Pareto optimal matchings with lower quotas

Katarína Cechlárová and Tamás Fleiner

Abstract

We consider the problem of allocating applicants to courses, where each applicant has a capacity, possibly greater than 1, and a subset of acceptable courses that she ranks in strict order of preference. Each course has a lower and an upper quota, indicating that if it is assigned some applicants then their number has to be between these two bounds. We further suppose that applicants extend their preferences over courses to preferences over bundles of courses lexicographically.

In this setting we present several algorithmic results concerned with the computation of Pareto optimal matchings (POMs). Firstly, we extend the Serial Dictatorship with Project Closures mechanism to the case when an applicant can be assigned more than one course. We show that that this mechanism is strategy-proof against reordering strategies only for some picking sequences and no mechanism is strategy-proof against dropping manipulations. We further show the intractability of the following problems: deciding about the Pareto optimality of a given matching, computation of a POM with maximum cardinality and computation of a POM in case of indifferences.

1 Introduction

We study two-sided matching markets with one-sided preferences. One side of the market, the set A (of agents, students, workers, researchers) has strict ordinal preferences over the other side, represented by the set C (of objects, schools, courses, firms, projects etc), but not vice versa. The aim is to match agents to objects. As preferences of agents are often conflicting, a suitable compromise for optimality notion has to be chosen. One of the most popular one is Pareto optimality, widely studied in the literature on matchings [1, 2, 10, 11, 12, 13, 20, 22, 28]. A matching is Pareto optimal if it is not possible to improve the match for at least one agent without making some other agent worse off.

The simplest matching problem arises when each agent can be matched to at most one object and vice versa. This case is often called the *House Allocation problem* and the famous Serial Dictatorship mechanism (SD for brevity) can be applied. SD means the following: agents are ordered into a picking sequence randomly or according to some rule and taken in this order, everybody who has her turn chooses the most preferred among objects that are still available.

Serial Dictatorship [1, 22] is also called queue allocation in [28], Greedy-POM in [2], or sequential mechanism in [9, 6]. Several authors proved independently that a matching in a house allocation problem is Pareto optimal if and only if it is produced by SD (Svensson in 1994 [28], Abdulkadiroğlu and Sönmez in 1998 [1], Abraham et al. in 2004 [2], Brams and King in 2005 [10]). However, this equivalence holds only in the one-to-one case, in many-to-many setting some additional conditions have to be fulfilled, see e.g., Cechlárová et al. [11].

Serial Dictatorship can easily be extended to the case when objects come in several copies (or, equivalently, if courses, firms, projects, etc, have greater capacity): simply, in the realization of SD the agents are allowed to choose each object several times, up to its capacity.

The case when agents may be assigned more than one object has also been considered by several authors [6, 10, 11, 12, 13]. The variant of Serial Dictatorship that allows one agent to pick immediately the best bundle of objects may output a POM that is very unfair. A generalization of SD in which an agent is allowed to pick only one object on her turn, but she is put into the picking sequence several times, is called a *Sequential Mechanism* to stress the difference. Some of the authors assumed that all objects are acceptable for each agent and hence all objects are assigned in each matching. Aziz et al. [6], Brams and King [10], Bouveret and Lang [9] explored several possible rules for creating the picking sequence with respect to the question which assignments can be achieved in this case.

Agents may also consider only certain sets of objects acceptable. This may be implied by various real-life restrictions, for example in connection with prices of objects and budgets of agents or some kind of complementarities. The case when feasible sets of objects form families closed with respect to inclusion was treated by Cechlárová et al [11] and the case when object represent university courses and certain prerequisites have to be fulfilled was dealt with by Cechlárová, Klaus and Manlove [13].

Another interesting case arises when objects represent courses (projects, or study programmes) that have lower quotas. Lower quotas may mean that each object has to receive at least the required number of assignees, like in [17]. When objects represent say projects or courses (see e.g. [8, 24]), then it is more plausible to assume that if the number of students (workers, researchers etc) assigned to a project (course) does not meet its lower quota then the project (course) will simply stay closed (i.e., it is assigned no agents). The latter case has been dealt with by Biró et al. [8] and by Arulselvan et al. [3]. Biró et al. [8] considered prospective students and study programmes. They assumed that study programmes also have preferences over the students (say, derived from students' academic performance) and the solution concept was stability. Arulselvan et al. [3] assumed that each possible agent-object partnership carries some weight and they considered the computational complexity of maximum weight matchings fulfilling lower and upper quotas.

As far as the authors are aware of, Pareto optimality of matchings with lower quotas of objects was studied in three works.

Goto et al. [17] considered a matching model in the context of students and schools. In their model each school is acceptable for each student and there is a family of regions (subsets of schools) that have lower as well as upper quotas. A matching is feasible if for each region the set of assigned students obeys its lower as well as upper quota. The authors showed that the problem to decide whether a feasible matching exists is NP-complete (even in the case when there are only lower quotas or when there are only upper quotas), unless the regions form a hierarchical family, i.e., two different regions are either disjoint or one of them is a subset of the other. For hierarchical lower quotas Goto et al. proposed several mechanisms and described the properties of the matchings that are produced by them.

In the model with project closures Monte and Tumennasan [24] proposed an algorithm called *Serial Dictatorship with Project Closures* (SDPC for short) for the case that each project is acceptable for each student and a student can be assigned to at most one project. Monte and Tumennasan showed that their mechanism is strategy-proof, it always produces a Pareto optimal matching, but there may be Pareto optimal matchings that cannot be obtained by SDPC. Kamiyama [19] generalized SDPC to the case when students declare some projects unacceptable and the sets of applicants assigned to a project may be required to fulfill some additional restrictions. However, Kamiyama did not give any complexity bound for his algorithm.

Finally, our main algorithmic tool is network flows. Let us mention that this approach has been used to find Pareto optimal matchings by Cecharová et al. [12] in the many-to-many case with indifferences and by Athanassoglu and Sethuraman [4] in the house allocation problem with fractional endowments.

