

An integer programming approach to curriculum-based examination timetabling

Alejandro Cataldo¹ · Juan-Carlos Ferrer¹ ·
Jaime Miranda² · Pablo A. Rey³ · Antoine Sauré⁴

Published online: 28 September 2016
© Springer Science+Business Media New York 2016

Abstract The examination timetabling problem (ETTP) consists in the assignment of specific dates to the exams of a set of courses assuming that the course enrollments are known. This problem is also known as post-enrollment ETTP. In this paper, we describe and solve a variant of the ETTP which has two particularities: (1) it does not assume the course enrollments as known and uses the curriculum of the degree program to evaluate potential conflicts in the exam schedules, and (2) it considers the exams and classrooms of multiple degree programs simultaneously. We refer to this variant of the ETTP as curriculum-based examination timetabling problem (CB-ETTP), a problem faced by many universities worldwide, being the Universidad Diego Portales (UDP) in Santiago of Chile one of them. To the best of our knowledge, this problem has not been described as such in the ETTP literature. We propose an approach to solve the CB-ETTP consisting of four sequential stages. The first stage

✉ Alejandro Cataldo
aecatald@uc.cl

Juan-Carlos Ferrer
jferrer@ing.puc.cl

Jaime Miranda
jmirandap@fen.uchile.cl

Pablo A. Rey
pablo.rey.cl@gmail.com

Antoine Sauré
asaure@uottawa.ca

¹ School of Engineering, Pontificia Universidad Católica de Chile, Av. Vicuña Mackenna 4860, Santiago, Chile

² Department of Management Control and Information Systems, School of Economics and Business, Universidad de Chile, Diagonal Paraguay 257, Santiago, Chile

³ Department of Management Control and Information Systems, School of Economics and Business, Universidad de Chile, Santiago, Chile

⁴ Telfer School of Management, University of Ottawa, 55 Laurier Avenue East, Ottawa, ON K1N 6N5, Canada

groups courses into clusters and generates classroom configurations called room patterns. The second stage assigns time slots and room patterns to course clusters. Then, the third stage assigns time slots and room patterns to individual courses. Finally, the fourth stage generates a definitive exam schedule assigning specific rooms to each course exam. We evaluate the performance of the proposed approach by applying it to real-world instances generated based on data provided by the Faculty of Engineering at the UDP. The results show a reduction in the number of conflicts and rescheduling with respect to the current exam scheduling practice used by this university.

Keywords Scheduling · Examination timetabling · Curriculum-based · Integer programming · Optimization · Problem decomposition

1 Introduction

Educational institutions routinely face a range of scheduling problems involving courses, classrooms, examinations and other aspects of their daily activities. In the operations research (OR) literature, this class of problems is known as timetabling. Timetabling problems typically involve a time horizon divided into a finite number of time periods or slots, a finite set of resources (e.g., classrooms, laboratories and professors), a finite set of events to be scheduled (e.g., examinations, classes, etc.) and a set of operational constraints (Qu et al. 2009b). These problems are usually NP-hard.

Solution approaches to timetabling problems have been designed for scheduling educational activities (Birbas et al. 2009; Jat and Yang 2011), sporting events (Duran et al. 2007; Goossens and Spieksma 2012), personnel at hospitals and other healthcare centers (Burke et al. 2004b, 2009; Gendreau et al. 2007; Maenhout and Vanhoucke 2010), medical patient appointments (Gupta and Denton 2008; Patrick et al. 2008; Sauré et al. 2012), air transport of cargo and passengers (Derigs and Friederichs 2013; Medard and Sawhney 2007) and drivers at public transit and rail transport companies (Leone et al. 2011; Schöbel 2012; Lezaun et al. 2007), among other examples.

The scheduling problems faced by educational establishments are often divided into two classes: course timetabling problems (CTTPs) and examination timetabling problems (ETTPs). CTTPs are further classified as post-enrollment course timetabling problems (PE-CTTPs) or curriculum-based course timetabling problems (CB-CTTPs) (Rudová et al. 2011). The main difference between the last two types of problems lies in the data used to determine the number of students enrolled in the various courses constituting different programs of study. In PE-CTTPs, the final number of students registered in each course is assumed to be known allowing scheduling conflicts to be determined directly. In CB-CTTPs, on the other hand, only an estimation of the number of registrants in each course is available and thus scheduling conflicts must be determined solely from the configuration of each program “curriculum”. In other words, schedule conflicts are determined from the distribution of the program’s constituent courses over the number of academic terms normally required to complete it.

Among recently published articles on educational timetabling, Lach and Lübbecke (2012) provide a detailed description of CTTPs and propose a new formulation of the problem using integer programming. The authors obtain better solutions for some of the benchmark instances based on real data from the University of Udine in Italy. Burke et al. (2010b) address the solution of CTTPs for the Udine instances by minimizing a linear combination of the violations of different sets of soft constraints using metaheuristics. Bonutti et al. (2012) provide a detailed review of the mathematical formulations, instances, types of information

and results found in the literature on CTTs. Other papers on this topic are [Dimopoulou and Miliotis \(2001\)](#), [Daskalaki et al. \(2004\)](#), [Lewis et al. \(2007\)](#), [Lü and Hao \(2010\)](#), [Sarin et al. \(2010\)](#), [MirHassani and Habibi \(2013\)](#), [Lü et al. \(2011\)](#), [Bellio et al. \(2012\)](#), and [Miranda et al. \(2012\)](#).

As regards ETTPs, many studies can be found in the OR literature. [Qu et al. \(2009b\)](#) defines these problems as “assigning a set of exams into a limited number of time slots (time periods) and rooms (of certain capacity) subject to a set of constraints”. In general, ETTPs must be solved in such a way that neither students nor classrooms are scheduled for more than one exam at the same time, producing conflict-free timetables, and ensuring that classroom seating capacity is not exceeded. Additional desirable (i.e., not mandatory) requirements can be modeled through terms added to an objective function that penalizes the solution if the additional requirements are not satisfied ([Lewis et al. 2007](#); [Bonutti et al. 2012](#)). These requirements vary from one institution to the other and can be incorporated into optimization models with multiple objective functions.

For ETTPs, only post-enrollment studies have been documented in the literature ([Lewis et al. 2007](#); [McCollum 2007](#); [Gogos et al. 2012](#); [McCollum et al. 2012](#)). However, many educational institutions schedule their exams before students register for individual courses and therefore base timetabling decisions on their curricula configurations. Typically, these establishments do not have enough classrooms available at times other than those at which the courses are taught and thus are forced to schedule specific examination periods each term. Since classes must be suspended, these time periods are necessarily short (7–10 days), making it harder to find an exam schedule that satisfies all the students.

Although none of the ETTP studies found in the literature address the pre-enrollment (curriculum-based) case, this type of situation is faced term after term by many higher education institutions around the world. One of these institutions is the Faculty of Engineering at the Universidad Diego Portales (UDP) in Chile, where three rounds of examinations are scheduled every term. Each round is defined by a time horizon (length of the examination period) and the number and types of classrooms that are available.

The main characteristics of the current exam scheduling process at the Faculty of Engineering at UDP (hereafter simply referred to as the Faculty) are as follows: (1) it is carried out manually by the Faculty’s Academic Secretary; (2) the resulting schedules are not known by the students when they register for courses; (3) it does not take students’ course registration into account; and (4) classroom assignments for each examination are determined once the definitive number of students registered in each course is known.

Another important aspect of the exam scheduling process is that students are allowed to register in all the courses for which they meet prerequisites and there is no schedule conflict. This complicates matters because it means that the number of registrants in each course is not known in advance and the schedules initially generated will generally have to be modified once registration is complete. This rescheduling process is a complex task since the final exam timetable must be such that every single student can take the exams for the courses he or she is registered in. As a result, it is common for some exams to be postponed or even scheduled outside of the exam period, and in some cases to be given twice on two different dates, forcing instructors to draw up two different exam papers.

With these shortcomings in the exam scheduling process, and as a starting point, we propose to solve the examination timetabling problem also in a pre-enrollment stage. This gives rise to a variant of the ETTP that we will call CB-ETTP, the ETTP based on study program curriculum configurations. The solution approach is built around an integer linear programming (ILP) model that seeks to assign a day, a time and a classroom type to the examination of each course. Both exam schedules and course schedules are provided to the

students, who are then responsible for choosing courses during the registration period that do not have exam schedule conflicts. This eliminates the need for any later exam rescheduling.

