$ to M. On the other hand if $f(v_i)$=`false` then add the pairs (g_i^1, x_i^1),

(g_i^1, x_i^2), (g_i^2, y_i^1), (g_i^2, y_i^2) to M. For each j $(1 \le j \le m)$, at least one literal in c_j is true under f. Let s be the minimum integer such that the literal at position s of c_j is true under f. Course $x(c_j^s)$ is still unmatched by construction; add $(a_j, x(c_j^s))$ to M. There remain $2n - m$ courses in X that are as yet unmatched in M; assign all these courses to h. Finally assign all courses in Z to b in M. It may be verified that M is a POM of size $D + 6n$ in I.

Now suppose that f admits no satisfying truth assignment. Let M be any POM in I. We will show that $|M| \le 6n$. Suppose not. Then $|M| > 6n$ and the only way this is possible is if at least one course in Z is matched in M. But only b can be assigned members of Z in M, and since all pairs of courses in Z are corequisites, it follows that $M(b) = Z$.

We next show that, for each i $(1 \le i \le n)$, either $\{(g_i^1, y_i^1), (g_i^1, y_i^2)\} \subseteq M$ or $\{(g_i^2, y_i^1), (g_i^2, y_i^2)\} \subseteq M$. Suppose this is not the case for some i $(1 \le i \le n)$. As a consequence of the corequisite restrictions on courses in Y, y_i^1 and y_i^2 are unmatched in M. Let M' be the matching obtained from M by deleting any assignee of g_i^1 worse than y_i^2 (if such an assignee exists) and by adding (g_i^1, y_i^1) and (g_i^1, y_i^2) to M. Then M' dominates M, a contradiction.

We claim that each course in X is matched in M. For, suppose that some course $x \in X$ is unmatched. Then let M' be the matching obtained from M by unassigning b from all courses in Z, and by assigning b to x. Then M' dominates M, a contradiction.

It follows that every course in $X \cup Y$ is matched in M. Since $|X \cup Y| = 6n$ and the applicants in $A \cup G \cup \{h\}$ have total capacity $6n$, every applicant in $A \cup G \cup \{h\}$ is full.

Create a truth assignment f in B as follows. For each i $(1 \le i \le n)$, if $(g_i^1, y_i^1) \in M$, set $f(v_i)$=`true`, otherwise set $f(v_i)$=`false`. We claim that f is a satisfying truth assignment for B. For, let j $(1 \le j \le m)$ be given. Then $(a_j, x(c_j^s)) \in M$ for some s $(1 \le s \le 3)$. If $x(c_j^s) = x_i^r$ for some i $(1 \le i \le n)$ and r $(r \in \{1,2\})$ then $f(v_i)$=`true` by construction. Similarly if $x(c_j^s) = \bar{x}_i^r$ for some $(1 \le i \le n)$ and r $(r \in \{1,2\})$ then $f(v_i)$=`false` by construction. Hence f satisfies B, a contradiction.

Thus if B is satisfiable then I admits a POM of size $D + 6n = 6n(\alpha - 1) + 1 + 6n > 6n\alpha$. If B is not satisfiable then any POM in I has size at most $6n$. Hence an α-approximation algorithm for MAX POM CACR implies a polynomial-time algorithm to determine whether B is satisfiable, a contradiction to the NP-completeness of (2,2)-E3-SAT.

It remains to show that $N^{1-\varepsilon} \le \alpha$. On the one hand, $N = 6n + D = 6n\alpha + 1 \le 7n\alpha = 7n^{\beta+1}$. Hence $n^\beta \ge N^{\frac{\beta}{\beta+1}} 7^{-\frac{\beta}{\beta+1}}$. On the other hand, $N = 6n\alpha + 1 \ge \alpha = n^\beta \ge 7^\beta$ as we may assume, without loss of generality, that $n \ge 7$. It follows that $7^{-\frac{\beta}{\beta+1}} \ge N^{-\frac{1}{\beta+1}}$. Thus

$$\alpha = n^\beta \ge N^{\frac{\beta}{\beta+1}} 7^{-\frac{\beta}{\beta+1}} \ge N^{\frac{\beta}{\beta+1}} N^{-\frac{1}{\beta+1}} = N^{\frac{\beta-1}{\beta+1}} = N^{1-\frac{2}{\beta+1}} \ge N^{1-\varepsilon}. \qquad \square$$

