Annals of Operations Research manuscript No.
(will be inserted by the editor)

University timetabling with heterogeneous lectures and compactness constraints

Luis M. Torres · Ramiro Torres

Received: date / Accepted: date

Abstract A variant of the Curriculum-Based Course Timetabling (CB-CTT) Problem arising from a practical application is studied. This variant includes two key features: lectures with heterogeneous durations and a compactness soft constraint for curricula. Each course lecture has a specific duration, indicating a number of consecutive time periods required for the lecture to take place. Moreover, each day in the schedule of any curriculum should not exceed a prescribed maximum length, where the length of a working day is defined as the time elapsed between the start of the first lecture until the end of the last lecture on that day. Finally, some soft constraints of the standard CB-CTT, such as minimum number of working days or room capacity, are considered as hard constraints in the variant studied here.

An integer programming model and a two-stage solution algorithm are proposed for this problem. At first, a reduced model obtained by discarding schedule compactness constraints is solved to optimality. The solution is then extended to an initial feasible solution of the original model, which is then improved using a standard branch-and-cut procedure. Three classes of symmetry-breaking inequalities are described for this integer program and tested on real-world instances.

Keywords curriculum-based course timetabling · integer programming · compact timetables · symmetry-breaking inequalities

This research was funded by grant PIJ-16-06 from Escuela Politécnica Nacional.
Preliminary results of this paper were presented at PATAT 2018.

Luis M. Torres
Centro de Modelización Matemática – ModeMat, Departamento de Matemática
Escuela Politécnica Nacional, Quito
Tel.: +593-2-2976300
E-mail: luis.torres@epn.edu.ec
ORCID iD: 0000-0002-0778-195X

Ramiro Torres
Centro de Modelización Matemática – ModeMat, Departamento de Matemática
Escuela Politécnica Nacional, Quito
Tel.: +593-2-2976300
E-mail: ramiro.torres@epn.edu.ec

1 Introduction

Automated timetabling is an important and very active research topic within the fields of combinatorial optimization and operations research. In a broad sense, timetabling can be considered as the problem of allocating given resources to time slots and places, with the aim of satisfying a given set of objectives to the highest possible extent, while taking into account a given set of constraints (Wren, 1996). Besides of educational institutions, timetabling problems appear in such diverse contexts as health care institutions, transportation, sports, employee management, etc.

Course timetabling plays a key role within the academic planning at higher education institutions. Roughly speaking, the University Course Timetabling Problem consists in assigning time slots and suitable classrooms or laboratories for each of the weekly lectures scheduled by an academic unit in a given term. This assignment must fulfill a set of *hard constraints*, such as the availability of lecturers and classrooms, or the prevention of schedule clashes for lecturers. Additionally, the assignment should also minimize an objective function accounting for the weighted sum of penalties associated with the violation of some *soft constraints*, such as schedule preferences for lecturers or schedule compactness for groups of students. In general, both hard and soft constraints may vary strongly from one institution to another, as they depend strongly on their internal organization.

Four International Timetabling Competitions (ITCs) have been organized with the aim of fostering the research in the field of educational timetabling, providing common ground for cross-fertilization of ideas among research groups, and bridging the gap between theory and practice. The first competition (ITC 2002) was organized in 2002 by the European Metaheuristic Network and focused in a simplified version of the university course timetabling problem (Paechter et al., 2003). Following its success, ITC 2007 was organized as the second competition, with three tracks on specific timetabling problems, as explained below (McCollum et al., 2010). The next ITC 2011 encouraged research on timetabling in the context of high school institutions (Post et al., 2013). Finally, the ongoing ITC 2019 has the goal of stimulating research on complex university course timetabling problems stemming from real-world data sets with diverse characteristics (Müller et al., 2018).

On occasion of ITC 2007, three problems were identified in the context of university timetabling: Examination Timetabling, Post-Enrollment-based Course Timetabling (PE-CTT), and Curriculum-Based Course Timetabling (CB-CTT). In the Examination Timetabling, exams have to be assigned to time slots within a given examination session, satisfying some additional hard constraints. The two other problems are variants of the Course Timetabling Problem described above. In PE-CTT students enroll for courses before the timetable is computed. Enrollment information is taken into account to determine possible timetable clashes for students, which must be avoided. In contrast, computation of a timetable in CB-CTT has to be carried out without any information about student enrollment, but on the basis of curricula that specify sets of lectures which the students are expected to follow. More precisely, in CB-CTT, the following sets are given (Bettinelli et al., 2015):

time periods : units of the time horizon considered in the planning. The planning horizon is a week (typically 5 or 6 teaching days), divided in *days*, which in turn are divided in *time slots*. A *time period* is a pair (day, time slot);

courses : each course consists of a given number of *lectures*, is taught by a *teacher* (or *lecturer*) and is attended by a given (estimated) number of *students*. A course can belong to some curricula and its lectures should spread across a *minimum number of working days*. For each course, a set of *unavailable time periods* is given, which correspond to the periods where the lecturer of the course is unavailable;

curricula : a curriculum corresponds to a set of courses that are expected to share some common students;

rooms : each room is characterized by a *capacity*, which represents the number of seats in the room.

The task of CB-CTT consists in assigning course lectures to time periods and rooms taking into account the following *hard constraints*:

H1 : all lectures of each course must be scheduled and they must be assigned to different time periods;

H2 : each room can host at most one lecture per time period;

H3 : courses taught by the same lecturer or belonging to the same curriculum cannot be scheduled in the same time period;

H4 : a lecture cannot be scheduled in an unavailable period of the corresponding lecturer.

The goal is to minimize the weighted sum (according to given weights) of multiple objectives, representing the cost for the violation of the following *soft constraints*:

S1 : for each course, a penalty is given for each day below the minimum number of working days;

S2 : lectures belonging to the same curriculum should be scheduled in consecutive time periods; a penalty is given for each *isolated lecture*, i.e., a lecture not adjacent to any other lecture of the same curriculum on the same day;

S3 : if a course lecture is scheduled to a room whose capacity is smaller than the expected number of students in the course, then a penalty is given for each student that cannot have a seat;

S4 : it is preferable that a course is always taught in the same room; otherwise, a penalty is given for each additional room used for a course.