Our contribution

In this paper we use the terminology of applicants and courses and give a further generalization of the SDPC, namely for the case when each applicant can be assigned to several courses (many-to-many case) and she is allowed to declare some courses unacceptable. We call this mechanism the *Generalized Serial Dictatorship with Project Closures*, briefly GSDP.

We assume that applicants rank order individual courses and these preferences are extended to preferences over bundles lexicographically. Lexicographic preferences have been intensively studied from the axiomatic perspective [7], [15], [25] but they are also a popular approach in matching theory, see [11, 12, 13, 26, 21]. Lexicographic preferences are crucial for the correctness of GSDPC.

In Section 2 we introduce the model and the necessary notions. Section 3 is devoted to the description, proof of correctness and complexity analysis of GSDPC. We also prove that given a matching, it is NP-hard to decide whether it is dominated. Strategic issues are treated in Section 4. In Section 5 we explore some structural properties of Pareto optimal matchings with lower quotas. Finally, in Section 6 we show that in case of indifferences the problem of finding a POM is NP-hard. Section 7 concludes.

2 Definitions and notation

A is the set of m applicants, C is the set of n courses. We suppose that each course $c \in C$ has a lower quota $\ell(c)$ and an upper quota $u(c)$, with $\ell(c) \le u(c)$. Each applicant $a \in A$ has a capacity $q(a)$ and a strict linear order (preference) $P(a)$ on a subset of C. This preference will simply be represented as an ordered list of courses, from the most preferred to the least preferred one. Sometimes we shall enclose a part of a preference list by a pair of square brackets, to indicate that the courses within them can be listed in an arbitrary strict order. With some abuse of notation, we shall say that a course c is *acceptable* for applicant a if $c \in P(a)$. The sum of applicants' capacities is denoted by $Q = \sum_{a \in A} q(a)$.

The preference profile $\mathcal{P}$ is the m-tuple of applicants' preferences, q, ℓ and u are the vectors of applicants' capacities and quotas of courses. The quadruple $I = (A, C, \mathcal{P}, q, \ell, u)$ is an instance of the *Course Allocation problem with Lower Quotas* (CALQ for brevity).

An *assignment* M is a subset of $A \times C$. The set of applicants assigned to a course c will be denoted by $M(c) = \{a \in A; (a, c) \in M\}$ and similarly, the set of courses assigned to an applicant a is $M(a) = \{c \in C; (a, c) \in M\}$.

An assignment M is a *matching* if the following two conditions (i) and (ii) are fulfilled:

(i) $M(a) \subseteq P(a)$, $|M(a)| \le q(a)$ for each $a \in A$;

(ii) $\ell(c) \le |M(c)| \le u(c)$ or $M(c) = \emptyset$ for each $c \in C$.

An assignment M is called a *partial* matching if it fulfils (i) and the following weaker condition:

(ii') $|M(c)| \le u(c)$ for each $c \in C$.

This means that in a partial matching some courses may violate their lower quotas.

Given a partial matching M, a course c is

- *open* if $M(c) \neq \emptyset$,
- *closed* if $M(c) = \emptyset$,
- *full*, if $M(c) = u(c)$;
- *demanding* if it is open and $|M(c)| < \ell(c)$;

- *satisfied* if $|M(c)| \geq \ell(c)$.

We denote by $\mathscr{O}(M)$, $\mathscr{C}(M)$, $\mathscr{F}(M)$, $\mathscr{D}(M)$ and $\mathscr{S}(M)$ the set of open, closed, full, demanding and satisfied courses in a partial matching M.

Given a partial matching M, its *residual demand* is

$$RD(M) = \sum_{c \in \mathscr{D}(M)} (\ell(c) - |M(c)|).$$

Notice that a partial matching M is a matching if and only if $RD(M) = 0$.

We suppose that applicants express their preferences only over individual courses and they compare bundles of courses *lexicographically* ([7, 15]). This means that an applicant prefers a bundle S to a bundle T if and only if her most preferred course in the symmetric difference $S \oplus T$ belongs to S. Notice that the lexicographic ordering of bundles of courses generated by a strict preference order $P(a)$ is also strict.

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

A Pareto optimal matching, briefly a POM, is a matching that is not dominated by any other matching. As the dominance relation is a partial order over the set of matchings and the set of all matchings is finite in an instance of CALQ, a Pareto optimal matching exists for each instance of CALQ.

The main tool in our algorithm are network flows. We refer to the monograph of Schrijver [27] for the basic terminology and properties of flows in networks. For reader's convenience we recall here the basic notions and results. A *network* is a directed graph $N = (V, E)$ with two distinguished vertices s and t, (called the *source* and the *sink*) and a *capacity* function $w : E \to \mathbb{R}^+$. For $S \subseteq E$, the symbol $w(S)$ denotes the total capacity of arcs in S. The set of arcs *entering* a vertex v is denoted by $\delta^{in}(v)$, and the set of arcs *leaving* a vertex v is denoted by $\delta^{out}(v)$; this notation is extended to a set of vertices U in place of a single vertex v in the natural way. A *flow* in N is a mapping $f : E \to \mathbb{R}^+$ that obeys the arc capacities, i.e., $f(e) \leq w(e)$ for each arc $e \in E$, and fulfils the *flow conservation condition*, i.e., $\sum_{e \in \delta^{in}(v)} f(e) = \sum_{e \in \delta^{out}(v)} f(e)$ holds for each vertex except the sink and the source. The *value* of flow f is the amount of flow leaving the source, i.e., $value(f) = \sum_{e \in \delta^{out}(s)} f(e)$. We shall use the Integrality Lemma (see Corollary 11.2c and Theorem 11.1 in [27]) which ensures that if all the arc capacities in a network N are integral and N admits a flow of size K then N also admits an *integral* flow of size K. Notice also that an integer $s - t$ flow of size K is a linear combination of K directed $s - t$ paths.

A *cut* in a network is a subset U of vertices that contains source s and does not contain sink t. The famous Maxflow-Mincut theorem states that in each network N, for each flow f and each cut U, we have $value(f) \leq w(\delta^{out}(U))$; moreover, a flow f is of maximum value if and only if there exists a cut U for which this inequality is fulfilled as equality [27, Theorem 10.2].