Our solution approach for the CB-ETTP uses clustering and patterns generation techniques (Respicio and Captivo 2005) to reduce the complexity of the problem and consequently the associated computation times (Causmaecker et al. 2009). More specifically, our approach facilitates the solution process by reducing the size of the problem instances. Classroom patterns are constructed to decrease the size of the solution space and allow the solutions generated by the linear relaxation of the ILP model to be less fractional.

The remainder of this article is organized as follows. Section 2 reviews the relevant literature on the examination timetabling problem. Section 3 describes the mathematical formulation of the proposed CB-ETTP. Section 4 introduces the solution approach adopted. Section 5 discusses the instances used to evaluate the effectiveness of the proposed approach. Section 6 sets out the results obtained. Finally, Sect. 7 presents our main conclusions and offers some suggestions for possible extensions and future research.

2 Literature review

Qu et al. (2009b) exhaustively review the existing literature on ETTPs. Concentrating especially on works appearing since 2004, the authors make a detailed analysis of the published research on the problem and the key trends and topics that have been raised. They divide the literature into two categories, the first of which focuses on the performance of different solution methods for benchmark instances. Among the most widely used benchmark instances are those from the University of Toronto (Carter et al. 1996), the University of Nottingham (Burke et al. 1996), the University of Melbourne (Merlot et al. 2003), the University of Udine (Bonutti et al. 2012) and the one recently introduced by the International Timetabling Competition in 2007 (ITC2007) (McCollum et al. 2008). The second category centers on finding concrete solutions to real-world ETTP applications.

There is an abundant literature that focuses on the description of real-world problems and ways of solving them. Some recent papers are: Ayob et al. (2007), McCollum (2007), Broek et al. (2009), Qu et al. (2009b), Wang et al. (2009), Al-Yakoob et al. (2010), Joshua et al. (2010), Burke et al. (2010b), Kahar and Kendall (2010), Sagir and Kamisli (2010), Thomas et al. (2010), Lach and Lübbecke (2012) and McCollum et al. (2012).

On methods for solving benchmark instances, some studies focus on the optimality gap while others concentrate on the time required to find a feasible solution. In the first group are papers using meta-heuristics such as local search (Burke et al. 2006; Abdullah et al. 2007a), tabu search (Abdullah et al. 2007b; Pais and Maral 2012), great deluge (Burke et al. 2004a), simulated annealing (Thompson and Dowsland 1996, 1998), and evolutionary algorithms (Cheong et al. 2009; Pillay and Banzhaf 2010). Other studies have been based on hyper-heuristics (Qu et al. 2009a; Pillay and Banzhaf 2009; Sabar et al. 2012b; Burke et al. 2012a; Demeester et al. 2012), multi-objective methods (Burke et al. 2001; Munford 2010; Cheong et al. 2009), hybrid approaches (Azimi 2005; Burke et al. 2010a; Turabieh and Addullah 2011; Burke et al. 2012b), integer programming (MirHassani 2006), constraint-based techniques (Huédé et al. 2006), and other approaches based on different mathematical methods (Asmuni et al. 2009; Mansour et al. 2011; Sabar et al. 2012a). In the second group are works such as Eley (2006), Petrovic et al. (2007), Pillay and Banzhaf (2010), Gogos et al. (2012) and Abdul-Rahman et al. (2014).

The articles by [Al-Yakoob et al. \(2010\)](#) and [McCollum et al. \(2012\)](#) formulate and solve the examination timetabling problem using integer linear programming models that represent the specific conditions of the particular real-world problems being addressed. The latter work also describes the implementation of examination timetabling systems in Europe, Australia and the United States. The studies by [Ayob et al. \(2007\)](#), [McCollum \(2007\)](#) and [Thomas et al. \(2010\)](#) also present the implementation of decision support systems for ETTs.

Multiple ways of evaluating the quality of the various solution approaches have been proposed in the literature. Frequently used criteria include the time required to find a feasible solution, the optimality gap and the solution approach's performance once it is implemented ([Lewis et al. 2007](#); [McCollum et al. 2012](#); [Thomas et al. 2010](#)).

In the solution approach presented in this paper we employ clustering and patterns generation techniques to reduce the complexity of the problem and thus facilitate its solution. A good example of the use of patterns is the paper mill cutting stock problem ([Aboudi and Barcia 1998](#); [Suliman 2006](#)). Patterns generation have also been employed by [Miranda et al. \(2012\)](#) for activity scheduling problems. The authors explicitly construct timetable patterns to solve the problem of creating course schedules. Clustering techniques are mentioned in [Thomas and Tajudin \(2006\)](#) as a way of eliminating potential symmetries between solutions and thereby reducing the size of the instances of the problem to be solved. Despite the advantages cited by the authors, we have not found any other work that resorts to these techniques in timetabling.

3 General description of the CB-ETTP

Given the background provided above, we considered a pre-enrollment version of the ETTP and included the allocation of rooms in the scheduling process, which is a contribution to the state-of-the-art of ETTs. In what follows, we present a general description and a solution approach for the curriculum-based examination timetabling problem. The proposed approach is based on a set of mathematical programming models and the use of classroom patterns and clustering of courses.

Consider the problem facing a post-secondary educational institution that must schedule the examinations for the set N of programs it offers. The institution has a curriculum configuration for each program $n \in N$ that determines the set of courses C_n the student must take over S_n academic terms. Since some courses may be included in more than one program, the set C of all courses to be scheduled satisfies $|C| \leq \sum_{n \in N} |C_n|$. The institution also has an estimate of the number of students that will register in each course c , which we denote by $ens_c \forall c \in C$.

The examination period, T , is represented by sets of L_t disjoint time slots making up each day $t \in T$. These time slots are of equal length, reflecting our assumption that all exams have the same duration. The total number of available slots over the entire examination period is $H = \sum_{t \in T} L_t$. The institution counts with R classrooms (hereafter called simply rooms), each with a known seating capacity determined by the number of students it can seat for an exam.

Any solution to the CB-ETTP must satisfy the following hard constraints: (1) each course exam must be assigned to a unique time slot; (2) exams for courses in the same term and program must be scheduled in different time slots; (3) each room must be assigned to no more than one course in each time slot; (4) seating capacities must not be exceeded;

(5) exams for courses taught by the same instructor must be scheduled in different time slots.

In addition, there will normally be certain preferences that are particular to the educational institution. These preferences will be modeled as soft constraints. For example: (1) exams for courses in the same term and program should not be scheduled in consecutive time slots on the same day; (2) exams for courses in consecutive terms of a given program should not be scheduled in the same time slot; (3) no program should have significantly more exam schedule conflicts than the others; (4) no term in a program should have more exam schedule conflicts than the others; (5) the rooms assigned to the exam for a given course should be as close together as possible; (6) exams for courses taught in the first term of a program should not be scheduled on the same day and ideally at least one day apart; (7) instructors' time preferences should be taken into account in the exam scheduling process; and (8) exams for courses taught by the same instructional team should be scheduled in different time slots.

The proposed solution approach for the CB-ETTP attempts to determine which time slots and rooms to assign to each course exam. The resulting schedules must satisfy all of the hard constraints while trying to minimize penalties for not satisfying the soft constraints.

If we consider what has been reported in [McCollum et al. \(2008\)](#) about the track on ETTPs in the ITC2007, we can conclude that the main differences with respect to the problem studied in this paper are: (1) we consider a pre-enrollment problem, while the ITC2007 focuses on the post-enrollment case; (2) we evaluate potential exam schedule conflicts through the specific curriculum configurations, whereas the ITC2007 does it directly on a student-by-student basis; (3) unlike the ITC2007, we also consider room allocation decisions, which in turn lead us to consider multiple academic programs as multiple programs share the same rooms.

4 Solution approach

The proposed solution approach for the CB-ETTP consists of four sequential stages as depicted in Fig. 1. The approach allows us to decrease the complexity of the problem by grouping courses into clusters, avoiding symmetric solutions, and separating exam scheduling decisions from room allocation decisions. The sequential solution of the problem implies that the output from each stage, but the last one, is the input for the stage that follows.

The first stage consists of the processing and transformation of the data that define the problem. It begins with the grouping of courses into clusters through a process to be explained in Sect. 4.1. Then, for each cluster, all combinations of rooms whose total seating capacity is greater than or equal to the estimated number of students who will register in the courses in that cluster are identified. These room combinations form a room pattern that is constructed according to the process described in Sect. 4.1.