With a few exceptions, educational timetabling problems are in general computationally hard to solve. Not surprisingly, research on the topic has been abundant and several comprehensive surveys have been published. Schaerf (1999) reviews various formulations and solution algorithms for three main classes of educational timetabling problems (School, Course, and Examination Timetabling), putting emphasis on Artificial Intelligence techniques. Burke et al. (1997) and Burke and Petrovic (2002) discuss different approaches for problem and data modeling developed at the *Automated Scheduling, Optimisation and Planning* (ASAP) research group at the University of Nottingham, as well as solution techniques based primarily on meta-heuristics and constraint logic programming. Petrovic and Burke (2004) explore in more detail four kinds of solution methods developed at ASAP, with the aim of raising the level of generality of existing algorithms: meta-heuristics, multicriteria optimization, case-based reasoning, and hyper-heuristic approaches. McCollum (2006, 2007) surveys the major challenges faced by researchers and practitioners working in the area of Course and Examination Timetabling at universities, with the purpose of motivating academics to bridge the gap between research and practice. The author addresses practical issues in the automated construction of timetables such as the number of interested parties and the diversity of data requirements. Kristiansen and Stidsen (2013) review previous works on algorithms, benchmark data-sets, and software systems for four classes of educational timetabling problems: University Course Timetabling, High School Timetabling, Examination Timetabling, and Student Sectioning (i.e., the task of splitting a course into sections and assigning students to each section, while respecting their individual course requests and some given limitations on resources). The authors specifically concentrate on articles dealing with methods tested on real data or implemented as software systems. Lewis (2008) examines solution algorithms for university timetabling problems based on meta-heuristics, underlining the various methods that have been proposed for dealing and differentiating between model constraints of varying importance. Some survey works concentrate on specific educational timetabling problems. Pillay (2014) provides a comprehensive review of High School Timetabling, categorizing available literature by the used solution methodology. Carter and Laporte (1996) and Qu et al. (2009) discuss and classify different algorithmic approaches for Examination Timetabling. Mirhassani and Habibi (2013) examine different model formulations for the University Course Timetabling Problem, comparing the most frequently used hard constraints, soft constraints, and objective values. The authors also review solution techniques proposed for the problem, based both on exact algorithms and heuristics, concluding that decomposition methods and hybridizing heuristics stand out as promising approaches that could lead to powerful solution tools. Bettinelli et al. (2015) focus on the Curriculum-Based variant of University Course Timetabling, providing an overview of mathematical models, lower bounds, and exact and heuristic algorithms proposed for CB-CTT. More recently, possible extensions of CB-CTT, motivated by practical applications, have also been discussed. For instance, a main limitation of CB-CTT lies in the fact that Student Sectioning has been neglected to keep the problem not too complex. Müller and Rudová (2016) propose a local search algorithm for generating student enrollments from an enriched curriculum based model that captures elective courses and course sections. Thus, instances from this extended variant of CB-CTT are transformed into instances of PE-CTT and tackled by solvers for the latter problem.

In this paper another variant of CB-CTT will be studied, which arises from a real-world application in the context of course timetabling at the Escuela Politécnica Nacional in Quito, Ecuador. This variant will be described in detail in the next section. An integer programming formulation for it is then proposed in Section 3. In Section 4, an alternative formulation with less variables is presented and some symmetry-breaking constraints are considered. Section 5 deals with the solution scheme proposed for this new formulation. Computational experiences are reported in Section 6. Finally, Section 7 contains some concluding remarks.

2 Problem description

Located in the city of Quito, the Escuela Politécnica Nacional (EPN) is one of the main technical universities in Ecuador, offering graduate and postgraduate programs in Engineering and Sciences. Teaching at EPN is organized on a semester basis, with the following activities being involved in the planning for each term: the list of courses to be offered is determined, lecturers are assigned to these courses, the expected number of enrolled students is estimated for each course, and a timetable is computed from this information. The planning is carried out in a decentralized way, with each academic unit (for instance, each faculty) producing the timetables for the programs under its management.

Students enroll in courses only after the timetable has been made available to the public. Thus, similarly as in CB-CTT, timetables have to be computed solely on the basis of curricula information. Moreover, due to internal regulations, students are not allowed to register in courses that have conflicting timetables, which means that poor timetables may have a strong negative impact in the duration of their studies. On the other hand, the timetabling problem at EPN has some remarkable differences to the standard version of CB-CTT introduced in the previous section. For instance, it has more hard constraints and less soft constraints than CB-CTT. In the following, these differences are highlighted in detail.

As in CB-CTT, the problem considered here consists in assigning a set of given course lectures to given sets of time periods and rooms, avoiding lecturer and curriculum conflicts. However, in this case, each course lecture has a given *duration*, i.e., a number of consecutive time periods required for the lecture to take place. Most lectures have a duration of two time periods, but there are some lectures whose durations equal to one or three time periods, and a few lectures (usually labs) have longer durations. Hence, the hard constraints H1 and H3 need to be slightly modified:

H1' : each course lecture of duration d must be scheduled to an interval of d consecutive time slots on the same day;

H3' : course lectures taught by the same lecturer or belonging to the same curriculum cannot be scheduled to overlapping intervals of time periods.

Due to internal policies at the university, not all rooms are available at all periods for a given academic unit. In fact, rooms are characterized by three properties in our problem: (i) *capacity*, expressed as the number of seats in the room; (ii) *availability*, defined by a set of time periods when the room may be used for teaching; and (iii) *type*, indicating the teaching facilities found in the room (e.g., classroom, computer lab, chemistry lab, etc.). Each course lecture specifies a required room type. A room is said to be *compatible* to a course lecture if its capacity is at least as large as the expected number of students in the course and if it has the type required by the lecture. Hard constraint H2 from CB-CTT is modified to take into account room availability:

H2' : each room can host at most one lecture per time period, and only if the room is available at that period.

Moreover, soft constraint S3 from CB-CTT is replaced by the following hard constraint in the timetabling problem at EPN:

H5 : each course lecture must be scheduled to a compatible room.

Constraint H4 remains unchanged in our problem.

Policies at EPN forbid scheduling more than one lecture of the same course on the same day. This hard constraint may be regarded as a special case of soft constraint S1 from CB-CTT, where the minimum number of working days equals to the number of lectures of the course:

H6 : for each course, at most one lecture may be taught on the same day, i.e., the number of working days must be equal to the number of course lectures.

Two soft constraints are considered in the problem. The first one accounts for the compactness of the timetable for each curriculum, an aspect which is also taken into consideration in CB-CTT. However, while in CB-CTT compactness is achieved by penalizing *isolated lectures* as in soft constraint S2, another approach is proposed here. For a given curriculum, the *length* of a working day is defined as the elapsed time from the starting of the first lecture to the end of the last lecture on that day. This value should be bounded from above:

S2' : for each curriculum, the maximum length of a working day should not exceed a prescribed limit; a penalty is given for each time period above the limit.

The other soft constraint deals with *lecturer preferences*. Due to hard constraint H4, lectures may only be scheduled to time periods where the lecturer is available for teaching. Additionally, the lecturer is allowed to specify different preference values for each of these time periods, and lectures should be scheduled only to the most-preferred time periods:

S5 : lectures should be scheduled only to time periods that have been assigned the highest preference value by the corresponding lecturer; otherwise, penalties are given for teaching at time periods with lower preference values.

No penalties are considered for courses taught in different rooms, i.e., soft constraint S4 from CB-CTT is not taken into account in the timetabling problem at EPN.

*Course conflicts.*As in the general CB-CTT, two kinds of course conflicts are identified in the timetabling problem addressed here. Two courses are said to have a *lecturer conflict* if they are taught by the same lecturer. The other kind of conflict is a *curriculum conflict*, which deserves further explanation.

Students pursuing some specific degree at EPN must approve a set of prescribed *subjects*, which are organized in a degree program. Among other things, this program specifies a level and a set of prerequisites for each subject. The *level* of a subject indicates the semester at which the student is expected to take that subject (for instance, Calculus belongs to the first level in almost all Engineering degrees); *prerequisites* are subjects that a student must have approved before being allowed to enroll in a course (e.g. Linear Algebra II has Linear Algebra I as prerequisite). Registration in courses according to the specified levels is not mandatory, but is expected from the students and assumed for the purposes of timetable computation.