3 Algorithm

First we show that the classical SD may fail to find a matching in an instance of CALQ, even in the simplest case with just two courses c_1, c_2, both with lower quota equal 2, and two applicants a_1, a_2 each with capacity equal 1. (Another example with more applicants and courses was given in [24].) Suppose applicant a_1 prefers course c_1 to course c_2 and applicant a_2 prefers course c_2 to course c_1. Irrespectively of the picking sequence, the classical SD produces the matching $M = \{(a_1, c_1), (a_2, c_2)\}$ which is not feasible, as it violates the lower quotas of both courses.

In this section we shall generalize *Serial Dictatorship with Project Closures* mechanism of Monte and Tumennasan [24] and Kamiyama [19]. Notice that these papers work only in the one-to-many case, i.e., they assume applicants' capacities equal 1. We want to treat the case where an applicant can be assigned to several courses.

Our mechanism will be called the *Generalized Serial Dictatorship with Project Closures* (GSDPC for short). Notice that this mechanism requires that the complete preference profile is known at its beginning.

First, applicants are ordered into a picking sequence σ, each applicant a appears $q(a)$ times in σ. Matching M_0 is initialized to be the empty matching and for each applicant a, pointer $best(a)$ is set to the first course in a's preference list $P(a)$.

GSDPC works in rounds. In the beginning of round k we have a partial matching M_{k-1} and the next applicant a from σ is taken. The mechanism tries to add a new pair to the current *partial* matching M_{k-1} by assigning a to her most preferred course in $P(a)\backslash M_{k-1}(a)$ while ensuring that no course will exceed its upper quota and all the currently open demanding courses (possibly a newly open course too) can still fulfil their lower quotas. While a is treated, pointer $best(a)$ moves down a's preference list until she is assigned a course, or until all courses in her list are unsuccessfully checked. When we explore whether course c can be assigned to a, we check whether c has enough free capacity, i.e., whether $|M_{k-1}(c)| < u(c)$, but also whether there are still enough applicants interested in all the demanding courses.

We keep for each applicant a the variable $rq(a)$, denoting her *residual capacity.* At the start of the algorithm, $rq(a)$ is set to $q(a)$ and in the beginning of each round where a is treated, $rq(a)$ is decreased by 1. Further, we denote by RC_{k-1} the sum of residual capacities of all the applicants in M_{k-1}. The vertices of network $N(M_{k-1})$ are applicants, courses, source s and sink t. The arc are:

First layer: (sa) for each $a \in A$, capacity $w(sa) = rq(a)$.

Second layer: (ac) for each $a \in A$ and each $c \in P(a)$, capacities of these arcs are 1.

Third layer: (ct) for each $c \in C$ with $w(ct) = \ell(c) - |M_{k-1}(c)|$ if $c \in \mathscr{D}(M_{k-1})$ and $w(ct) = 0$ otherwise.

Arcs from the second layer are deleted during the algorithm in a way that we now describe. Round k starts with network $N(M_{k-1})$ and deals with applicant a who is in position k in the picking sequence σ. We take course $c = best(a)$ and delete the arc (ac) from $N(M_{k-1})$. If $|M_{k-1}(c)| = u(c)$, we increase $best(a)$ and move to the next course in her preference list, otherwise pair (a, c) is provisionally added to M_{k-1} to obtain a new (provisional) partial matching M_k. Depending on the properties of course c, capacities of arcs of the third layer are updated as described below, to get network $N(M_k)$ and we check whether this network admits a flow of value $RD(M_k)$. If such a flow exists, pair (a, c) becomes a fixed addendum to M_{k-1} and network $N(M_k)$ together with the flow f_k is made the starting point of the new round. Otherwise we return to M_{k-1} and $N(M_{k-1})$, $best(a)$ is moved to the next position in a's list (if any) and the next course in the preference list of a is explored.

For the correctness of the algorithm the following lemma is crucial.

Lemma 1. *There exists a matching M such that $M_k = M_{k-1} \cup \{(a, c)\} \subseteq M$ if and only if $N(M_k)$ admits a flow f_k of value $RD(M_k)$.*

Proof. If the network $N(M_k)$ admits a flow f_k of value $RD(M_k)$ then due to the Integrality Lemma it also admits an integral flow of the same value. This flow defines the augmentation of $M_{k-1} \cup \{(a, c)\}$ into a matching M by adding all the pairs (a', c') corresponding to arcs $(a'c')$ in $N(M_k)$ with nonzero flow f_k.

Conversely, let there exist a matching M that contains $M_{k-1} \cup \{(a,c)\}$. We take all pairs $(a',c') \in M \backslash (M_{k-1} \cup \{(a,c)\})$, define the flow f_k to be equal 1 along all the corresponding arcs $(a'c')$ and complete it along the arcs from the first and the third layer to ensure the flow conservation condition. It is easy to see that no arc capacity is exceeded. Further, since all the courses in $\mathscr{D}(M_k)$ fulfil their lower quotas in M, the desired value of f_k is also achieved. □

Theorem 2. *GSDPT outputs a Pareto optimal matching.*

Proof. Let us denote the output of GSDPT by M. First we show that M is a matching. Using the Maxflow - Mincut theorem, in each round k we have

$$0 \leq RD(M_k) \leq value(f_k) \leq w(\delta^{out}\{s\}) = RC_k$$

In the last round r the residual capacity $RC_r = 0$, hence $RD(M_r) = 0$, so $M = M_r$ is a matching.