In the second stage, an optimization model is solved which assigns time slots and room patterns to the examinations for the courses in each cluster. The model attempts to minimize the violation of the following constraints: (1) exams for the first-year courses of a given program should be scheduled at least one day apart; (2) if exams for courses taught in the same term and program are scheduled on the same day, they should not be assigned to consecutive time slots; and (3) exams for courses in consecutive terms in a given program should be scheduled in different time slots. This optimization problem is called the curriculum-based examination timetabling cluster allocation problem (CB-ETTP-C) and its solution determines

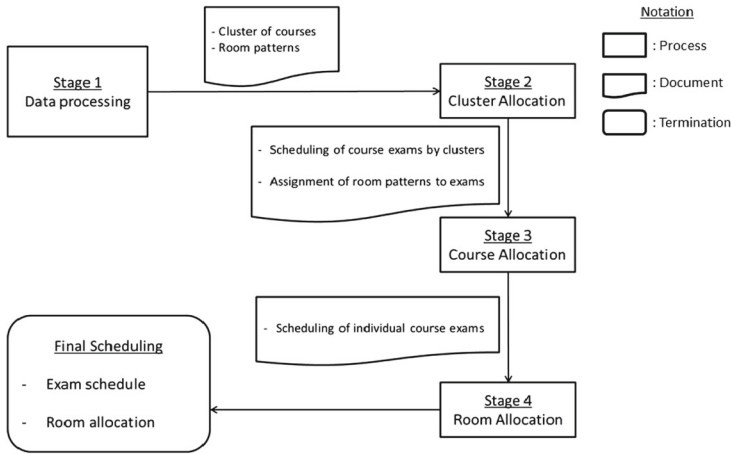


Fig. 1 Proposed hierarchical solution approach for the CB-ETTP

the time slots and number of rooms of each type (e.g., small, medium and large) to assign to each course cluster.

Because of the hard constraint forbidding an excess of students over the resulting seating capacity and the fact that the final number of course registrants is not known until the course registration process is completed, a safety capacity margin is required for room assignments. To this end, we define *seating capacity* as the number of seats in the room pattern assigned to a course exam and *room utilization* as the percentage of the seating capacity accounted for by the estimated number of students who will register in the course (Beyrouthy et al. 2010). A maximum room saturation level known as the *saturation tolerance* is then defined for room assignments. If a room assignment exceeds this preset level, the corresponding course is classified as a *saturated course assignment*. For example, consider course c with an estimated number of students $ens_c = 80$. The exam for this course is assigned to a room pattern consisting of two rooms of capacity 60 and 40 students, respectively. The seating capacity of this room pattern is 100 students, the sum of the individual room capacities, and the room utilization is then $80/100 = 0.8$. If the saturation tolerance is set at 0.75, then this would be a saturated course assignment.

In the third stage, the course exams are scheduled on the basis of the solution to the CB-ETTP-C. The resulting schedules determine the day, time slot and number of rooms of each type to be assigned to each course. We solve an integer linear programming model that simultaneously minimizes (1) the total number of saturated course assignments and (2) the number of saturated course assignments in the time slot with the greatest number of them. This second optimization problem is called the curriculum-based examination timetabling course allocation problem (CB-ETTP-CA).

Finally, in the fourth stage, the solution obtained for the CB-ETTP-CA is used to determine the final assignment of rooms to courses. The aim in this case is to minimize the maximum distance between the rooms assigned to each course. This third optimization problem is called the curriculum-based examination timetabling room allocation problem (CB-ETTP-RA) and it is solved independently for each time slot.

In the next four subsections, we describe in detail each of the stages introduced above.

4.1 Stage 1: Data processing

To reduce the complexity of the problem, courses that are indistinguishable in terms of requirements (program, term, prerequisites and estimated number of students) are first grouped into clusters. The grouping process is as follows: (1) if course $c \in C$ belongs to more than one program, it is considered to be an individual cluster; (2) if courses $c, d \in C$ satisfy each of the following conditions, they are grouped into the same cluster: (i) they are both found only in one program; (ii) they are taught in the same term and belong to the same program; and (iii) the estimated number of students to register for each course is “similar”. This is when the quotient of the estimated number of registrants by the minimum seating capacity is the same for each course. A group of courses that meets the second criterion above is called a *course cluster*. Each course can belong to only one cluster.

The clustering process not only reduces the number of decision variables in the model but also allows for symmetries between different solutions to be eliminated. The latter can be seen in the following example. Assume that two courses, denoted by c and d , belong to the same program and term, and that the expected number of registrants in each course is the same. The solution that schedules the exam for course c in time slot 1 and the exam for course d in time slot 5 would provide the same objective value as the solution that schedules the exam for course c in time slot 5 and the exam for course d in time slot 1.

The clustering process results in the definition of a set I of course clusters, where $|I| \leq |C| \leq \sum_{n \in N} |C_n|$. Each cluster $i \in I$ is described by three parameters: Q_i , Ens_i and U_i . Parameter Q_i corresponds to the number of courses that are part of cluster i . Parameter Ens_i represents the maximum expected number of students across the courses in cluster i . Thus, $Ens_i = \max_{c \in Q_i} \{ens_c\} \forall i \in I$, where ens_c is the estimate of the number of students in course c . Finally, parameter U_i represents the program curriculum information and indicates how cluster i relates to the other course clusters.

Once the courses have been grouped into clusters, we classify the available rooms into a set Z of different types according to their seating capacity. This reduces the size of the problem even more, making it easier to solve. Then, we construct *room patterns*. Pattern p is denoted by a vector $(a_1^p, a_2^p, \dots, a_{|Z|}^p)$, where a_l^p is the number of rooms of type $l = 1, \dots, |Z|$ contained in the pattern. Pattern p is *valid* for a given cluster i if $Ens_i \leq \sum_{l=1}^{|Z|} a_l^p q_l$, where q_l indicates the seating capacity of a room of type l . It is said that a valid pattern p *dominates* a valid pattern f if $a_1^p \leq a_1^f, a_2^p \leq a_2^f, \dots, a_{|Z|}^p \leq a_{|Z|}^f$ and the inequality is strictly satisfied at least in one case. Otherwise, patterns p and f would be the same. The patterns used in the mathematical formulation of the problem are those that are not dominated by any other pattern.

The use of clustering and patterns generation techniques as pre-processing activities allows us: (1) to reduce the amount of decision variables in the problem formulation, which in turn reduces the dimensionality of the model, improving solution times; and (2) to preset valid solutions for the allocation of rooms to exams, reducing solution times even more. This pre-processing approach was successfully used by Miranda et al. (2012) to schedule lecture times. Despite of their advantages, these two techniques—to the best of our knowledge—have not been used simultaneously in the literature to solve ETPP problems.

By the end of this stage, courses clusters are created, parameters Q_i , Ens_i and U_i are computed and the valid room patterns for each cluster are identified. This information is an input for Stage 2.

4.2 Stage 2: Integer programming model for scheduling clusters (CB-ETTP-C)

The various elements used in the formulation of the CB-ETTP-C are described below:

Sets

I	Set of course clusters.
H	Set of time slots.
T	Set of days (in the examination period).
F	Set of programs offered.
S	Set of program-terms.
Z	Set of room types.
P	Set of room patterns.

Indexes

i, k	Course clusters $i, k \in I$.
h	Time slot $h \in H$.
t	Day $t \in T$.
f	Program $f \in F$.
s	Program-term $s \in S$.
z	Room type $z \in Z$.
p	Room pattern $p \in P$.

Subsets

$G_i \subseteq P$	Set of valid room patterns for cluster $i \in I$.
$FS_f \subseteq I$	Set of course clusters for first term of program $f \in F$.
$SC_s \subseteq I$	Set of course clusters for program-term $s \in S$.
$E_s \subseteq I$	Set of course clusters for program-term $s + 1 \in S$ following program-term $s \in S$.
$L_t \subseteq H$	Set of available time slots on day $t \in T$.

Parameters

Q_i	Number of courses in cluster $i \in I$.
QN_{zp}	Number of rooms of type $z \in Z$ in room pattern $p \in P$.
QS_z	Number of rooms of type $z \in Z$ that are available.
$B_{h,(h+1)}$	Binary parameter that is equal to 1 if the time slot $h \in H$ and the time slot $h + 1 \in H$ belong to the same day, and 0 otherwise.

Penalties

α	Penalty for assigning consecutive time slots belonging to the same day to exams for courses in the same program-term.
β	Penalty for scheduling exams for courses in the first term of a program on consecutive days.
γ	Penalty for assigning the same time slot to exams for courses in consecutive program-terms.

Variables

x_{hip}	Binary variable equal to 1 if time slot h is assigned an exam for one of the courses in cluster i with room pattern p , and 0 otherwise.
-----------	--

- w_{sh} Binary variable equal to 1 if the exam for a course in term s is assigned to time slot h and the exam for another course in the same term s is assigned to time slot $h + 1$, where h and $h + 1$ are on the same day. Its value is 0 otherwise.
- y_{ft} Binary variable equal to 1 if exams for courses in first term of program f have been assigned to day t and day $t + 1$, and 0 otherwise.
- v_{sh} Binary variable equal to 1 if exams for courses in program-term s and program-term $s + 1$ have been assigned to time slot h , and 0 otherwise.