One of the first steps in the planning for a semester consists in sectioning the students, i.e., classifying the students of each degree program into groups. All students within a given *student group* are expected to register in the same set of subjects. This classification is done manually and depends to a large extent on the expertise of the staff in charge of the planning. However, certain basic rules are followed: for each level of the program, there must be at least one student group expected to register in all subjects of that level; and, the total number of student groups in a degree program should not be much larger than the number of levels. Moreover, the set of subjects related to a student group cannot contain two subjects where one is a prerequisite of the other.

For each subject included in the academic offer of a given semester, one or more course sections have to be scheduled. Each course section is assigned to a lecturer and to one or more student groups, which may even belong to different degree programs. Conversely, each student group expected to register in a given subject is assigned to exactly one course section corresponding to that subject. Since there exists no risk of confusion, in the following we use the term *course* to refer to a course section. The *curriculum* of a student group is the set of all courses assigned to this student group. Two courses have a curriculum conflict if they share a common student group, i.e., if they belong to the same curriculum.

Two course lectures are defined to have a lecturer (resp. curriculum) conflict if the corresponding courses have a lecturer (resp. curriculum) conflict. Thus, hard constraint H3' can be reformulated as follows:

H3' : lectures with lecturer or curriculum conflicts must be scheduled to disjoint intervals of time periods.

3 Integer programming formulation

In this section an integer programming formulation is described for the timetabling problem at EPN. This formulation is based on the ideas proposed by Burke et al. (2011), where some modifications are included to model the specific requirements that have been detailed in the previous section.

*Notation.*The following sets and parameters are used in the model:

- C : set of courses,
- S_c : set of lectures ("sessions") of course $c \in C$,
- S : set of all lectures, $S := S \cup_{c \in C} S_c$,
- G : set of student groups,
- T : set of lecturers ("teachers"),
- C_g : set of courses (curriculum) assigned to student group $g \in G$,
- C_t : set of courses held by lecturer $t \in T$,
- D : set of days in the planning horizon,
- H : set of one-hour time periods,
- H_d : set of time periods in day $d \in D$,
- H_s : set of time periods at which lecture $s \in S$ is allowed to start,
- R : set of rooms,
- R_s : set of rooms compatible with lecture $s \in S$,
- R_c : set of rooms compatible with course $c \in C$, $R_c := \cup_{s \in S_c} R_s$.

Each course lecture $s \in S$ has a given *duration* δ_s, indicating the number of consecutive time periods required for s to take place. Observe that s may only start at certain time periods, due to the schedule availability of the corresponding teacher, but also due to its duration. For instance, $\delta_s > 1$ implies that s cannot start in the last time slot of any day. As listed above, H_s denotes the set of allowed starting time periods for s. The duration δ_c of a course is defined as the total duration of its lectures, i.e., $\delta_c := \sum_{s\in S_c} \delta_s$. Moreover, $\delta_{\max} := \max_{s\in S}\{\delta_s\}$ denotes the maximum duration of any lecture.

As noted in the previous section, each room $r \in R$ is characterized by three properties: (i) capacity, given by the number of seats in r; (ii) availability, consisting in a set of time periods in which r may be used for teaching; and (iii) type, indicating the teaching facilities found in the room. Moreover, each course lecture $s \in S$ specifies a required room type (classroom, computer lab, specific lab, etc.). Room r is said to be *compatible* to course lecture s, if the capacity of r is at least as large as the expected number of students in the course $c \in C$ for which $s \in S_c$, and if r has the type required by s. The set R_s contains all rooms compatible with course lecture $s \in S$. Room availability is expressed by means of a binary parameter ψ_{hr} which equals to one if and only if r is available at time period $h \in H$.

The start time (expressed as hour of the day) of a period $h \in H$ is denoted by an integer parameter a_h. Given a student group $g \in G$ and its corresponding curriculum C_g, a day $d \in D$ is said to be a *working day* for g if there is at least one lecture of a course $c \in C_g$ scheduled on that day. Moreover, in this case the length of working day d is defined as the elapsed time (expressed in hours) from the start of the first lecture to the end of the last lecture of a course from C_g scheduled on d, i.e., if the first lecture is scheduled to start at time period h_1 and if the last lecture is scheduled to finish at time period h_2, then the length of the working day is given by $a_{h_2} + 1 - a_{h_1}$. For each student group, the maximum length of a working day should not exceed a prescribed maximum length $L^{\max}$; otherwise, a penalty cost of W_L is incurred for each excess period.

Each lecturer $t \in T$ is available for teaching only at a certain set of time periods $H_t \subset H$. Moreover, the lecturer is allowed to specify preferences for teaching classes among the time periods in H_t. This is carried out by assigning to each $h \in H_t$ a preference value $\omega_{th} \in \Omega$, where $\Omega \subset \mathbb{Z}_+$ is a finite set of non negative integers with $0 \in \Omega$. A value $\omega_{th} = 0$ signalizes the highest preference, while higher values of ω_{th} indicate lower preferences. This values can be interpreted as penalty costs incurred for violating lecturer preferences.

Five sets of decision variables are used in the model. Binary variables $y_{shr} \in \{0,1\}$ indicate the room and start period assignment of lectures, with $y_{shr} = 1$ if and only if lecture $s \in S$ is scheduled to start at period $h \in H_s$ in room $r \in R_s$. Binary variables x_{chr} indicate whether any lecture of course $c \in C$ is assigned to take place during period $h \in H$ in room $r \in R_c$. For any student group $g \in G$ and any working day $d \in D$ for g, the non negative variables t^1_{gd} and t^2_{gd} represent the start time of the first lecture and the end time of the last lecture of a course from C_g on d, respectively. Finally, non negative variables L_g measure the excess hours of the maximum length of a working day for g with respect to the prescribed limit $L^{\max}$, i.e.,

$$L_g := \max\left\{\max_{d\in D}\{t^2_{gd} - t^1_{gd} - L^{\max}\}, 0\right\}. \qquad (1)$$

The *course assignment formulation* $\mathcal{F}_C$ for the timetabling problem at EPN reads as follows:

$$\min \sum_{t\in T}\sum_{h\in H}\sum_{c\in C_t}\sum_{r\in R_c} \omega_{th} x_{chr} + W_L \sum_{g\in G} L_g \qquad (2)$$

s.t.

$$\sum_{r\in R_s}\sum_{h\in H_s} y_{shr} = 1, \quad \forall s \in S, \tag{3}$$

$$x_{c,h+k,r} \geq y_{shr}, \quad \forall c \in C,\ s \in S_c,\ h \in H_s,\ r \in R_s,\ k \in \{0,\dots,\delta_s - 1\}, \tag{4}$$

$$\sum_{r\in R_c}\sum_{h\in H} x_{chr} = \delta_c, \quad \forall c \in C, \tag{5}$$

$$\sum_{s\in S_c}\sum_{r\in R_s}\sum_{h\in H_d\cap H_s} y_{shr} \leq 1, \quad \forall c \in C,\ d \in D, \tag{6}$$

$$\sum_{c\in C} x_{chr} \leq \psi_{hr}, \quad \forall h \in H,\ r \in R, \tag{7}$$

$$\sum_{c\in C_t}\sum_{r\in R_c} x_{chr} \leq 1, \quad \forall h \in H, \forall t \in T, \tag{8}$$

$$\sum_{c\in C_g}\sum_{r\in R_c} x_{chr} \leq 1, \quad \forall h \in H, \forall g \in G, \tag{9}$$