Now we show that M is a POM. To get a contradiction, suppose that there exists a matching M' that dominates M. So there exists an applicant a who prefers $M'(a)$ to $M(a)$; let $c(a)$ be the most preferred course in $M'(a) \backslash M(a)$ and let $k(a)$ be the round where a and $c(a)$ were considered. Let a be the applicant for whom $k(a)$ is minimum. Then M' restricted to pairs assigned up to the round $k(a) - 1$ is equal to $M_{k(a)-1}$. Matching M' proves that there exists a matching augmenting $M_{k(a)-1} \cup \{(a, c(a))\}$, so the network $N(M_{k(a)-1} \cup \{(a, c(a))\})$ contains the desired flow, hence the algorithm should have assigned $c(a)$ to a, a contradiction. □

To perform the algorithm efficiently, we shall not compute flow f_k in round k from scratch, we rather see how to update network $N(M_{k-1})$ to $N(M_k)$ and to extend flow f_{k-1} to get a flow f_k of size $RD(M_k)$.

At the beginning of round k where applicant a is dealt with, firstly $rq(a)$ is decreased by 1 and so the capacity of arc (sa) is decreased by 1. When exploring the pair $(a, c = best(a))$, arc (ac) is deleted from $N(M_{k-1})$ and other changes are performed, depending on the character of c. We distinguish several cases and notation M_k always means the provisional partial matching $M_{k-1} \cup \{(a,c)\}$. Recall that $value(f_{k-1}) = RD(M_{k-1})$ and for each course $c \in \mathscr{C}(M_{k-1})$ the capacity of arc (ct) is 0.

(a) $c \in \mathscr{C}(M_{k-1})$ and $\ell(c) = 1$. Then $RD(M_k) = RD(M_{k-1})$. This means that the capacities of the arcs of the third layer do not change, but perhaps flow f_{k-1} used up the whole residual capacity of applicant a. In other words, flow f_{k-1} along arc (sa) was equal to $rq(a)$. If that was the case, we have to perform one search in $N(M_k)$ to make up for the lost one unit of flow value.

(b) $c \in \mathscr{C}(M_{k-1})$ and $\ell(c) > 1$. Then $RD(M_k) = RD(M_{k-1}) + \ell(c) - 1$. This means that the capacity of arc (ct) is increased from zero to $\ell(c) - 1$. Again, possibly one unit of flow value was lost because of $f_{k-1}(sa) = rq(a)$, so to get the flow of the desired value, we need at most $\ell(c)$ searches in $N(M_k)$.

(c) $c \in \mathscr{D}(M_{k-1})$. Then $RD(M_k) = RD(M_{k-1}) - 1$, namely capacity of arc (ct) is decreased by 1. Now, if we had $f_{k-1}(ac) = 1$, nothing has to be done. If the flow f_{k-1} into vertex c used exclusively arcs different from (ac), we choose one of them arbitrarily, say $(a'c)$, and decrease the flow along this arc and along the arc (sa') by one.

(d) $c \in \mathscr{S}(M_{k-1})$. Now $RD(M_k) = RD(M_{k-1})$. Again, possibly one unit of flow value was lost if $f_{k-1}(sa) = rq(a)$, so one search in $N(M_k)$ is enough to make up for the one unit of flow value.

To be able to derive the complexity bound for the algorithm, let us first estimate how many operations we need to check whether the pair (a, c) can be added to the current partial matching. Each admissible pair (a, c) is explored at most once, hence we may need $|L_P|$ checks, where $|L_P|$ is the total length of the preference list. In the worst case, when the algorithm tries to open course c that was closed so far (point (b)), as many as $\ell(c)$ searches in network are needed. One search in a network is linear in the number of its arcs, and all the networks in the algorithm have at most $|A| + |L_P| + |C|$ arcs. Therefore the complexity bound of the algorithm is $O(|L_P|^2 max_{c \in C} \ell(c))$, which can be bounded by $O(m^3 n^2)$, where $m = |A|$ and $n = |C|$. The previous discussion is summarized as follows:

Theorem 3. *Algorithm GSDPC correctly computes a POM in a* CALQ *instance in* $O(|L_P|^2 max_{c \in C} \ell(c))$ *time.*

Monte and Tumennasan [24] showed that SDPC is not able to produce some Pareto optimal matchings. We strenghten their result to show that this holds already for two applicants and lower quotas not exceeding 2.

Example 4. Let the set of courses be $C = \{c_1, c_2, r\}$, all courses have lower as well as upper quota equal 2. The set of applicants is $A = \{a_1, a_2\}$, each has capacity 1 and their preferences are as follows:

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

If a_1 is the first applicant in the picking sequence, the resulting POM will be $M_1 = \{(a_1, c_1), (a_2, c_1)\}$; otherwise the GSDPC will output $M_2 = \{(a_1, c_2), (a_2, c_2)\}$. Notice that $M_3 = \{(a_1, r), (a_2, r)\}$ is also a POM, but it cannot be obtained by GSDPC. □

Since GSDPC does not produce all POMs, an approach that takes any matching and improves it until a POM is obtained could be considered. However, here we show that this approach is also unlikely to lead to a polynomial algorithm, as testing Pareto optimality for CALQ is NP-hard.

> **Problem** CALQ-DOMINANCE.
> **Instance:** Instance I of CALQ, a matching M.
> **Question:** Does there exist a matching that dominates M?

Theorem 5. CALQ-DOMINANCE *is NP-complete even in the case when* $q(a) = 1$ *for each* $a \in A$ *and no lower quota of a course exceeds 3.*

Proof. CALQ-DOMINANCE is in NP, since given a matching M' it can be verified in polynomial time whether it dominates M. To prove completeness, take the following NP-complete problem:

> **Problem** EXACT-3-COVER.
> **Instance:** $J = (X, \mathcal{T})$, where $X = \{x_1, x_2, \ldots, x_{3m}\}$ and $\mathcal{T} = \{T_1, T_2, \ldots, T_n\}$ is a collection of 3-element subsets of X.
> **Question:** Does there exist $\mathcal{T}' \subseteq \mathcal{T}$ such that $|\mathcal{T}'| = m$ and $\mathcal{T}'$ covers X?

To get some intuition, first we present a simpler proof that requires a course with a large lower quota.