Having defined the necessary notation, the integer programming model used to solve the course cluster exam scheduling problem can be formulated as follows:

$$\begin{aligned}
 \text{(CB-ETTP-C) } \min \quad & \sum_{s \in S} \sum_{h \in H} \alpha w_{sh} + \sum_{f \in F} \sum_{t \in T} \beta y_{ft} \\
 & + \sum_{s \in S} \sum_{h \in H} \gamma v_{sh}
 \end{aligned} \tag{1}$$

subject to:

$$\sum_{h \in H} \sum_{p \in G_i} x_{hip} = Q_i \quad \forall i \in I. \tag{2}$$

$$\sum_{p \in G_i} x_{hip} \leq 1 \quad \forall h \in H; i \in I. \tag{3}$$

$$\sum_{i \in SC_s} \sum_{p \in G_i} x_{hip} \leq 1 \quad \forall h \in H; s \in S. \tag{4}$$

$$\begin{aligned}
 \sum_{h \in L_t} \sum_{i \in FS_f} \sum_{p \in G_i} x_{hip} &\leq 1 \\
 \forall f \in F; t \in T.
 \end{aligned} \tag{5}$$

$$\begin{aligned}
 \sum_{i \in I} \sum_{p \in G_i} Q N_{z,p} x_{hip} &\leq Q S_z \\
 \forall z \in Z; h \in H.
 \end{aligned} \tag{6}$$

$$\begin{aligned}
 \sum_{i \in SC_s} \sum_{p \in G_i} x_{hip} + \sum_{i \in SC_s} \sum_{p \in G_i} x_{(h+1)ip} &\leq 1 + w_{sh} \\
 \forall s \in S; h, h + 1 \in H : B_{h,(h+1)} &= 1.
 \end{aligned} \tag{7}$$

$$\begin{aligned}
 \sum_{h \in L_t} \sum_{i \in FS_f} \sum_{p \in G_i} x_{hip} + \sum_{h \in L_{t+1}} \sum_{i \in FS_f} \sum_{p \in G_i} x_{hip} &\leq 1 + y_{ft} \\
 \forall f \in F; t, t + 1 \in T.
 \end{aligned} \tag{8}$$

$$\begin{aligned}
 \sum_{i \in SC_s} \sum_{p \in G_i} x_{hip} + \sum_{i \in E_s} \sum_{p \in G_i} x_{hip} &\leq 1 + v_{sh} \\
 \forall s \in S; h \in H.
 \end{aligned} \tag{9}$$

$$x_{hip} \in \{0, 1\} \quad \forall h \in H; i \in I; p \in G_i. \tag{10}$$

$$w_{sh}, v_{sh} \in \{0, 1\} \quad \forall s \in S; h \in H. \tag{11}$$

$$y_{ft} \in \{0, 1\} \quad \forall f \in F; t \in T. \tag{12}$$

The CB-ETTP-C seeks to minimize the total penalty associated with the resulting exam schedule. The first term in the objective function (1) represents the total penalty for assigning

Table 1 Description of the constraints in the CB-ETTP-C

Constraint	Description
(2)	Each cluster is assigned as many time slots as the number of courses in it.
(3)	Each course exam must be scheduled in a single time slot.
(4)	Exams for courses in the same term and program must not be scheduled in the same time slot.
(5)	Exams for the first-term courses in each program must be scheduled on different days.
(6)	No more rooms of each type than those available in each time slot must be assigned.
(7)	Determines the value of variable w_{sh} .
(8)	Determines the value of variable y_{ft} .
(9)	Determines the value of variable v_{sh} .
(10)–(12)	All variables are binary.

exams for two or more courses in the same program-term to consecutive time slots on the same day. The second term corresponds to the total penalty for scheduling two or more exams for first-term courses in the same program on consecutive days. Finally, the third term is the total penalty for assigning the same time slot to exams for courses in consecutive program-terms. A brief description of constraints (2)–(12) is provided in Table 1.

The solution to the CB-ETTP-C indicates the time slots that should be assigned to the courses that belong to each cluster. This information is used as an input for Stage 3.

4.3 Stage 3: Mixed integer programming model for scheduling course examinations (CB-ETTP-CA)

The various elements used in the formulation of the CB-ETTP-CA are listed below together with a brief explanation:

Sets

- C Set of courses.
- I Set of course clusters.
- H Set of time slots.

Indexes

- c Course $c \in C$.
- i Course cluster $i \in I$.
- h Time slot $h \in H$.

Subsets

- $CI_i \subseteq C$ Set of courses in course cluster $i \in I$.

Parameters

- χ_{ih} Binary parameter that is equal to 1 if a course in cluster $i \in I$ has been assigned time slot $h \in H$, and 0 otherwise. Its value comes from the solution to the CB-ETTP-C.
- ens_c Estimate of the number of students who will register in course $c \in C$.
- $sattol$ Saturation tolerance (i.e., the pre-determined percentage of the seating capacity to be maintained as a safety margin due to the uncertainty regarding final student enrollment levels).

τ_{ch} Binary parameter that is equal to 1 if no more than $(1 - sattol)$ percent of the seating capacity is used when course $c \in C$ is assigned to time slot $h \in H$, and 0 otherwise. This parameter is determined from the seating capacity assigned to cluster $i \in I$ in time slot $h \in H$ by the CB-ETTP-C.

Variables

Q_{ch} Binary variable that is equal to 1 if the exam for course c is assigned to time slot h , and 0 otherwise.

η_{ch} Binary variable that is equal to 1 if the seating capacity assigned to a course satisfies the saturation tolerance ($sattol$), and 0 otherwise.

φ Continuous variable that helps us to determine the number of saturated course assignments in the time slot with the highest number of them.

Thus, the mixed integer programming model used to solve the course examination scheduling problem can be formulated as follows:

$$(CB-ETTP-CA) \quad \min \quad |C| \cdot \varphi + \sum_{h \in H} \sum_{c \in C} \eta_{ch} \tag{13}$$

subject to:

$$\sum_{h \in H} Q_{ch} = 1 \quad \forall c \in C. \tag{14}$$

$$\sum_{c \in C I_i} Q_{ch} = \chi_{ih} \quad \forall i \in I; h \in H. \tag{15}$$

$$\eta_{ch} = (1 - \tau_{ch}) \cdot Q_{ch} \quad \forall h \in H; c \in C. \tag{16}$$

$$\varphi \geq \sum_{c \in C} \eta_{ch} \quad \forall h \in H. \tag{17}$$

$$Q_{ch}, \eta_{ch} \in \{0, 1\} \quad \forall c \in C; h \in H. \tag{18}$$

$$\varphi \geq 0 \tag{19}$$

The CB-ETTP-CA model above seeks to reduce the number of saturated course assignments once rooms are assigned. The first term in the objective function (13) is the penalty associated with the most saturated time slot (i.e., with the slot with the greatest number of saturated course assignments). The second term is the total number of saturated course assignments. The weighting factor of $|C|$ for the first term in the objective function ensures this term is given more importance in the minimization process. A brief description of constraints (14)–(19) is provided in Table 2.

The solution to the CB-ETTP-CA indicates the time slot assigned to the exam for each course. This information is used as an input for Stage 4.

4.4 Stage 4: Mixed integer programming model for room assignment (CB-ETTP-RA)

The various elements used to formulate the CB-ETTP-RA are described below:

Sets

- C Set of courses.
- R Set of rooms.
- Z Set of room types.

Table 2 Description of the constraints in the CB-ETTP-CA

Constraint	Description
(14)	Each course exam must be assigned to one time slot
(15)	Each course exam must be assigned to a time slot allocated to the cluster to which it belongs
(16)	Determines the value of variable η_{ch}
(17)	Helps determine the value of variable φ
(18)–(19)	The type of the variables

Indexes

- c Course $c \in C$.
- r, \hat{r} Rooms $r, \hat{r} \in R$.
- z Room type $z \in Z$.

Subsets

$B_z \subseteq R$ Set of rooms of type $z \in Z$.

Parameters

- M_{cz} Number of rooms of type $z \in Z$ required by the exam for course $c \in C$. This information is provided by the solution to the CB-ETTP-C and CB-ETTP-CA.
- $D_{r\hat{r}}$ Distance from room $r \in R$ to room $\hat{r} \in R$.