$$t^1_{gd} \leq a_h + M(1 - \sum_{c\in C_g}\sum_{r\in R_c} x_{chr}), \quad \forall g \in G, d \in D, h \in H_d, \tag{10}$$

$$t^2_{gd} \geq (a_h + 1)\sum_{c\in C_g}\sum_{r\in R_c} x_{chr}, \quad \forall g \in G, d \in D, h \in H_d, \tag{11}$$

$$L_g \geq t^2_{gd} - t^1_{gd} - L^{\max}, \quad \forall g \in G, \forall d \in D, \tag{12}$$

$$y_{shr} \in \{0,1\}, \quad \forall s \in S,\ h \in H_s,\ r \in R_s,$$

$$x_{chr} \in \{0,1\}, \quad \forall c \in C,\ h \in H,\ r \in R_c,$$

$$t^1_{gd}, t^2_{gd} \geq 0, \forall g \in G, d \in D,$$

$$L_g \geq 0, \forall g \in G.$$

The objective function (2) is the weighted sum of two components measuring the penalty costs incurred for violating lecturer preferences and the prescribed maximum length of a working day. Constraints (3) require each lecture $s \in S$ to be scheduled to start at a valid time period $h \in H_s$ in a compatible room $r \in R_s$. Constraints (4) are enforcing constraints linking the variables x_{chr} and y_{shr}. Inequalities (5) establish that each course must be assigned a total number of time periods which equals to its duration. Together, these three families of restrictions account for hard constraints H1' and H5 from the EPN timetabling problem. Hard constraint H4 is also enforced by these inequalities, as unavailable periods of lecturers are taken into account in the definition of the sets H_s. Inequalities (6) prevent more than one lecture of the same course to be scheduled on the same day, accounting for hard constraint H6. Inequalities (7) specify classroom availability and correspond to hard constraint H2'. Hard constraint H3' is expressed through inequalities (8) and (9) which forbid the simultaneous scheduling of lectures with lecturer or curriculum conflicts, respectively. Finally, constraints (10) to (12) enforce the correct assignment of values to variables t^1_{gd}, t^2_{gd}, and L_g. Parameter M in (10) is a sufficiently large positive constant. For instance, M can be chosen to be equal to 24, as t^1_{gd} refers to a time expressed in hours.

4 Improving the formulation

In this section some ideas for improving the integer programming model $\mathcal{F}_C$ are presented. Observe that, in any feasible solution of $\mathcal{F}_C$, the values of the course assignment variables x_{chr} are uniquely determined from the values of the binary variables y_{shr} indicating the scheduled start of lectures.

For any lecture $s \in S$ of a course $c \in C_t$ taught by a teacher $t \in T$, and for any time period $h \in H_s$, the quantity

$$\hat{\omega}_{sh} := \sum_{j=0}^{\delta_s - 1} \omega_{t,h+j} \tag{13}$$

equals the total penalty costs incurred for violating lecturer preferences if s is scheduled to start at h. Let $S_r := \{s \in S \,:\, r \in R_s\}$ be the set of lectures to which a room $r \in R$ is compatible. Moreover, define $S_{ch} := \{s \in S_c \,:\, h \in H_s\}$ to be the set of lectures of course $c \in C$ that may start at time period $h \in H$ and let $K_{sh} := \{k \in H_s \,:\, 0 \leq h - k \leq \delta_s - 1\}$ be the set of time periods at which lecture $s \in S$ may have started, given that s is taking place at time period h. Then, an alternative formulation for the university course timetabling problem at EPN with fewer binary variables is the *session start formulation* $\mathcal{F}_S$ stated below:

$$\min \sum_{s \in S} \sum_{h \in H_s} \sum_{r \in R_s} \hat{\omega}_{sh} y_{shr} + W_L \sum_{g \in G} L_g \tag{14}$$

s.t.

$$\sum_{r \in R_s} \sum_{h \in H_s} y_{shr} = 1, \quad \forall s \in S, \tag{15}$$

$$\sum_{s \in S_c} \sum_{r \in R_s} \sum_{h \in H_d \cap H_s} y_{shr} \leq 1, \quad \forall c \in C,\ d \in D, \tag{16}$$

$$\sum_{s \in S_r} \sum_{k \in K_{sh}} y_{skr} \leq \psi_{hr}, \quad \forall h \in H,\ \forall r \in R, \tag{17}$$

$$\sum_{c \in C_t} \sum_{s \in S_c} \sum_{k \in K_{sh}} \sum_{r \in R_s} y_{skr} \leq 1, \quad \forall h \in H,\ \forall t \in T, \tag{18}$$

$$\sum_{c \in C_g} \sum_{s \in S_c} \sum_{k \in K_{sh}} \sum_{r \in R_s} y_{skr} \leq 1, \quad \forall h \in H,\ \forall g \in G, \tag{19}$$

$$t^1_{gd} \leq a_h + M\Big(1 - \sum_{c \in C_g} \sum_{s \in S_{ch}} \sum_{r \in R_s} y_{shr}\Big), \quad \forall g \in G,\ d \in D,\ h \in H_d, \tag{20}$$

$$t^2_{gd} \geq (a_h + 1) \sum_{c \in C_g} \sum_{s \in S_c} \sum_{k \in K_{sh}} \sum_{r \in R_s} y_{skr}, \quad \forall g \in G,\ d \in D,\ h \in H_d, \tag{21}$$

$$L_g \geq t^2_{gd} - t^1_{gd} - L^{\max}, \quad \forall g \in G,\ d \in D, \tag{22}$$

$$y_{shr} \in \{0,1\}, \quad \forall s \in S,\ h \in H_s,\ r \in R_s,$$

$$t^1_{gd}, t^2_{gd} \geq 0, \forall g \in G,\ d \in D,$$

$$L_g \geq 0, \forall g \in G.$$

The objective function (14) aims to minimize the total penalty costs due to violations of lecturer preferences and the maximum length of working days for student groups. Constraints (15) require that each lecture has to be scheduled to start at one allowed time period and in a compatible room. Constraints (16) prevent more than one lecture of the same course to be scheduled on the same day. Constraints (17) account for classroom availability. Constraints (18) and (19) avoid simultaneous scheduling of lectures with curriculum or lecturer conflicts. Finally, constraints (20) - (22) set appropriate values for variables t^1_{gd}, t^1_{gd}, and L_g. Observe that (19) implies that the sums of the assignment variables on the right-hand side of (20) and (21) are constrained to be equal to zero or one.