For an instance J of EXACT-3-COVER we construct an instance I of CALQ. The set of applicants is $A = \{b_1, \ldots, b_{3m}\}$, i.e., there is one applicant for each element $x \in X$. The set of courses is $C = \{c_1, \dots, c_n, d\}$. This means there is one *regular* course c_i for each set $T_i \in \mathcal{T}$ with $\ell(c_i) = u(c_i) = 3$ and one *super* course d with $\ell(d) = u(d) = 3m$.

The preferences of applicants are as follows (recall that the courses written in square brackets can appear in any strict order):

$$P(b_i) : [c_j; x_i \in T_j], d \qquad \text{for } i = 1, \dots, 3m.$$

Let the matching M be such that all applicants are assigned to the super course d.

If there exists a matching M' that dominates M then any applicant that improves with respect to M causes that course d must be closed. This means that all applicants, so as not to become worse off, must be assigned to some regular course. Because of the preferences of applicants and the lower and upper quotas of courses, this means that M is dominated if and only if J admits an exact 3-cover.

Conversely, if J admits an exact cover $\mathcal{T}'$, then assigning each applicant b_i to the course corresponding to a set in $\mathcal{T}'$ that covers x_i provides a matching that dominates M.

To achieve that no course will have lower quota greater than 3, we modify the previous proof using a trick that replaces the super course by many courses with small lower and upper quotas and defines applicants' preferences over them in a cyclic manner.

Now we define an instance I of CALQ as follows. The set of applicants is $A = B \cup Y \cup Z$, where $B = \{b_1, \dots, b_{3m}\}$, $Y = \{y_1, \dots, y_{3m}\}$ and $Z = \{z_1, \dots, z_{3m}\}$, i.e., there are three applicants b_i, y_i, z_i for each $x_i \in X$. The set of courses is $C \cup D \cup E$, where $C = \{c_1, \dots, c_n\}$, $D = \{d_1, \dots, d_{3m}\}$ and $E = \{e_1, \dots, e_{3m}\}$. Courses in C are called *regular* courses, c_j corresponds to $T_j \in \mathcal{T}$ and $\ell(c_j) = u(c_j) = 3$, for $j = 1, 2, \dots, n$. Further, $\ell(d_i) = u(d_i) = 3$ and $\ell(e_i) = u(e_i) = 2$ for each $i = 1, 2, \dots, 3m$.

Preferences of applicants are as follows. Here, and in the rest of the proof, index $i-1$ for $i = 0$ means $3m$.

$$\begin{array}{lll} P(b_i): & [c_j; x_i \in T_j], d_i & \text{for } i = 1, 2, \dots, 3m \\ P(y_i): & e_i, d_i & \text{for } i = 1, 2, \dots, 3m \\ P(z_i): & e_{i-1}, d_i & \text{for } i = 1, 2, \dots, 3m \end{array}$$

Finally, let the matching M be such that $M(d_i) = \{a_i, y_i, z_i\}$ for $i = 1, \dots, 3m$ and all the other courses are closed.

Now suppose that J admits an exact cover $\mathcal{T}'$; let $\mathcal{T}' = \{T_{k_1}, T_{k_2}, \dots, T_{k_m}\}$. In this case, matching M' defined by $M'(b_i) = c_{k_j}$ if $x_i \in T_{k_j}$ and $M'(e_i) = \{y_i, z_{i+1}\}$ for all $i = 1, \dots, 3m$ (here, $z_{3m+1} = z_1$) dominates M.

Conversely, let a matching M' dominate M. Then at least one applicant $a \in A$ prefers $M'(a)$ to $M(a) = d$, hence $M'(a) \notin D$. Let $M(d) = \{b_i, y_i, z_i\}$. Now distinguish three cases.
Case 1. $a = b_i$. Then $M'(b_i) \in C$, $M'(y_i) = e_i$ and $M'(z_i) = e_{i-1}$. This means that courses e_i and $e_{i-1}$are open. The lower quotas of courses in E require that also $M'(z_{i+1}) = e_i$ and $M'(y_{i-1}) = e_{i-1}$ and so also courses d_{i+1} and d_{i-1} must be closed in M'.
Case 2. $a = y_i$. Then $M'(y_i) = e_i$. Now the lower quota of course e_i implies that also $M'(z_{i+1}) = e_i$ and so also course d_{i+1} must be closed in M'.
Case 3. $a = z_i$. Then $M'(z_i) = e_{i-1}$. Again the lower quota of course e_{i-1} implies that $M'(y_{i-1}) = e_{i-1}$ and so also course d_{i-1} must be closed in M'.

By induction, in all the cases we get that all courses in D must be closed, so all applicants $b_i, i = 1, \dots, 3m$ must be assigned to courses in C. Since the lower quotas of regular courses are equal 3, the regular courses open in M' define and exact cover of X. $\square$

4 Strategic issues

When considering strategic issues in CALQ we assume that applicants knows the picking sequence. We distinguish two types of manipulations:
reordering manipulations: changing the order of the entries in the preference list;
dropping manipulations: declaring some courses in the preference lists unacceptable.

Kamiyama [19] proved that SDPC is strategy-proof, however, he only considered reordering manipulations. Such mechanisms are important also for practical reasons. For example, in the Hungarian centralized university admission system, after the preliminary results are announced, an applicant can change her prerefence list, however, she is not allowed to withdraw a previously submitted application.

Kamiyama's result relies on the fact that each applicant has capacity 1 and it does not carry over to greater capacities, as is shown by the following example.

Example 6. Consider a CALQ instance I with $A = \{a_1, a_2\}$, $C = \{c_1, c_2\}$. Both applicants prefer course c_1 to course c_2, but capacity of a_1 is 2 while capacity of a_2 is 1. Both courses have upper quota 2, and c_2 has also lower quota equal 2. There are two Pareto optimal matchings M_1 and M_2 in I:

$$M_1(a_1) = M_1(a_2) = \{c_1\}; \qquad M_2(a_1) = \{c_1, c_2\}, M_2(a_2) = \{c_2\}.$$

Suppose that GSDPC is run with the picking sequence a_1, a_2, a_1. If both applicants act truthfully, then GSDPC outputs M_1. However, a_1 can achieve M_2, that is more preferred by her, if she falsifies her preferences to c_2, c_1. $\square$

The above example is in line with observations by Cechlárová et al. [12] and Hosseini and Larson [18] that 'interleaving' mechanisms (i.e. such allow a different agent to pick between two picks of one agent) are manipulable. Next we prove the following result that is an analogy of Theorem 11 in [12] that deals with many-to-many matchings with ties in preference lists. We call a policy σ *contiguous* if for each applicant all her occurences in σ form a contiguous interval.