B Appendix: Examples

Example 14. Construct a CAPR instance in which $A = \{a_1, a_2\}$ and $C = \{c_1, c_2, c_3, c_4\}$. Each applicant has capacity 2 and each course has capacity 1. The prerequisites of both applicants are the same, and are as follows:

$$c_1 \to c_3; \qquad c_2 \to c_4.$$

The applicants have the following preference lists:

$$P(a_1) : c_1, c_2, c_4, c_3$$
$$P(a_2) : c_2, c_1, c_3, c_4$$

The sequential allocation mechanism with policy $\sigma = a_1, a_2, a_1, a_2$ will assign to applicant a_1 the bundle $\{c_1, c_4\}$ and to applicant a_2 the bundle $\{c_2, c_3\}$. Clearly, neither of the assigned bundles fulfils the prerequisites.

Now suppose that in the sequential allocation mechanism an applicant is allowed to choose the most-preferred undersubscribed course *for which she already has all the prerequisites*. Let the policy start with a_1, a_2. Applicant a_1 can choose neither c_1 nor c_2, as these courses require a prerequisite that she is not assigned yet. So she chooses c_4. Similarly, applicant a_2 will choose c_3. When these applicants are allowed to pick their next course, irrespective of the remainder of the policy, a_1 must choose c_2 and a_2 must choose c_1. So in the resulting matching M we have $M(a_1) = \{c_2, c_4\}$ and $M(a_2) = \{c_1, c_3\}$. This matching is clearly not Pareto optimal, since both applicants will strictly improve by exchanging their assignments. □

Example 15. Consider a CACR instance in which $A = \{a_1, a_2, \ldots, a_n\}$ is the set of applicants and $C = \{c_1, c_2, \ldots, c_{2n}\}$ is the set of courses, for some $n \geq 1$. Assume that each course has capacity 1, whilst each applicant has capacity n and ranks all courses in increasing indicial order. Also suppose that the prerequisites for each applicant are as follows:

$$c_1 \to c_2 \to \cdots \to c_{2n}.$$

There are n POMs: in the POM M_i $(1 \leq i \leq n)$, a_i is assigned the set of courses $\{c_{n+1}, c_{n+2}, \ldots, c_{2n}\}$ and no course is assigned to any other applicant. Given any policy, let a_i be the first applicant considered during an execution of SM-CAPR. When Explore is called on c_1, $2n$ courses are checked, then Explore is called on c_2 leading to $2n - 1$ courses being checked, and so on. This continues until Explore is called on c_{n+1}, leading to the matching M_i being constructed at this point. Note that even if the entire recursive process were to halt as soon as $|S| > q(a_i)$, the total number of courses checked at this step of SM-CAPR is still $\Omega(n^2)$. Similarly, the number of courses checked at each other applicant's turn in the policy is also $\Omega(n^2)$; the only difference is that in each such case SM-CAPR determines that c_{n+i} is full immediately, for each i $(1 \leq i \leq n)$. The overall number of steps used by SM-CAPR is then $\Omega(n^3) = \Omega(Ln_2)$. □

Example 16. SM-CAPR is not able to produce all POMs, even in the case when there are only two applicants a_1, a_2 and the capacity of each course is 1. We provide two instances to illustrate this. In I_1 the prerequisites of all applicants are the same. In I_2 they are different, but each course has at most one prerequisite.

In I_1, we have $C = \{c_1, c_2, c_3\}$, and course c_1 has two prerequisites as follows:

$$c_1 \to c_2; \qquad c_1 \to c_3. \tag{1}$$

Each applicant has capacity 3 and the following preference list: c_1, c_2, c_3.

Depending on the policy, SM-CAPR outputs either the matching that assigns all three courses to a_1, or the matching that assigns all three courses to a_2. However, it is easy to see that the two matchings that assign c_2 to one applicant and c_3 to the other one are also Pareto optimal.