4.1 Dealing with symmetries

Feasible solutions of $\mathcal{F}_S$ show (at least) three different types of symmetries, regarding the scheduling of lectures of the same course, as well as the room assignment of lectures. More precisely, let $s_1, \ldots, s_m$ be lectures of the same course $c \in C$ having the same duration and sharing the same set of compatible rooms, i.e., $\delta_{s_1} = \cdots = \delta_{s_m}$ and $R_{s_1} = \cdots = R_{s_m}$. Assume a feasible solution of $\mathcal{F}_S$ assigns these lectures to time periods $h_1, \ldots, h_m$ and to rooms $r_1, \ldots, r_m$, respectively, i.e.,

$$y_{s_i h_i r_i} = 1, \quad , \forall i \in \{1, \ldots, m\}.$$

For any permutation π on the set $\{1, \ldots, m\}$, it is straightforward to prove that the assignment

$$y_{s_i h_{\pi(i)} r_{\pi(i)}} = 1, \quad , \forall i \in \{1, \ldots, m\},$$

defines a new feasible solution with the same objective value. Indeed, neither the schedules of teachers nor the schedules of students groups are changed, as all rescheduled lectures $s_1, \ldots, s_m$ belong to the same course. Moreover, for any $i \in \{1, \ldots, m\}$, the lecture s_i is assigned to a compatible room $r_{\pi(i)} \in R_{s_{\pi(i)}}$, since $R_{s_{\pi(i)}} = R_{s_i}$. This room is available for the whole interval of time required by s_i, since it was previously assigned to lecture $s_{\pi(i)}$ which has the same duration.

Analogously, consider a feasible solution of $\mathcal{F}_S$ in which lectures $s_1, \ldots, s_n$ have been scheduled to start at the same time period $h \in H$ in rooms $r_1, \ldots, r_n$, respectively. Moreover, assume the lectures have the same duration and share the same compatible rooms, i.e., $\delta_{s_1} = \cdots = \delta_{s_n}$ and $R_{s_1} = \cdots = R_{s_n}$. Then, with a similar argument as above, for any permutation π on the set $\{1, \ldots, n\}$, the assignment given by

$$y_{s_i h r_{\pi(i)}} = 1, \quad , \forall i \in \{1, \ldots, n\},$$

defines a new feasible solution of $\mathcal{F}_S$ with the same objective value.

Finally, assume a solution of $\mathcal{F}_S$ consists in assigning a lecture $s \in S$ to start at a time period $h \in H_s$ in a room $r \in R_s$. Moreover, let $r_1, \ldots, r_l$ be rooms which are: (i) compatible with s, (ii) available during the whole time interval when s takes place, and, (iii) not assigned to any other lecture during this interval. Then, a new feasible solution of the same cost can be obtained by switching the room assignment of s from r to any of the other rooms, i.e., by setting

$$y_{shr} = 0 \quad \text{and} \quad y_{shr_i} = 1,$$

for any $i \in \{1, \ldots, l\}$.

To deal with these symmetries, precedence constraints are included in the model. Let $\Delta := \{1, \ldots, \delta_{\max}\}$ and Q be the set of room types. Observe that the set R of rooms can be partitioned as $R = \uplus_{q \in Q} R^q$, where R^q is the set of rooms of type $q \in Q$. For any course $c \in C$, let $S_c^{q\delta}$ be the set of lectures from S_c that have duration equal to $\delta \in \Delta$ and require a room of type $q \in Q$. Since any two lectures of the same course have the same group of students, they require rooms of the same capacity. Thus, any two lectures of the same course share the same set of compatible rooms if and only if they require the rooms of the same type, i.e., $R_{s_i} = R_{s_j}$ holds for any $s_i, s_j \in S_c^{q\delta}$.

An order relation is defined on the set of lectures by means of a bijection $\sigma : S \to \{1, \ldots, |S|\}$. Making use of this relation, the following linear constraint avoids symmetries of the first type. For any two lecturers $s_i, s_j \in S_c^{q\delta}$, with $\sigma(s_i) < \sigma(s_j)$, and any day $d \in D$ of the week, the constraint states that s_i may be scheduled to take place on d only if s_j is scheduled on a later day:

$$\sum_{h \in H_d \cap H_{s_i}} \sum_{r \in R_{s_i}} y_{s_i h r} \leq \sum_{\hat{d} \in D_d^+} \sum_{h \in H_{\hat{d}} \cap H_{s_j}} \sum_{r \in R_{s_j}} y_{s_j h r}, \tag{23}$$

where D_d^+ is the set of days of the week that come after day d. If d is the last (working) day of the week, D_d^+ is the defined to be the empty set and the right-hand side of (23) evaluates to zero. Thus, these constraints prevent a lecture $s_i \in S_c^{q\delta}$ to be scheduled on the last day of the week whenever there is a lecture $s_j \in S_c^{q\delta}$ with $\sigma(s_i) < \sigma(s_j)$. Conversely, s_j cannot be scheduled on the first day of the week, as otherwise s_i cannot be scheduled on any other day without violating (23). This idea can be further extended to set the values of some variables to zero. Indeed, assume the lectures from $S_c^{q\delta}$ have been relabeled as $\tilde{s}_1, \ldots, \tilde{s}_m$ such that $\sigma(\tilde{s}_i) < \sigma(\tilde{s}_j)$ holds for all $1 \leq i < j \leq m$. Then, the next equation follows from (15) and (23) for any $i \in \{1, \ldots, m\}$:

$$\sum_{d \in D_{i-1}^F} \sum_{h \in H_d \cap H_{\tilde{s}_i}} \sum_{r \in R_{\tilde{s}_i}} y_{\tilde{s}_i h r} + \sum_{d \in D_{m-i}^L} \sum_{h \in H_d \cap H_{\tilde{s}_i}} \sum_{r \in R_{\tilde{s}_i}} y_{\tilde{s}_i h r} = 0, \tag{24}$$

where D_i^F and D_i^L denote the sets of the i first and the i last working days of the week, respectively.

Let n_s be the number of students expected to attend lecture $s \in S$ and u_r the capacity of room $r \in R$. To prevent symmetries of the second type, an order relation is defined on the set of rooms by means of a bijection $\hat{\sigma} : R \to \{1, \ldots, |R|\}$, such that $\hat{\sigma}(r) < \hat{\sigma}(\hat{r})$ implies $u_r \leq u_{\hat{r}}$. Moreover, assume in the following that the bijection σ on the set of lectures defined above has been chosen such that $\sigma(s) < \sigma(\hat{s})$ implies $n_s \leq n_{\hat{s}}$. Denote by $S^{q\delta}$ the set of lectures (belonging to any course) that have duration equal to $\delta \in \Delta$ and require a room of type $q \in Q$. Assume that two lectures $s, \hat{s} \in S^{q\delta}$, with $\sigma(s) < \sigma(\hat{s})$, have been scheduled to start at the same time period $h \in H_s \cap H_{\hat{s}}$ in rooms $r, \hat{r} \in R_s \cap R_{\hat{s}}$, respectively. Observe that $R_s \cap R_{\hat{s}} = R_{\hat{s}}$, as both lectures require the same type of room and $n_{\hat{s}} \geq n_s$. The following inequality requires that in this case $\hat{\sigma}(r) < \hat{\sigma}(\hat{r})$ must hold:

$$y_{shr} \leq \sum_{\hat{r} \in R^+(r)} y_{\hat{s}h\hat{r}} + \sum_{\hat{h} \neq h} \sum_{\hat{r} \in R_{\hat{s}}} y_{\hat{s}\hat{h}\hat{r}}, \tag{25}$$

where $R^+(r) := \{\hat{r} \in R_{\hat{s}} : \hat{\sigma}(\hat{r}) > \hat{\sigma}(r)\}$. The expression on the right-hand side evaluates to zero if and only if both sums equal to zero. This occurs if $\hat{s}$ is scheduled to start at period h (second sum evaluates to zero) and assigned to a room $\hat{r} \in R_{\hat{s}}$ with $\hat{\sigma}(\hat{r}) < \hat{\sigma}(r)$ (first sum equals to zero). In this case $y_{shr} \leq 0$, and s is not allowed to start at period h in room r.