Variables

- u_{cr} Binary variable that is equal to 1 if the exam for course c is assigned to room r , and 0 otherwise.
- d_c Continuous variable that allows us to determine the maximum distance between the rooms assigned to the exam for course $c \in C$.
- γ Continuous variable that helps us to determine the maximum distance among all room assignments (i.e., the maximum value taken by variable d_c across all courses).

Given that the decisions for different time slots are independent of each other, the CB-ETTP-RA is solved for each time slot separately. Thus, the mixed integer programming model used for assigning rooms to exams can be formulated as follows:

$$(CB-ETTP-RA) \quad \min \quad |C| \cdot \gamma + \sum_{c \in C} d_c \tag{20}$$

subject to:

$$\sum_{r \in B_z} u_{cr} = M_{cz} \quad \forall c \in C; z \in Z. \tag{21}$$

$$\sum_{c \in C} u_{cr} \leq 1 \quad \forall r \in R. \tag{22}$$

$$d_c \geq (u_{cr} + u_{c\hat{r}} - 1) \cdot D_{r\hat{r}} \quad \forall c \in C; r, \hat{r} \in R. \tag{23}$$

$$\gamma \geq d_c \quad \forall c \in C. \tag{24}$$

$$u_{cr} \in \{0, 1\} \quad \forall c \in C; r \in R. \tag{25}$$

$$d_c \geq 0 \quad \forall c \in C. \tag{26}$$

$$\gamma \geq 0 \tag{27}$$

Table 3 Description of the constraints in the CB-ETTP-RA

Constraint	Description
(21)	Each exam must be assigned the number of rooms of each type determined by the solution to the CB-ETTP-C and CB-ETTP-CA
(22)	Each room must be assigned to no more than one exam
(23)	Helps determine the value of variable d_c
(24)	Helps specify the value of variable γ
(25)–(27)	The type of the variables

The CB-ETTP-RA seeks to minimize the distance between the rooms assigned to the various exams. The first term in the objective function (20) represents the penalty associated with the distance between the rooms assigned to the exam whose rooms are the most distant from each other. The second term represents the total penalty associated with the maximum distance between the rooms assigned to the different courses. The weighting factor $|C|$ in the first term, which is just the total number of courses being considered, ensures the first term is given more importance when the objective function is minimized. A brief description of constraints (21)–(27) is provided in Table 3.

The solution to the CB-ETTP-RA is a room-to-exam assignment. Thus, once the four stages of the proposed solution approach are sequentially completed, the exams for all the courses are scheduled and appropriate groups of rooms are already assigned.

5 Benchmark instances

To evaluate the performance of the proposed solution approach, we consider seven real-world instances of the CB-ETTP corresponding to seven different academic terms at the Faculty of Engineering at the Universidad Diego Portales (UDP), Chile. These instances correspond to the second term of 2005, the first and second term of 2006, the first term of 2007 and the first term of 2013, 2014 and 2015. We will refer to them as udp -(year-term), so the instance associated with the second term of 2005, for example, will be called udp -(2005-02). We chose these instances because the information used by the Faculty to manually schedule exams in each case was available to the authors. It is important to note that in Chile the academic year has two semesters. All these instances are available at <http://orincancercare.org/asaure/datasets>.

Between 2005 and 2007, the Faculty offered four programs, 12 terms each. Three rounds of exams were held in each term, the first two for midterms and the third for finals. Each round consisted of 7 days, beginning on a Saturday and ending the following Saturday but excluding Sunday. The exams were scheduled between 8 AM and 6 PM on weekdays (4 time slots per day) and between 8 AM and 2 PM on Saturdays (2 time slots per day). Each time slot was 3 hours long and each round consisted of 24 time slots.

From 2005 to 2007, the Faculty had 33 rooms available for course exams, classified into three types: 12 small, 14 medium and 7 large. A small room could seat 30 students, a medium room 55 and a large room 90. This classification was obtained by grouping rooms of similar size. The seating capacity of each room type was then defined as the seating capacity of the smallest room in each group.

Table 4 Number of time slots and rooms available for each instance

Instance	Available time slots	Days	Small rooms	Medium rooms	Large rooms
udp-(2005-02)	24	7	12	14	7
udp-(2006-01)	24	7	12	14	7
udp-(2006-02)	24	7	12	14	7
udp-(2007-01)	24	7	12	14	7
udp-(2013-01)	25	5	16	15	6
udp-(2014-01)	25	5	16	15	6
udp-(2015-01)	25	5	16	15	6

Table 5 Characteristics of the evaluation instances

Instance	Number of exams	Estimated no. of students	Number of programs	Number of terms	Number of program-terms	Number of conflicts	<i>CD</i>	<i>CP</i>
udp-(2005-02)	166	1153	4	12	48	3554	0.16	0.68
udp-(2006-01)	172	1213	4	12	48	3891	0.18	0.73
udp-(2006-02)	160	1189	4	12	48	3651	0.15	0.64
udp-(2007-01)	177	1256	4	12	48	4151	0.19	0.76
udp-(2013-01)	132	1521	3	12	36	2044	0.12	0.47
udp-(2014-01)	115	1416	3	12	36	2029	0.15	0.52
udp-(2015-01)	126	1537	3	10	30	2341	0.15	0.53

Between 2013 and 2015, the Faculty offered three programs. From 2013 to 2014, the duration of these programs was 12 terms, while in 2015 the duration of all of them was reduced to 10 terms. Three rounds of exams were held in each term, the first two for midterms and the third for finals. Each round consisted of 5 days, beginning on a Monday and ending on a Friday. The exams were scheduled between 8 AM and 6 PM. Each time slot was 2 hours long and each round consisted of 25 time slots.

From 2013 to 2015, the Faculty had 37 rooms available for course exams, classified into three types: 16 small, 15 medium and 6 large. A small room could seat 30 students, a medium room 55 and a large room 90.

The relevant characteristics of each instance are summarized in the first five columns of Table 5. Information about the number of time slots and rooms available is presented in Table 4.

To characterize the complexity of each instance we consider the conflict density (*CD*) measure used by Pillay and Banzhaf (2009), Qu et al. (2009a), Wang et al. (2009), Pillay and Banzhaf (2010) and Gogos et al. (2012). This measure is calculated as the ratio of the actual number of schedule conflicts to the maximum possible number of conflicts that could arise. In formal terms,

$$CD = \frac{\sum_{c \in C} \sum_{\bar{c} \in C: \bar{c} \neq c} \varepsilon_{c\bar{c}}}{PT}$$

where $\varepsilon_{c\bar{c}}$ is a binary indicator that is equal to 1 if the exams for courses *c* and \bar{c} have conflicting exam times when they are scheduled in the same time slot, and 0 otherwise. *PT*

Table 6 Results from data processing and execution times (Stage 1)

Instance	Number of course clusters	Number of room patterns	Processing time (s)	Number of x_{hip} variables	Number of constraints
udp(2005-02)	131	305	87	958,920	13,771
udp(2006-01)	122	345	92	1,010,160	13,546
udp(2006-02)	111	289	76	769,896	13,271
udp(2007-01)	119	312	88	891,072	13,471
udp(2013-01)	56	224	42	313,600	3868
udp(2014-01)	55	196	36	296,500	3992
udp(2015-01)	54	298	43	402,300	3816

is the maximum possible number of conflicts given the number of exams to be scheduled. That is $PT = |C|^2 - |C|$.

The CD values for each instance are displayed in Table 5. For 2005 to 2007, the average conflict density turned out to be 0.17. For 2013 to 2015, the average conflict density was 0.14.

Also shown in Table 5 for each instance is the probability of having at least one conflict, denote by CP . This measure, suggested by Wang et al. (2009), is calculated using the following formula:

$$CP \approx 1 - \left(1 - \frac{1}{|H|}\right)^{CD \times |C|}$$

where CD is the conflict density, $|C|$ is the number of exams to be scheduled and $|H|$ is the total number of time slots available.

The two complexity measures described above provide a good indication of the difficulty of a particular instance. As the number of conflicting relationships approaches zero, the value of CD gets smaller and solving a particular instance becomes easier. This is also reflected in the value of CP , which gets closer to zero indicating that the probability of having a conflict is lower. As the number of conflicting relationships increases, the value of CD approaches the value of 1. The complexity of the problem in such a case will depend on the total number of time slots available. The greater the availability, the lower the probability of having scheduling conflicts. In other words, when the value of CD is close to one, the value of CP will depend on $|H|$. The larger the value of $|H|$, the lower the probability of having a conflict.

6 Results

The different integer programming models were implemented in GAMS 23.8 with CPLEX 12 as the solver. The computer used to run the algorithm was a 2.4 GHz Intel Core i7-3630QM PC with 6GB of RAM. The same computer was used for all the instances.