In I_2, we have $C = \{c_1, c_2, c_3\}$. Now the prerequisites of the applicants are different:

$$c_1 \to_{a_1} c_3; \qquad c_2 \to_{a_2} c_3, \tag{2}$$

Each applicant has capacity 2 and their preferences are as follows:

$$P(a_1) : c_1, c_2, c_3 \qquad P(a_2) : c_2, c_1, c_3.$$

There are 3 different POMs, as follows:

$$\begin{array}{ll} M_1(a_1) = \{c_1, c_3\}, & M_1(a_2) = \emptyset; \\ M_2(a_1) = \emptyset, & M_2(a_2) = \{c_2, c_3\}; \\ M_3(a_1) = \{c_2\}, & M_3(a_2) = \{c_1, c_3\}. \end{array}$$

SM-CAPR outputs M_1 with policy $\sigma = a_1, a_2$ and M_2 with policy $\sigma = a_2, a_1$. Notice that M_3 cannot be obtained by SM-CAPR. □

Example 17. Let I be an instance of CAPR in which there are two applicants, a_1, a_2, each of which has capacity 2, and four courses, c_1, c_2, c_3, c_4, each of which has capacity 1. The prerequisites of both applicants are the same, and are as follows:

$$c_1 \to c_2; \qquad c_3 \to c_4.$$

The utilities of the courses for the applicants are as follows:

$$\begin{aligned} u_{a_1}(c_1) = u_{a_2}(c_3) = 6 \\ u_{a_1}(c_3) = u_{a_2}(c_1) = 4 \\ u_{a_1}(c_4) = u_{a_2}(c_2) = 3 \\ u_{a_1}(c_2) = u_{a_2}(c_4) = 0 \end{aligned}$$

Regardless of the policy, SM-CAPR constructs the matching $M = \{(a_1, c_1), (a_1, c_2), (a_2, c_3), (a_2, c_4)\}$. M is not a POM as it is dominated by $M' = \{(a_1, c_3), (a_1, c_4), (a_2, c_1), (a_2, c_2)\}$. □

Example 18. Consider a CAPR instance I in which $A = \{a_1, a_2\}$ and $C = \{c_1, c_2, \ldots, c_n\}$ for some $n \geq 1$. Let the preferences of the applicants be

$$P(a_1) : c_1, c_2, \ldots, c_n \qquad P(a_2) : c_n$$

and let $c_i \to c_{i+1}$ for each applicant ($1 \leq i \leq n-1$). Assume that a_1 has capacity n, whilst the capacity of a_2 and the capacity of every course is 1.

There are two POMs in I: if SM-CAPR is executed relative to the policy a_1, a_2 then we obtain the POM M_1 that assigns all the n courses to a_1 and nothing to a_2; if instead the policy is reversed, we obtain the POM M_2 that assigns nothing to a_1 and the single course c_n to a_2. Hence executing SM-CAPR relative to different policies can give rise to POMs with arbitrarily large difference in cardinality. It follows that SM-CAPR is not in general a constant-factor approximation algorithm for MAX POM CAPR. However, notice that in this example the cardinality of the downset of each course is not bounded by a constant; enforcing such a condition could improve the approximation possibilities. □

Example 19. The SM-CACR mechanism can be extended without difficulty to the variant of CACR (considered in this example only) in which corequisites can be applicant-specific. However it is no longer true that the mechanism is capable of reaching all POMs relative to a suitable policy, as we now illustrate. Consider a CACR instance with two applicants and three courses. Suppose that each applicant has capacity 2, and that each course has capacity 1. Assume that the applicants have the following preference lists:

$$\begin{aligned} P(a_1) :& \quad c_1, c_2, c_3 \\ P(a_2) :& \quad c_2, c_1, c_3 \end{aligned}$$

Assume that each applicant has as corequisites the first and last courses on her list. Then SM-CACR will return the matching $\{(a_i, c_i), (a_i, c_3)\}$ if the first applicant in the policy is a_i ($i \in \{1, 2\}$). However the matching $M = \{(a_1, c_2), (a_2, c_1)\}$ is also Pareto optimal and cannot be obtained by SM-CACR. □

Katarína Cechlárová
Institute of Mathematics, Faculty of Science
P.J. Šafárik University
Košice, Slovakia
Email: katarina.cechlarova@upjs.sk

Bettina Klaus
Faculty of Business and Economics
University of Lausanne
Lausanne, Switzerland
Email: Bettina.Klaus@unil.ch

David F. Manlove
School of Computing Science
University of Glasgow
Glasgow, United Kingdom
Email: david.manlove@glasgow.ac.uk