Theorem 7. *GSDPC with a* contiguous *policy is strategy-proof against reordering strategies.*

Proof. Without loss of generality suppose a contiguous policy σ is

$$\underbrace{a_1, a_1, \ldots, a_1}_{q(a_1)\text{-times}}, \underbrace{a_2, a_2, \ldots, a_2}_{q(a_2)\text{-times}}, \ldots, \underbrace{a_{i-1}, a_{i-1}, \ldots, a_{i-1}}_{q(a_{i-1})\text{-times}}, \underbrace{a_i, a_i, \ldots, a_i}_{q(a_i)\text{-times}}, \ldots$$

and that applicant a_i is the first applicant in σ who benefits by falsifying her true preferences $P(a_i) = c_1, c_2, \ldots, c_r, \ldots, c_s$ to preferences $P'(a_i)$. Let M' be the matching obtained by GSDPC with a_i's preferences $P'(a_i)$ and let M be the matching obtained when a_i reports her true preferences. Obviously, a_i prefers $M'(a_i)$ to $M(a_i)$; let c_r be her most preferred course in $M'(a_i) \backslash M(a_i)$ and let p be the first round in GSDPC when a_i makes her choice. As all the preceding applicants act truthfully, the partial matchings M_{p-1} and M'_{p-1} are equal. Finally, let q be the round of GSDPC when the pair (a_i, c_r) is considered. When a_i reported her true preferences, she was not assigned course c_r, since the network $N(M_q) = N(M_{q-1} \cup (a_i, c_r))$ did not admit a flow of size $RD(M_q)$. In other words, the number of directed paths of the form $s \to a_k \to c' \to t$ for $k \geq i$ and $c' \in \mathscr{D}(M_q)$ is less than $RD(M_q)$. Let us realize that every time a new course c in a_i' preference list is considered, one such path disappears, since the arc $(a_i c)$ is deleted from the network. It is obvious that a_i cannot benefit by reordering courses that apper before c_r. Now distinguish two cases.

- Suppose a_i puts course c_r later in her preference list than in position r. The courses in $\mathscr{D}(M_{q-1})$ still need to be satisfied and since the number of available paths decreased, the corresponding network still does not admit a flow of value $RD(M_{q-1})$, so a_i cannot receive c_r in M'.
- Suppose a_i puts course c_r earlier in her preference list than in position r and so suppose that the pair (a_i, c_r) is considered in round $q' \leq q$. The residual demand of the partial matching in round q' may be smaller than that in the round where c_r was treated with true preferences of a_i. This can only happen in the case when some of the courses $c_1, c_2, \ldots, c_{r-1}$, say c_t, was closed before a_i was treated and it was open (i.e., assigned to a_i) before a_i considered c_r. However, this means that when falsifying her preferences in this way, a_i did not receive a more preferred course that she would have received when acting truthfully, so she has not improved.

Hence, a_i cannot benefit by reordering and the assertion is proved. □

For dropping strategies, we have the following stronger result.

Theorem 8. *There is no Pareto optimal mechanism for* CALQ *that is strategy-proof against dropping strategies.*

Proof. Consider a CALQ instance I with $A = \{a_1, a_2\}$, $C = \{c_1, c_2\}$. Applicant a_1 prefers course c_1 to course c_2, applicant a_2 prefers course c_2 to course c_1. Both applicants have capacity 1. Both courses have upper as well as lower quota equal 2. Clearly, I admits exactly two POMs: in M_1 both applicants are assigned to c_1, in M_2 they are assigned to c_2.

Suppose that there exists a truthful Pareto optimal mechanism φ. If φ outputs M_1 then a_2 has an incentive to lie, since when she declares only c_2 acceptable, the only POM for the new instance is M_2. On the other hand, if φ outputs M_2 then a_1 will be better off if she declares only c_1 acceptable, since in this case φ must output M_1. □

5 Structural results

Finding a POM with minimum cardinality is NP-hard even in the simplest one-to-one-case [2, Theorem 2]. A POM with maximum cardinality can be found efficiently if no other restrictions than capacities of applicants and courses are imposed [2, Theorem 1], [11, Theorem 6]. However, additional structure may make the problem intractable. Cechlárová et al. [11] prove the NP-hardness of finding a maximum cardinality POM for the price-budget case and Cechlárová et al. [13] in the presence of prerequisites. Our next theorem proves a similar result for the case with lower quotas.

Theorem 9. *Finding a POM with maximum cardinality in an instance of* CALQ *is NP-hard, even if no lower quota exceeds 4 and the capacity of each applicant is 1.*

Proof. We shall present a polynomial transformation from INDEPENDENT SET for cubic graphs. Let $\langle G, K\rangle$, where $G = (V, E)$ is a graph and K an integer, be an instance of INDEPENDENT SET. Let $V = \{v_1, \ldots, v_n\}$, $E = \{e_1, \ldots, e_m\}$. For G we construct an instance I of CALQ. The set of courses is $C = C_V \cup C_E$, where $C_V = \{c(v_1), \ldots, c(v_n)\}$, $C_E = \{c(e_1), \ldots, c(e_m)\}$. Each vertex-course $c(v_j)$ has lower and upper quota equal 4, each edge-course $c(e_j)$ has the lower as well as upper quota 1. The set of applicants in I is $A = A_V \cup A_E$, with $A_V = \{a(v_1), \ldots, a(v_n)\}$, $A_E = \{a(e_1), \ldots, a(e_m)\}$. Each applicant has capacity 1. Vertex applicant $a(v_j)$ considers only the vertex course $c(v_j)$ acceptable; for edge applicant $a(e_j)$ corresponding to edge $e_j = \{v_i, v_k\}$ the preference list is

$$P(a(e_j)) : \quad [c(v_i), c(v_k)], c(e_j).$$

Now we argue that I admits a POM of cardinality $m+K$ if and only if G has an inclusion maximal independent set of cardinality K.