Table 6 shows the number of course clusters and room patterns generated for each instance, together with the corresponding execution times. Also in this table is the number of variables and constraints associated with the resulting CB-ETTP-C model for each instance (the result

from Stage 1). The number of variables and constraints provides a clear indication of the computational complexity of the problem in each case and a justification for resorting to a hierarchical solution approach that separates the problem into the assignment of time slots to course clusters and the assignment of rooms to courses.

The results for each CB-ETTP-C model are shown in Table 7. The first three columns of this table show the number of cases in which first-year course exams were scheduled on consecutive days (number of Type 1 conflicts), the number of exams for courses in the same program and term that were scheduled in consecutive time slots (number of Type 2 conflicts), and the number of cases in which exams for courses in the same program and consecutive terms were scheduled on the same day and time slot (number of Type 3 conflicts). For all seven instances, a Type 1 conflict was penalized 10 times more than a Type 2 conflict and 20 times more than a Type 3 conflict. The penalty values, although arbitrary, were defined directly by the person in charge of the manual scheduling process to represent the importance given by the Faculty to each type of conflict.

The fourth column in Table 7 corresponds to the *room utilization* rate, which is defined as the average over the percentage of available rooms assigned to exams in each time slot. We can see that the room utilization rate varies between 36 and 72 % for the instances being considered. These values suggest an excessive number of rooms available in each time slot. However, there was at least one conflict and one time slot with full room assignment for each of the first four instances. This suggests that increasing the room utilization rate may, in some cases, increase the number of conflicts.

The results for the different CB-ETTP-CA instances, based on those obtained for the CB-ETTP-C instances, are summarized in Table 8. The value for the saturation tolerance parameter was set at 10 % for all instances, meaning that saturated course assignments were defined as those for which exams required 90 % or more of the number of seats assigned. The third column in Table 8 corresponds to the number of saturated course assignments. The values range from 26 to 66, representing between 20.63 and 37.50 % of the number of courses being considered. The number of saturated course assignments in the time slot with the highest number of them (value of φ in Table 8) ranged from 3 to 7.

The main assignment criterion used to solve the CB-ETTP-RA instances was the distance between the rooms assigned to the different courses. Room closeness was measured on a scale defined in collaboration with the person in charge of the manual scheduling process as follows: value of 1 for rooms on the same floor of the same building; value of 2 for rooms on different floors of the same building; value of 3 for rooms in different buildings of the same faculty; and value of 4 for rooms in different faculties.

The results obtained for the CB-ETTP-RA instances are summarized in Table 9. As noted earlier, this model is solved separately for each time slot. This means that close to 25 independent problems were solved for each instance. The second column in Table 9, γ_{MAX} , shows the maximum value of γ (maximum distance among all room assignments) for each instance. From the results, we can see that the proposed approach succeeded in ensuring that all the rooms assigned to the different courses were located in the same faculty.

The total execution times, including data processing, model construction and solution times, were of no more than 30 minutes for each instance. The time required to construct

Table 7 Results and execution times for the different CB-ETTP-C instances (Stage 2)

Instance	Number of Type 1 conflicts	Number of Type 2 conflicts	Number of Type 3 conflicts	Room utilization rate (%)	Number of time slots with full room assignment	Solution time (s)	Model construction time (s)
udp(2005-02)	0	4	1	59.12	1	44	373
udp(2006-01)	1	4	2	68.11	3	98	401
udp(2006-02)	0	4	1	65.09	2	55	398
udp(2007-01)	0	4	3	71.78	4	128	499
udp(2013-01)	2	9	0	40.22	0	60	61
udp(2014-01)	2	7	0	36.00	0	53	52
udp(2015-01)	2	15	2	38.92	0	69	69

Table 8 Results and execution times for the different CB-ETTP-CA instances with saturation tolerance parameter $sattol = 10\%$ (Stage 3)

Instance	Number of exams	No. of saturated course assignments	% of saturated course assignments	φ	Solution time (s)	Model construction time (s)
udp(2005-02)	166	54	32.53 %	5	0.09	0.06
udp(2006-01)	172	61	35.47 %	6	0.09	0.06
udp(2006-02)	160	60	37.50 %	5	0.09	0.07
udp(2007-01)	177	66	37.29 %	7	0.08	0.08
udp(2013-01)	132	31	23.48 %	3	0.08	0.08
udp(2014-01)	115	31	26.96 %	3	0.13	0.11
udp(2015-01)	126	26	20.63 %	3	0.08	0.08

Table 9 Results and execution times for the different CB-ETTP-RA instances (Stage 4)

Instance	γ_{MAX}	Average course-room closeness value ($\sum d_c/ C $)	Solution time (s)	Model construction time (s)
udp(2005-02)	2	1.74	0.47	1.91
udp(2006-01)	2	1.76	0.48	1.92
udp(2006-02)	2	1.81	0.50	1.96
udp(2007-01)	3	2.11	0.75	2.18
udp(2013-01)	3	0.90	1.00	1.13
udp(2014-01)	3	0.96	0.89	0.77
udp(2015-01)	3	0.90	1.08	0.95

the different instances is displayed in Table 6, while model construction and solution times for the three models in the proposed CB-ETTP solution approach are shown in Tables 7, 8 and 9.

Table 10 compares the exam schedules obtained through the proposed solution approach, denoted by PA, with those generated by the Faculty using a manual scheduling method, denoted by MM. The results show how the proposed solution approach clearly outperforms the manual method. Three observations follow. First, under the manual method significant rescheduling is required for each instance, while under the proposed solution approach no rescheduling is needed. The most common reasons for rescheduling under the manual method, once the definitive course enrollments were known, were: (1) having insufficient capacity assigned and (2) having at least one student with two exams the same day and time slot. Second, the number of conflicts of each type under the proposed approach is the same or much lower for all the instances. Finally, the proposed solution approach outperforms the manual method with regards to seating capacity utilization. It uses a lower percentage of seats, thus leaving a greater safety margin in case course registrations turn out to be significantly higher than estimated.

Table 10 Comparison of the results obtained using a manual method (MM) and the proposed solution approach (PA)

Instance	No. of Type 1 conflicts		No. of Type 2 conflicts		No. of Type 3 conflicts		Room utilization rate (%)		Seating capacity utilization rate (%)		Number of reschedulings	
	MM	PA	MM	PA	MM	PA	MM	PA	MM	PA	MM	PA
udp(2005-02)	10	0	4	4	18	1	66.12	59.11	91.42	86.10	4	0
udp(2006-01)	6	1	4	4	14	2	61.41	68.18	89.66	83.31	5	0
udp(2006-02)	5	0	4	4	9	1	68.43	65.55	87.09	83.98	2	0
udp(2007-01)	6	0	4	4	11	3	70.23	71.98	91.90	85.54	4	0
udp(2013-01)	3	2	17	9	15	0	36.72	40.22	68.00	59.33	8	0
udp(2014-01)	3	2	17	7	14	0	42.37	36.00	67.19	57.93	13	0
udp(2015-01)	3	2	25	15	16	2	39.35	38.92	70.13	56.23	13	0

7 Conclusions

In this paper, we describe and solve a variant of the examination timetabling problem (ETTP) that uses the curriculum configuration of different academic programs (i.e., the list of courses students should normally take each term to complete the different programs) to evaluate possible conflicts in the resulting exam schedules. We refer to this variant of the ETTP as curriculum-based examination timetabling problem (CB-ETTP). The ETTPs found in the literature assume that the number of students registered in each course is known at the time of scheduling exams. In the CB-ETTP, these numbers are uncertain and the definition and quantity of exam schedule conflicts must be determined mainly on the basis of the curriculum configurations. In addition, we consider room assignment as an essential part of the scheduling process.

The proposed solution approach for the CB-ETTP consists of three mathematical programming models that are solved sequentially. The first model assigns time slots and room patterns to course clusters. The use of room patterns and course clusters seeks to eliminate possible symmetries and reduce the size of the problem, simplifying the solution process. The second model assigns time slots and room patterns to individual courses based on the solution to the first model. Finally, the third model takes into account the solution to the second model and assigns specific rooms to each course exam.

Some of the advantages of the proposed solution approach are: (1) it simultaneously incorporates decisions on room assignment and exam scheduling for multiple programs; (2) it avoids symmetry problems in the solution process via the use of course clusters; (3) it significantly simplifies the solution of the course clusters assignment problem (CB-ETTP-C) through the use of room patterns, as the assignment of specific rooms would either require excessive execution times or simply not be solvable; and (4) it deals with the uncertainty regarding student enrollment levels by incorporating a safety margin for seating capacity.