Finally, symmetries of the third type are addressed as follows. For any $h \in H$ and $\delta \in \Delta$, let I_h^δ be the interval of time periods $\{h, \dots, h+\delta-1\}$ and define $R_h^{q\delta} := \{r \in R^q : \psi_{\hat{h}r} = 1, \hat{h} \in I_h^\delta\}$ to be the set of rooms of type $q \in Q$ that are available during I_h^δ. Let $r, \hat{r} \in R_h^{q\delta}$, with $\hat{\sigma}(r) < \hat{\sigma}(\hat{r})$. Observe that $S_r \subseteq S_{\hat{r}}$, i.e., any lecture compatible with r is also compatible with $\hat{r}$, as $\hat{r}$ has at least the same capacity as r, and both rooms have the same type. The following constraint prevents any lecture in S_r with duration δ to be scheduled to start at h in $\hat{r}$ as long as r is free during time interval I_h^δ:

$$\sum_{s \in S^{q\delta} \cap S_r} y_{sh\hat{r}} \leq \sum_{\hat{h} \in I_h^\delta} \sum_{\hat{s} \in S_r} \sum_{k \in K_{\hat{s}\hat{h}}} y_{\hat{s}kr}, \tag{26}$$

where, as indicated at the beginning of this section, $K_{\hat{s}\hat{h}} := \{k \in H_{\hat{s}} : 0 \leq \hat{h} - k \leq \delta_s - 1\}$ is the set of periods at which lecture $\hat{s}$ may have started if it is taking place at period $\hat{h}$. The right-hand side equals to zero if r is not being used by any lecture during the interval I_h^δ. In this case $y_{sh\hat{r}}$ is forced to be equal to zero for all lectures of duration δ which can assigned to r, i.e., for all lectures in $S^{q\delta} \cap S_r$. In particular, any lecture of duration δ, requiring a room of type q, and scheduled to start at period h must be assigned to a compatible free room $r \in R_h^{q\delta}$ having the smallest possible value of $\hat{\sigma}(r)$.

5 Solution scheme

Preliminary tests on the available practical instances revealed that the compactness requirements seem to be the component of the session start formulation $\mathcal{F}_S$ which makes the model computationally hard to solve. This observation motivates the solution approach proposed in this paper, which consists of two phases carried out in cascade. In the first phase, a reduced model is solved which is obtained from $\mathcal{F}_S$ by dropping variables and constraints related to compactness of schedules for student groups (i.e., neglecting soft constraint S2'). In the second phase, the best solution obtained for this reduced model is propagated to a feasible initial solution for $\mathcal{F}_S$, and the latter model is solved by a branch-and-cut approach to improve the objective function value. In both phases, a general purpose MIP solver is used for handling the integer programming models; more details are provided in the next section.

5.1 Phase 1: Obtaining a feasible schedule

In this phase a reduced formulation $\mathcal{F}_S^R$ consisting of the objective function

$$\min \sum_{s \in S} \sum_{h \in H_s} \sum_{r \in R_s} \hat{\omega}_{sh} y_{shr}$$

together with constraints (15)-(19) is solved to optimality. The resulting integer program has $O(|S| \cdot |H| \cdot |R|)$ binary variables and $O(|S| + |C| \cdot |D| + |H|\,(|R| + |T| + |G|))$ constraints. This means $O(|G| \cdot |D|)$ fewer (continuous) variables and $O(|G| \cdot |H|)$ fewer constraints than in the original model $\mathcal{F}_S$.

5.2 Phase 2: Improving compactness

A feasible solution for the session start formulation $\mathcal{F}_S$ can be computed in a straightforward manner from the optimal solution for $\mathcal{F}_S^R$ obtained in the first phase. Indeed, given solution values y^*_{shr} for the binary variables indicating the start of lectures, the best-possible values for the remaining model variables are determined as follows. For each group of students $g \in G$ and each working day $d \in D$ for g, let:

$$t^{*1}_{gd} := \min\{a_h : h \in H_d, \sum_{c \in C_g} \sum_{s \in S_c} \sum_{r \in R_s} y^*_{shr} = 1\},$$

$$t^{*2}_{gd} := \max\{(a_h + 1) : h \in H_d, \sum_{c \in C_g} \sum_{s \in S_c} \sum_{k \in K_{sh}} \sum_{r \in R_s} y^*_{skr} = 1\}.$$

Moreover, if $d \in D$ is *not* a working day for g, i.e., if

$$\sum_{h \in H_d} \sum_{c \in C_g} \sum_{s \in S_c} \sum_{r \in R_s} y^*_{shr} = 0,$$

then t^{*1}_{gd} and t^{*2}_{gd} are set to be equal to zero. Finally, for each $g \in G$, let L^*_g be defined by (1).

The feasible solution $(y^*_{shr}, t^{*1}_{gd}, t^{*2}_{gd}, L^*_g)$ provides an upper bound for the optimal value of the session start formulation $\mathcal{F}_S$ and is also used as a start solution for the second phase. During this phase, a branch-and-cut algorithm with a prescribed time limit is applied on model $\mathcal{F}_S$ to explore the solution search space with the aim of improving compactness of the timetable.

6 Computational results

Computational tests were performed on real-world instances stemming from two faculties at EPN: the Faculty of Sciences and the Faculty of Chemical Engineering. The Faculty of Sciences offers four degree programs in Mathematics, Physics, Mathematical Engineering, and Economy. Each of these four programs has a duration of nine academic semesters and, altogether, they comprehend 337 possible subjects. Meanwhile, two degree programs in Chemical and Agroindustrial Engineering including a total of 201 possible subjects are managed by the Faculty of Chemical Engineering. Details on the test instances are shown in Table 1. The first two instances (sciences and chemical) correspond to the complete planning of one academic term for each of the faculties, encompassing the scheduling of the course lectures for all degree programs offered by each faculty. The following six instances (sciences2, ..., chemical4) have been obtained by simplifying the two original instances in some way, considering only either a subset of the degree programs or a subset of the subjects and lecturers for each program. From left to right,

Instance	$\lvert C\rvert$	$\lvert S\rvert$	$\lvert T\rvert$	$\lvert G\rvert$	$\lvert R\rvert$	$\overline{\lvert H_t\rvert}$	$\overline{\delta_{C_t}}$	$\overline{\lvert R_s\rvert}$	W_L
sciences	115	272	68	25	22	25.1	7.9	12.9	328
chemical	162	252	54	26	42	37.3	8.7	11.0	4
sciences2	47	111	37	33	13	21.6	5.9	6.4	3
sciences3	91	223	59	33	20	26.4	7.4	11.8	18
sciences4	70	168	53	14	18	24.4	6.2	11.2	276
chemical2	149	231	55	28	42	37.4	8.1	10.0	3
chemical3	136	206	44	24	42	35.2	8.8	10.7	155
chemical4	105	186	40	20	23	32.5	8.4	12.5	3

Table 1 Description of test instances.

the columns report the following information: instance name, number of courses, number of course lectures, number of lecturers, number of student groups, number of rooms, average number of available periods in the week per lecturer, average teaching workload (number of teaching periods in the week) per lecturer, average number of available rooms per lecture, and the value of the weight parameter W_L for the compactness term in the objective function (14).