So let $W \subseteq V$ be a maximal independent set of G. The corresponding matching is

$$M = \{(a(v_i), c(v_i)); v_i \in W\} \cup \{(a(e_j), c(v_i)); e_j \cap W = \{v_i\}\} \cup \{(a(e_j), c(e_j)); e_j \cap W = \emptyset\}.$$

Cardinality of M is $m+K$. To show that M is a POM we argue that it can be obtained by GSDPC with any policy that starts with all the applicants $a(v), v \in W$. On their arrival, these applicants open the set of courses, corresponding to the vertices in W; let us denote the set of these courses by C_W. Due to the lower quotas of courses, all the edge applicants corresponding to edges incident upon vertices in W will be on their turns assigned to some course in C_W; as W is independent, no conflict occurs. No other vertex course can be open during GSDPC, since W is a maximal independent set. Finally, each edge applicant $a(e)$ coresponding to an edge e not incident upon vertices in W, is assigned to $c(e)$, as the courses corresponding to its endvertices are closed, as argued above.

Conversely, let M be a POM of size $m + K$. As there are only m edge applicants, it follows that at least K vertex applicants are assigned to their associated vertex courses and hence at least K vertex courses are open. Let us denote by W the set of vertices in G corresponding to the vertex courses open in M. As the lower quota of each vertex course is 4 (i.e., the vertex degree plus 1), all the edge applicants corresponding to edges incident upon vertices in W are assigned to the corresponding vertex courses. Finally, as the capacity of each applicant is 1, no two courses corresponding to adjacent vertices can be open. Hence, W is an independent set in G of size at least K. □

6 Indifferences

Preference list of an applicant a who has indifferences can be represented by a list $(C_1^a, C_2^a, \dots, C_k^a)$ of disjoint subsets (*tiers*) of the set of courses with the following interpretation: applicant a is indifferent between all the courses in one tier and she prefers course $c_1 \in C_i^a$ to a course $c_2 \in C_j^a$ if and only if $i < j$. (Notice that if a's preferences are strict then each tier is a singleton.) Such preferences can be extended to lexicographic preference over bundles of courses by means of the *generalized characteristic vector* of a bundle [12]. The generalized characteristic vector $\rho_a(D)$ of a bundle D according to applicant a is equal to the vector $(|D \cap C_1^a|, |D \cap C_2^a|, \dots, |D \cap C_k^a|)$ and applicant a prefers bundle D_1 to bundle D_2 if $\rho_a(D_1)$ is lexicographically greater than $\rho_a(D_2)$. In the special case when all the acceptable courses of an applicant a form a single tier, applicant a simply prefers a bundle containing more courses.

We show that the problem to find a POM in CALQ in the presence of indifferences is hard, even in a very restricted case.

Theorem 10. *Finding a POM in an instance of* CALQ *with indifferences is NP-hard, even if no lower quota exceeds 4 and each applicant is indifferent between all her acceptable courses.*

Proof. We shall present a polynomial transformation from INDEPENDENT SET for cubic graphs. Let $\langle G, K\rangle$, where $G = (V, E)$ is a graph and K an integer be an instance of INDEPENDENT SET. Let $V = \{v_1, \dots, v_n\}$, $E = \{e_1, \dots, e_m\}$. For G we construct an instance I of CALQ. The set of courses is $C = C_V \cup C_E$. Each vertex-course $c(v)$, $v \in V$ has lower and upper quota equal 4, each edge-course $c(e)$, $e \in E$ has the lower as well as upper quota 1. The set of applicants in I is $A = A_E \cup \{a_0\}$; each applicant $a(e)$, $e \in E$ has capacity 1. Applicant a_0 has capacity n and she considers all the vertex courses $c(v)$ acceptable. For the edge applicant $a(e)$ corresponding to edge $e = \{v, u\}$ the only acceptable courses are $c(v), c(u)$ and $c(e)$. All applicants are indifferent between all their acceptable courses.

First we show that any independent set $W \subseteq V$ of G defines a matching in I. Namely, create a matching $M(W)$ in the following way. Applicant a_0 is assigned to all courses $c(v)$ for $v \in W$. Edge applicants corresponding to edges incident upon vertices in W are assigned to these vertex courses, other edge applicants (if any) are assigned to their associated edge courses. It is easy to see that no capacity or quota is violated, since W is independent. Further, all edge applicants are assigned in $M(W)$.

Conversely, let M be a matching in I and let $W = \{v \in V; c(v) \in \mathscr{O}(M)\}$. Because of the lower quotas of vertex courses, an open course $c(v)$ must be assigned all edge applicants associated with edges incident upon vertex v. Since each applicant has capacity 1, no two courses associated with adjacent vertices can be open. Hence, W is an independent set in G.

Now we know that matchings in I and independent sets in G are in a one-to-one correspondence. Finally we show that a matching $M(W)$ is a POM if and only if W is an independent set with maximum cardinality. Namely, suppose that there exists an independent set Z in G with $|Z| > |W|$. No edge applicant can improve, as she is full in $M(W)$ and a_0 prefers $M(Z)$ to $M(W)$ if and only if she is assigned more courses in $M(Z)$, which in turn happens if $|Z| > |W|$. The converse implication is obvious. □

7 Conclusion and open problems

In this paper we extended the algorithms for finding a Pareto optimal matching with lower quotas of courses of Monte and Tumennasan [24] and Kamiyama [19] to the case when applicants may be assigned to more than one course. In addition, we explored the strategic issues in this algorithm and proved several intractability results, namely that the problems of finding a POM with maximum cardinality, deciding Pareto optimality of a given matching and finding any POM in case with indifferences are NP-hard.

For further research we propose the following open problems.