The proposed solution approach was applied to a series of real-world timetabling instances provided by the Universidad Diego Portales, Chile. For every instance, the results obtained were satisfactory in terms of both solution quality and execution times. Specifically, the resulting schedules were better than those generated by the University's manual method because they did not require rescheduling. The number of schedule conflicts obtained through the proposed solution approach was much lower, in most cases, than that resulting from the manual method. Type 3 conflicts, for example, were reduced in at least 75%. In addition, the seating capacity utilization rate decreased on average from 80.77 to 73.20%. In terms of execution times, all the instances were solved in less than 30 minutes. This is including data processing, model construction and solution times.

One of the central assumptions of the proposed solution approach is that reasonably good estimates of course enrollment levels are available. If the levels could be described in probabilistic terms, an alternative approach would be to use stochastic programming.

At many universities, students register in specific courses based on course prerequisites (i.e., curriculum configuration), course timetables and exam schedules. Formulating and solving the combined course and exam timetabling problem would ensure better coordination so students registration options are more likely to be free of conflicts.

With respect to future research, a topic of particular interest is the case of exams that require two or more consecutive time slots and/or must be taken in two or more stages. In such situations, minimum and maximum time periods between consecutive stages would have to be incorporated.

Acknowledgements This research was partially supported by the Millennium Institute Complex Engineering Systems (ISCI), Chile [Grants ICM-FIC: P05-004-F, CONICYT: FB0816]. The authors would like to thank Claudio Gutiérrez, Camila Ramos, Catalina Parga, José Tomás Marquínez, Daniel Herl and Daniel Zúñiga for their assistance during the course of this research project.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

- Abdullah, S., Ahmadi, S., Burke, E., & Dror, M. (2007a). Investigating Ahuja–Orlin’s large neighbourhood search approach for examination timetabling. *OR Spectrum*, 29, 351–372.
- Abdullah, S., Ahmadi, S., Burke, E., Dror, M., & McCollum, B. (2007b). A tabu based large neighbourhood search methodology for the capacitated examination timetabling problem. *Journal of Operational Research*, 58, 1494–1502.
- Abdul-Rahman, S., Burke, E., Bargiela, A., McCollum, B., & Özcan, E. (2014). A constructive approach to examination timetabling based on adaptive decomposition and ordering. *Annals of Operations Research*, 218, 3–21.
- Aboudi, R., & Barcia, P. (1998). Determining cutting stock patterns when defects are present. *Annals of Operations Research*, 82, 343–354.
- Al-Yakoob, A., Sherali, H., & Al-Jazzaf, M. (2010). A mixed-integer mathematical modeling approach to exam timetabling. *Computer Management Science*, 7, 19–46.
- Asmuni, H., Burke, E., Garibaldi, J., McCollum, B., & Parkes, A. (2009). An investigation of fuzzy multiple heuristic orderings in the construction of university examination timetables. *Computer and Operations Research*, 36, 981–1001.
- Ayob, M., Malik, A., Abdullah, S., Razak Hamdam, A., Kendall, G., & Qu, R. (2007). Solving a practical examination timetabling problem: A case study. *Lecture Notes in Computer Science*, 4707, 611–624.
- Azimi, Z. N. (2005). Hybrid heuristics for examination timetabling problem. *Applied Mathematics and Computation*, 163, 705–733.
- Bellio, R., Di Gaspero, L., & Schaerf, A. (2012). Design and statistical analysis of a hybrid local search algorithm for course timetabling. *Journal of Scheduling*, 15, 49–61.
- Beyrouthy, C., Burke, E., McCollum, B., McMullan, P., & Parkes, A. (2010). University space planning and space-type profiles. *Journal of Scheduling*, 13(4), 363–374.
- Birbas, T., Daskalaki, S., & Housos, E. (2009). School Timetabling for quality student and teacher schedules. *Journal of Scheduling*, 12, 177–197.
- Bonutti, A., De Cesco, F., Di Gaspero, L., & Schaerf, A. (2012). Benchmarking curriculum-based course timetabling: Formulations, data formats, instances, validation, visualization, and results. *Annals of Operations Research*, 194, 59–70.
- Burke, E., Bykov, Y., Newall, J., & Petrovic, S. (2004a). A time predefined local search approach to exam timetabling problems. *IIE Transactions*, 36, 509–528.
- Burke, E., Bykov, Y., & Petrovic, S. (2001). A multicriteria approach to examination timetabling. *Lecture Notes in Computer Science*, 2079, 118–131.
- Burke, E., De Caemaeker, P., Vanden Berghe, G., & Van Landeghem, H. (2004b). The state of the art of nurse rostering. *Journal of Scheduling*, 7, 441–499.
- Burke, E., Eckersley, A., McCollum, B., Petrovic, S., & Qu, R. (2010a). Hybrid variable neighbourhood approaches to university exam timetabling. *European Journal of Operational Research*, 206, 46–53.
- Burke, E., Kendall, G., Misir, M., & Özcan, E. (2012). Monte carlo hyper-heuristics for examination timetabling. *Annals of Operations Research*, 2012, 73–90.
- Burke, E., Li, J., & Qu, R. (2009). A Pareto-based search methodology for multi-objective nurse scheduling. *Annals of Operations Research*, doi:10.1007/s10479-009-0590-8.
- Burke, E., Maracek, J., Parkes, A., & Rudová, H. (2010b). Decomposition, reformulation, and diving in university course timetabling. *Computers and Operations Research*, 37(3), 582–597.
- Burke, E., Newall, J., & Weare, R. (1996). A memetic algorithm for university exam timetabling. *Lecture Notes in Computer Science*, 1153, 241–250.
- Burke, E., Petrovic, S., & Qu, R. (2006). Case-based heuristic selection for timetabling problems. *Journal of Scheduling*, 9, 115–132.

- Burke, E., Pham, N., Qu, R., & Yellen, J. (2012b). Linear combination of heuristics for examination timetabling. *Annals of Operations Research*, *194*, 89–109.
- Carter, M., Laporte, G., & Lee, S. (1996). Examination timetabling: Algorithmic strategies and applications. *Journal of Operational Research Society*, *47*(3), 373–383.
- Cheong, C., Tan, K., & Veeravalli, B. (2009). A multi-objective evolutionary algorithm for examination timetabling. *Journal of Scheduling*, *12*, 121–146.
- Daskalaki, S., Birbas, T., & Housos, E. (2004). An integer programming formulation for a case study in university timetabling. *European Journal of Operational Research*, *153*, 117–135.
- De Causmaecker, P., Demeester, P., & Vanden Berghe, G. (2009). A decomposed metaheuristic approach for a real-world university timetabling problem. *European Journal of Operational Research*, *195*, 307–318.
- De Leone, R., Festa, P., & Marchitto, E. (2011). A bus driver scheduling problem: A new mathematical model and a GRASP approximate solution. *Journal of Heuristics*, *17*, 441–466.
- De Werra, D. (1985). An introduction to timetabling. *European Journal of Operational Research*, *19*, 151–162.
- Demeester, P., Bilgin, B., De Causmaecker, P., & Vanden Berghe, G. (2012). A hyperheuristic approach to examination timetabling problems: Benchmarks and a new problem from practice. *Journal of Scheduling*, *15*, 83–103.
- Derigs, U., & Friederichs, S. (2013). Air cargo scheduling: Integrated models and solution procedures. *OR Spectrum*, *35*(2), 325–362.
- Dimopoulou, M., & Miliotis, P. (2001). Implementation of a university course and examination timetabling system. *European Journal of Operational Research*, *130*, 202–213.
- Duran, G., Guajardo, M., Miranda, J., Sauré, D., Souyris, S., Weintraub, A., et al. (2007). Scheduling the Chilean soccer league by integer programming. *Interfaces*, *7*(6), 539–552.
- Eley, M. (2006). Some experiments with ant colony algorithms for the exam timetabling. *Lecture Notes in Computer Science*, *4150*, 492–499.
- Gendreau, M., Ferland, J., Gendron, B., Hail, N., Jaumard, B., Lapierre, S., et al. (2007). Physician scheduling in emergency rooms. *Lecture Notes in Computer Science*, *3867*, 53–66.
- Gogos, G., Alefragis, P., & Housos, E. (2012). An improved multi-staged algorithmic process for the solution of the examination timetabling problem. *Annals of Operations Research*, *194*, 203–221.
- Goossens, D., & Spijksma, F. (2012). Soccer schedules in Europe: An overview. *Journal of Scheduling*, *15*, 641–651.
- Gupta, D., & Denton, B. (2008). Appointment scheduling in health care: Challenges and opportunities. *IIE Transactions*, *40*(9), 800–819.
- Jat, S., & Yang, S. (2011). A hybrid genetic algorithm and tabu search approach for post enrolment course timetabling. *Journal of Scheduling*, *14*, 617–637.
- Joshua, J., & Tajudin, A. (2006). Visualizing the examination timetabling data using clustering method and TreeMaps. In *Proceedings of the 2nd IMT-GT regional conference on mathematics, statistics and applications*.
- Joshua, J., Tajudin, A., Bahari, B., & Leow, A. (2010). Exploration of rough sets analysis in real-world examination. *Lecture Notes in Computer Science*, *6729*, 173–182.
- Kahar, M., & Kendall, G. (2010). The examination timetabling problem at Universiti Malaysia Pahang: Comparison of a constructive heuristic with an existing software solution. *European Journal of Operational Research*, *207*, 557–565.
- Lach, G., & Lübbecke, M. (2012). Curriculum based course timetabling: New solutions to Udine benchmark instances. *Annals of Operations Research*, *194*, 255–272.
- Le Huédé, F., Grabisch, M., Labreuche, C., & Savéant, P. (2006). MCS—A new algorithm for multicriteria optimisation in constraint programming. *Annals of Operations Research*, *147*, 143–174.
- Lewis, R., Paechter, B., & McCollum, B. (2007). Post enrolment based course timetabling: A description of the problem model used for track two of the second international timetabling competition. Cardiff Accounting and Finance Working Papers A2007/3, Cardiff University, Cardiff Business School, Accounting and Finance Section.
- Lezaun, M., Pérez, G., & Sáinz de la Maza, E. (2007). Rostering in a rail passenger carrier. *Journal of Scheduling*, *10*, 245–254.
- Lü, Z., & Hao, J. (2010). Adaptive tabu search for course timetabling. *European Journal of Operational Research*, *200*, 235–244.
- Lü, Z., Hao, J., & Glover, F. (2011). Neighborhood analysis: A case study on curriculum-based course timetabling. *Journal of Heuristics*, *17*, 97–118.
- Maenhout, B., & Vanhoucke, M. (2010). Branching strategies in a branch-and-price approach for a multiple objective nurse scheduling problem. *Journal of Scheduling*, *13*, 77–93.
- Mansour, N., Isahakian, V., & Ghalayini, I. (2011). Scatter search technique for exam timetabling. *Applied Intelligence*, *34*, 299–310.