In all instances, a planning horizon of $|D| = 6$ days (from Monday to Saturday) has been considered, accounting for a total of $|H| = 76$ time periods (14 periods each day from Monday to Friday and 6 periods on Saturday). The values for parameter a_h indicating the start time of a period in (20) and (21) range from 7 to 21. Hence, $M = 100$ was chosen as a sufficiently large value for the constant appearing in the first of these constraint families.

Most of the lectures have a duration equal to two, but the distribution of these values varies between instances stemming from the Faculty of Sciences and the Faculty of Chemical Engineering. In the first case, between 90% and 97% of the lectures in each instance have a duration of two, while the percentage of lectures with duration equal to one ranges from 3% to 7%, and 4% or less of the lectures have durations equal to three. Meanwhile, lectures with duration equal to two account only for between 63% and 66% of all lectures in the instances stemming from the Faculty of Chemical Engineering. Between 24% and 27% of the lectures in these instances have a duration equal to one, while lectures with duration equal to three account for between 8% and 10% of the lectures. Moreover, there are some specific labs, representing less than 3% of the lectures, whose durations equal to four or five time periods.

A set $\Omega := \{0, 1, 100\}$ of three possible preference values was used to express the schedule preferences of lecturers. A value $\omega_{th} = 0$ indicates that lecturer $t \in T$ prefers to teach a lesson at period $h \in H$; a value of 1 indicates that t neither likes nor dislikes teaching a lesson at h; and a value of 100 indicates that t strongly dislikes the idea of teaching at period h. Lecturer preferences were then obtained from polls carried out among the faculty. In the Faculty of Chemical Engineering, lecturers marked on average 37 periods in the week with preferences in $\{0, 1\}$, while the corresponding figure in the Faculty of Sciences was of 24 periods per week on average. The average teaching workload for a lecturer lies around 8 periods per week in the Faculty of Chemical Engineering and 7 periods per week in the Faculty of Sciences. Parameters $\hat{\omega}_{sh}$ in the objective function (14) were computed from the values of ω_{th} as indicated in the formula (13).

All experiments were performed on an Intel Core i7 3.60 GHz with 8 GB RAM running Ubuntu 14.01. The integer programs were solved using the Gurobi Optimizer solver 8.1 (Gurobi Optimization, 2019).

Compactness requirements for the schedules of student groups are the computationally most challenging component of the session start formulation $\mathcal{F}_S$, as shown in Table 2. The optimality gaps and running times obtained for $\mathcal{F}_S$ on the eight test instances (columns %gap and T, respectively) are compared to the corresponding values obtained for the reduced formulation $\mathcal{F}_S^R$, which is derived from $\mathcal{F}_S$ by dropping variables and constraints related to compactness. In both cases, the Gurobi Optimizer solver was run using its default settings. While only one instance could be solved to optimality within a time limit of 5,000 seconds for model $\mathcal{F}_S$, all instances were solved to optimality within 181 seconds running time for model $\mathcal{F}_S^R$. The number of variables is about the same in both models, whereas the number of constraints in $\mathcal{F}_S^R$ is roughly one half of the number of constraints in $\mathcal{F}_S$.

The two-phase solution approach described in Section 5 consists in extending the optimal solution obtained for $\mathcal{F}_S^R$ to a feasible initial solution for $\mathcal{F}_S$, by computing appropriate values for variables t^1_{gd}, t^2_{gd}, and L_g. Since one of the main purposes of the present article is to study the effect of compactness requirements on model $\mathcal{F}_S$, the penalty cost W_L in the objective function (14) has been adjusted to keep the term measuring violations of the prescribed maximum length of a working day much larger than the term measuring violations of lecturer preferences. More precisely, the values for W_L reported in Table 1 were obtained by applying the formula $W_L := \lfloor \frac{\mathrm{UB}^R}{2} \rfloor$ to the optimal values for $\mathcal{F}_S^R$ reported in column UB^R, in the lower part of Table 2. The last two columns UB and %S2' depict the value of the optimal solution of $\mathcal{F}_S^R$, when considered as a feasible solution of $\mathcal{F}_S$, as well as the weight of the compactness term within this value, expressed as a percentage. With the chosen values for W_L, the compactness term accounts for between 95% and 98% of the total solution cost in the test instances.

Results for the application of the two-phase solution strategy under four different scenarios are reported in Table 3. The columns of the table show the instance name, followed by data for each phase of the solution approach, i.e., for each one of the models $\mathcal{F}_S^R$ and $\mathcal{F}_S$. For the first phase, the table displays the number of variables and constraints in $\mathcal{F}_S^R$, together with the value of the best feasible solution (UB^1), the optimality gap ($\%\mathrm{gap}^1$), the running time (T^1) and number of nodes explored in the branch-and-bound tree (N^1). For the second phase, the number of variables and constraints in $\mathcal{F}_S$, the value of the initial feasible solution computed in the first phase (UB^2_0), the best feasible solution value (UB^2), the best lower bound (LB^2), the optimality gap ($\%\mathrm{gap}^2$), the running time (T^2), and the number of nodes explored in the branch-and-bound tree (N^2) are depicted.

In the first scenario, models $\mathcal{F}_S^R$ and $\mathcal{F}_S$ are solved using the Gurobi Optimizer configured to run without presolving and without cut generation. As expected, the solution scheme achieves its worst performance under this scenario, with optimality gaps for the eight test instances ranging from 91.1% to 98.7% after 5,000 seconds running time in the second phase. Moreover, even if model $\mathcal{F}_S^R$ is solved to optimality in seven instances, running times in the first phase are larger on average than the corresponding running times reported in the lower part of Table 2. The instance sciences4 could not be solved to optimality within the time limit of 5,000 seconds for the first phase.

Session start formulation $\mathcal{F}_S$						
Instance	Vars.	Cons.	UB	LB	%gap	T
sciences	211,074	15,322	5,110	1,064.13	79.2	5,000.14
chemical	199,718	20,166	92	22.82	75.2	5,000.05
sciences2	49,292	5,698	51	38.43	24.7	5,000.26
sciences3	157,998	12,519	153	114.88	24.9	5,000.04
sciences4	119,482	11,146	1,344	725.11	46.0	5,001.18
chemical2	167,384	15,466	21	7.00	66.7	5,000.04
chemical3	159,497	16,977	1,005	572.26	43.1	5,000.06
chemical4	168,650	13,926	12	12.00	-	2,737.84

Reduced formulation $\mathcal{F}_S^R$							
Instance	Vars.	Cons.	UB^R	%gap	T	UB	%S2'
sciences	210,645	9,166	656	-	180.59	30,374	98
chemical	199,289	11,776	9	-	29.21	255	96
sciences2	49,110	3,299	6	-	0.67	161	96
sciences3	157,634	7,199	37	-	3.25	1,572	98
sciences4	119,170	6,538	552	-	66.18	15,478	96
chemical2	167,124	9,677	7	-	18.99	122	94
chemical3	159,159	10,633	311	-	5.48	6,212	95
chemical4	168,325	7,940	6	-	16.96	150	96

Table 2 Effect of the compactness requirements on the computational hardness of the problem. Comparison of optimality gaps and running times of the original session start formulation $\mathcal{F}_S$ and the reduced formulation $\mathcal{F}_S^R$ (neglecting compactness) on all test instances.