1. Notice that to achieve the intractability results, we needed that lower quotas of some courses are 3 or more. If the lower quotas of courses do not exceed 2, will polynomial algorithms for these problems be possible?
2. Although we have shown that no Pareto optimal mechanism is strategy-proof against dropping strategies, we do not know whether successful manipulations could be efficiently computed. Which information would applicants need for a successful manipulation?
3. In an interesting related model, called the Group activity Selection problem, the lower and upper quotas of activities (in place of courses) are not imposed by some exogeneous authority (for example a law), rather they are in a sense part of participants' preferences. Darman et al. [14] studied various stability concepts like individual rationality, Nash stability, core and strong core. What could be said about Pareto optimal allocations in this case?

8 Acknowledgement

This work was supported by grants 1/0344/14 and 1/0142/15 from the Slovak Scientific Grant Agency VEGA (Cechlárová) and by OTKA grant K108383 (Fleiner). The authors gratefully acknowledge the support of COST Action IC1205 Computational Social Choice.

References

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

[2] D. Abraham, K. Cechlárová, D. Manlove and K. Mehlhorn, *Pareto optimality in house allocation problems*, LNCS. **3341**, ISAAC 2004, 3-15 (2004).

[3] A. Arulselvan, Á. Cseh, M. Groß, D.F.Manlove, J. Matuschke, *Many-to-one Matchings with Lower Quotas: Algorithms and Complexity*, LNCS 9472, 176-187, 2015.

[4] S. Athanassoglu, J. Sethuraman, *House allocation with fractional endowments*, Int J Game Theory 40: 481-513, 2011.

[5] H. Aziz, P. Biró, J. Lang, J. Lesca, J. Monnot, *Optimal reallocation under additive and ordinal preferences*, arXiv:1604.01091v1, 2016.

[6] H. Aziz, T. Walsh, L. Xia, *Possible and Necessary Allocations via Sequential Mechanisms*, IJCAI 2015, 468-474, see also arXiv:14122226v1.

[7] Barberà, S., Bossert, W., Pattanaik, P. K., 2004. Ranking sets of objects. In: Barberà, S., Hammond, P. J., Seidl, C. (Eds.), Handbook of Utility Theory. Vol. II. Kluwer Academic Publishers, Ch. 17, pp. 893–977.

[8] P. Biró, T. Fleiner, R.W. Irving, D.F. Manlove, *The college admissions problem with lower and common quotas*, Theoret. Comput. Sci. **411**(34-36), 3136-3153, 2010.

[9] S. Bouveret, J. Lang, *A general elicitation-free protocol for allocating indivisible goods*, In: Proc. 22nd IJCAI, AAAI Press, 73-78, 2011.

[10] S. J. Brams, D. L. King, *Efficient Fair Division Help the Worst off or Avoid Envy?*, Rationality and Society 17(4), 387-421, 2005.

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

[12] K. Cechlárová, P. Eirinakis, T. Fleiner, D. Magos, D. F. Manlove, I. Mourtos, E. Oceľáková, B. Rastegari, *Pareto optimal matchings in many-to-many markets with ties*, M. Hoefer (Ed.): SAGT 2015, LNCS 9347, pp. 27-39, 2015.

[13] K. Cechlárová, B . Klaus, D. F. Manlove, *Pareto optimal matchings with prerequisites constraints*, arXiv:1603.00858v1, 2016.

[14] A. Darmann, E. Elkind, S. Kurz, J. Lang, J. Schauer, G.Woeginger, *Group Activity Selection Problem*, WINE 2012, LNCS 7695, 156-169, 2012, see also arXiv:1401.8151v1, 2014.

[15] P.C. Fishburn, *Axioms for lexicographic preferences*, Review of Economic Studies 4

[16] M.R. Garey, D.S. Johnson, *Computers and intractability*, Freeman, San Francisco (1979).

[17] M. Goto, N. Hashimoto, A. Iwasaki, Y. Kawasaki, S. Ueda, Y. Yasuda, M. Yokoo, *Strategy-proof Matching with Regional Minimum Quotas*, Proc. AAMAS 2014.

[18] H. Hosseini, K.Larson, *Strategyproof quota mechanism for multiple assignment problems*, ArXiV 1507.07064, 2015.

[19] N. Kamiyama, *A note on the serial dictatorship with project closures*, Operations Research Letters 41 (2013) 559-561.

[20] P. Krysta, D. Manlove, B. Rastegari, J. Zhang, *Size versus truthfulness in the house allocation problem*, arXiv:1404.5245v1.

[21] E. Fujita, J. Lesca, A. Sonoda, T. Todo, M. Yokoo, *A Complexity Approach for Core-Selecting Exchange with Multiple Indivisible Goods under Lexicographic Preferences*, Proc. AAAI '15, 2015.

[22] M. Manea, *Serial dictatorship and Pareto optimality*, Games and Economic Behaviour 61 (2007), 316-330.

[23] D. Manlove, *Algorithmics Of Matching Under Preferences*, World Scientific Publ. (2013).

[24] D. Monte, N. Tumennasan, *Matching with quorums*, Economics Letters **120**, 14-17 (2013).

[25] P. K. Pattanaik, B. Peleg, *An axiomatic characterization of the lexicographic maximin extension of an ordering over a set to the power set*, Social Choice and Welfare 1(2), 113-122, 1984.

[26] D. Saban, J. Sethuraman, *A note on object allocation under lexicographic preferences*, Journal of Mathematical Economics 50, 283-289 (2014).

[27] A. Schrijver: *Combinatorial Optimization: Polyhedra and Efficiency*, Springer, Algorithms and Combinatorics 24, 2003.

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

Katarína Cechlárová
Institute of Mathematics
Faculty of Science, P.J. Šafárik University
Jesenná 5, Košice
Slovakia
Email: `katarina.cechlarova@upjs.sk`

Tamás Fleiner
Budapest University of Technology and Economics and MTA-ELTE Egerváry Research Group
Magyar tudósok körútja 2, Budapest
Hungary
Email: `fleiner@cs.bme.hu`