- McCollum, B., McMullan, P., Burke, E., Parkes, A., & Qu, R. (2008). *The second international timetabling competition: Examination timetabling track*. N Ireland: Technical Report Queen's Belfast University.
- McCollum, B. (2007). A perspective on bridging the gap between research and practice in university timetabling. *Lecture Notes in Computer Science*, 3867, 3–23.
- McCollum, B., McMullan, P., Parkes, A., Burke, E., & Qu, R. (2012). A new model for automated examination timetabling. *Annals of Operations Research*, 194, 291–315.
- Medard, C., & Sawhney, N. (2007). Airline crew scheduling from planning to operations. *European Journal of Operational Research*, 183, 1013–1027.
- Merlot, L., Boland, N., Hughes, B., & Stuckey, P. (2003). A hybrid algorithm for the examination timetabling problem. In E. K. Burke & P. De Causmaecker (Eds.) *Lecture notes in computer science: Vol 2740 Practice and theory of automated timetabling IV: selected papers from the 4th international conference* (pp. 207–231). Berlin: Springer.
- Miranda, J., Rey, P., & Robles, J. (2012). UdpSkeduler: A web architecture based decision support system for course and classroom scheduling. *Decision Support Systems*, 52, 505–513.
- MirHassani, S. (2006). Improving paper spread in examination timetables using integer programming. *Applied Mathematics and Computation*, 179, 702–706.
- MirHassani, S., & Habibi, F. (2013). Solution approaches to the course timetabling problem. *Artificial Intelligence Review*, 39(2), 133–149.
- Munford, C. (2010). A multiobjective framework for heavily constrained examination timetabling problems. *Annals of Operations Research*, 180(1), 3–31.
- Pais, T., & Maral, P. (2012). Managing the tabu list length using a fuzzy inference system: An application to examination timetabling. *Annals of Operations Research*, 194, 341–363.
- Patrick, J., Puterman, M., & Queyranne, M. (2008). Dynamic multipriority patient scheduling for a diagnostic resource. *Operations Research*, 56(6), 1507–1525.
- Petrovic, S., Yang, Y., & Dror, M. (2007). Case-based selection of initialisation heuristics for metaheuristic examination timetabling. *European Journal of Operational Research*, 33, 772–785.
- Pillay, N., & Banzhaf, W. (2009). A study of heuristic combinations for hyper-heuristic systems for the uncapacitated examination timetabling problem. *European Journal of Operational Research*, 197, 482–491.
- Pillay, N., & Banzhaf, W. (2010). An informed genetic algorithm for the examination timetabling problem. *Applied Soft Computing*, 10, 457–467.
- Qu, R., Burke, E., & McCollum, B. (2009a). Adaptive automated construction of hybrid heuristics for exam timetabling and graph colouring problems. *European Journal of Operational Research*, 198, 392–404.
- Qu, R., Burke, E., McCollum, B., Merlot, L., & Lee, S. (2009b). A survey of search methodologies and automated system development for examination timetabling. *Journal of Scheduling*, 12(1), 55–89.
- Respicio, A., & Captivo, M. (2005). *Metaheuristics: Progress as real problem solvers*. Operations Research/Computer Science Interfaces Series, Springer, chap Bi-Objective Sequencing of Cutting Patterns (Vol. 32, pp. 227–241).
- Rudová, H., Müller, T., & Murray, K. (2011). Complex university course timetabling. *Journal of Scheduling*, 14, 187–207.
- Sabar, N., Ayov, M., Qu, R., & Kendall, G. (2012b). A graph coloring constructive hyper-heuristic for examination timetabling problems. *Applied Intelligence*, 37, 1–11.
- Sabar, N., Kendall, G. M., & Qu, R. (2012a). A honey-bee mating optimization algorithm for educational timetabling problems. *European Journal of Operational Research*, 216, 533–543.
- Sagir, M., & Kamisli, Z. (2010). Exam scheduling: Mathematical modeling and parameter estimation with the analytic network process approach. *Mathematical and Computer Modelling*, 52, 930–941.
- Sarin, S., Wang, Y., & Varadarajan, A. (2010). A university timetabling problem and its solution using Benders' partitioning—A case study. *Journal of Scheduling*, 13, 131–141.
- Sauré, A., Patrick, J., Tyldesley, S., & Puterman, M. (2012). Dynamic multi-appointment patient scheduling for radiation therapy. *European Journal of Operational Research*, 223(2), 573–584.
- Schaerf, A. (1999). A survey of automated timetabling. *Artificial Intelligence Review*, 13(2), 87–127.
- Schöbel, A. (2012). Line planning in public transportation: Models and methods. *OR Spectrum*, 34, 491–510.
- Suliman, S. (2006). Pattern generating procedure for the cutting stock problem. *International Journal of Production Economics*, 74, 293–301.
- Thomas, J., & Tajudin, A. (2006). Visualizing the examination timetabling data using clustering method and TreeMaps. In *Proceedings of the 2nd IMT-GT regional conference on mathematics, statistics and applications*.
- Thomas, J., Tajudin, A., & Belaton, B. (2010). Information visualization approach on the university examination timetabling problem. In M. L. Huang, Q. V. Nguyen & K. Zhang (Eds.), *Visual information communication* (pp. 255–264). US: Springer.

- Thompson, J., & Dowsland, K. (1996). Variants of simulated annealing for the examination timetabling problem. *Annals of Operations Research*, *63*, 105–128.
- Thompson, J., & Dowsland, K. (1998). A robust simulated annealing based examination timetabling system. *Computers and Operations Research*, *25*, 637–648.
- Turabieh, H., & Addullah, S. (2011). An integrated hybrid approach to the examination timetabling problem. *Omega*, *39*, 598–607.
- Van den Broek, J., Hurkens, C., & Woeginger, G. (2009). Timetabling problems at the TU Eindhoven. *European Journal of Operational Research*, *196*, 877–885.
- Wang, S., Bussieck, M., Guignard, M., Meeraus, A., & O'Brien, F. (2009). Term-end exam scheduling at United States military academy/west point. *Journal of Scheduling*, *13*, 375–391.
- Wren, A. (1996). Scheduling, timetabling and rostering—A special relationship. In E. Burke, P. Ross (Eds), *Practice and Theory of Automated Timetabling*, Lecture Notes in Computer Science (Vol. 1153, pp. 46–75).