The second scenario consists in solving the integer programming models with the Gurobi Optimizer running in its default configuration, i.e., under the same conditions as the tests reported on Table 2. The optimality gaps after a time limit of 5,000 seconds for the second phase range from 0% to 72.2% for the eight test instances, with one instance solved to optimality. Except for instance chemical4, the optimality gaps obtained by the two-stage solution method are smaller than the gaps obtained when solving the session start formulation from scratch as in Table 2, with an average reduction of 10.3%.

Constraints (23) and (24) are included in both models to break symmetries of the first type (i.e., symmetries involving the scheduling of lectures belonging to the same course) under the third scenario, resulting in a slight increase in the number of constraints. The best performance of the proposed solution method is attained for this scenario. Three of the eight test instances are solved to optimality within the prescribed time limit for the second phase. For the remaining five instances, the optimality gaps range from 11.3% to 56.5%. Compared to the results obtained under the second scenario, the optimality gap decreases on all but one instances, with an average decrease of 9.5%. The results obtained during the first phase outperform the corresponding results under the second scenario, as well. The running time needed to find the optimal solution for the reduced model decreases in five of the eight instances, with an average decrease of 23.7 seconds. Moreover, all instances are solved either in the root node or in the second node of the branch-and-bound tree, whereas under the second scenario two instances required the exploration of more than 1,000 nodes.

The impact of including inequalities (25) and (26) for breaking symmetries of the second and third types (involving the assignment of rooms) was considered under the fourth scenario. Due to the large number of these inequalities, they were not added directly to the models, but separated dynamically as lazy constraints. Columns nL^1 and nL^2 indicate the number of inequalities added during each phase. Constraints for dealing with symmetries of the first type were also kept under this scenario. The performance of the solution algorithm does not improve after the inclusion of the new classes of symmetry-breaking inequalities. On the contrary, the results obtained for all test instances are significantly worse than the results under the second scenario. The optimality gaps for the two complete instances (sciences and chemical) surpass 95%, while for the six simplified instances the values range from 28.7% to 87.3%. Model $\mathcal{F}_S^R$ is solved to optimality on all instances during the first phase, but the running times are significantly higher than the corresponding running times under the second and third scenarios.

7 Conclusions

An integer programming model has been proposed for a variant of the Curriculum-Based Timetabling problem arising from a real-world application. Two key features of this model are the existence of course lectures with different durations, as well as a soft constraint prescribing a maximum length for the working day of student curricula, with the aim of providing compact schedules for students. Computational tests reveal that these compactness requirements are particularly difficult to handle by the integer programming solver. This observation motivates the use of a two-phase solution approach in which compactness is at first neglected from the model and included only at a second stage, after a solution has been found, which is optimal with respect to the remaining component of the objective function. It is worth noting that this approach produces better results than solving the original model from scratch, even if the weights in the objective are chosen in such a way that the compactness term accounts for more than 95% of the value of the initial feasible solution.

Due to its relationship with graph coloring problems, timetabling problems are known to admit feasible solutions with several symmetries. For this model three types of symmetries were studied: one of them regarding the time period and room assignment for lectures within the same course, and the other two related to the room assignment for lectures. To deal with these symmetries, three classes of symmetry-breaking constraints have been proposed and tested, obtaining different results. While the use of inequalities that avoid the first kind of symmetries significantly improves the performance of the solution algorithm on the test instances, including the other two families of inequalities leads to worse computational results. This may be due to the fact that the reduction in the size of the search space obtained by adding these inequalities is outweighed by the increase in the size of the integer programs. Further analysis is needed in order to determine if and how room symmetries can be exploited to devise better solution methods in this specific variant of CB-CTT.

References

Bettinelli A, Cacchiani V, Roberti R, Toth P (2015) An overview of curriculum-based course timetabling. Top 23(2):350–354, DOI 10.1007/s11750-015-0366-z

Scenario I: No cuts, no presolve, no sym. ineqs.															
Phase 1							Phase 2								
Instance	Vars.	Cons.	UB^1	$\%gap^1$	T^1	N^1	Vars.	Cons.	UB^2_0	UB^2	LB^2	$\%gap^2$	T^2	N^2
sciences	210,645	9,166	656	-	118.5	1	211,074	15,322	34,444	34,444	446.6	98.7	5,001.7	5,104
chemical	199,289	11,776	9	-	3,073.0	1,163	199,718	20,166	254	254	6.0	97.6	5,001.4	797
sciences2	49,110	3,299	6	-	0.5	0	49,292	5,698	161	71	6.2	91.1	5,000.1	1,357,810
sciences3	157,634	7,199	37	-	3.1	0	157,998	12,519	1,572	1,102	37.1	96.6	5,009.6	72,424
sciences4	119,170	6,538	552	18.8	5,002.6	378,136	119,482	11,146	16,429	5,476	448.3	91.8	5,003.2	116,242
chemical2	167,124	9,677	7	-	297.9	104	167,384	15,466	124	124	7.0	94.3	5,000.3	7,304
chemical3	159,159	10,633	311	-	3.2	0	159,497	16,977	6,314	6,314	311.0	95.0	5,001.0	13,033
chemical4	168,325	7,940	6	-	579.5	2,409	168,650	13,926	150	150	3.5	97.6	5,000.3	6,203
Scenario II: Cuts, presolve, no sym. ineqs.														
Phase 1							Phase 2							
Instance	Vars.	Cons.	UB^1	$\%gap^1$	T^1	N^1	Vars.	Cons.	UB^2_0	UB^2	LB^2	$\%gap^2$	T^2	N^2
sciences	210,645	9,166	656	-	180.6	1,099	211,074	15,322	30,374	3,696	1,342.0	63.7	5,000.1	5,370
chemical	199,289	11,776	9	-	**29.2**	1	199,718	20,166	255	81	22.5	72.2	5,000.1	5,277
sciences2	49,110	3,299	6	-	**0.7**	0	49,292	5,698	161	51	41.9	17.8	5,003.8	1,892,690
sciences3	157,634	7,199	37	-	**3.3**	0	157,998	12,519	1,572	153	153.0	-	2,281.1	20,652
sciences4	119,170	6,538	552	-	66.2	2,265	119,482	11,146	15,478	1,064	802.1	24.6	5,000.1	12,737
chemical2	167,124	9,677	7	-	19.0	1	167,384	15,466	122	17	7.0	58.8	5,000.8	20,882
chemical3	159,159	10,633	311	-	5.5	1	159,497	16,977	6,212	787	703.0	**10.7**	5,000.1	5,621
chemical4	168,325	7,940	6	-	17.0	1	168,650	13,926	150	17	12.0	29.4	5,000.2	21,466
Scenario III: Cuts, presolve, lecture sym. ineqs.														
Phase 1							Phase 2							
Instance	Vars.	Cons.	UB^1	$\%gap^1$	T^1	N^1	Vars.	Cons.	UB^2_0	UB^2	LB^2	$\%gap^2$	T^2	N^2
sciences	210,645	10,256	656	-	**51.4**	1	211,074	16,412	35,632	3,148	1,368.5	**56.5**	5,000.2	5,770
chemical	199,289	12,155	9	-	29.7	1	199,718	20,545	246	73	25.7	**64.8**	5,000.0	286
sciences2	49,110	3,790	6	-	0.8	0	49,292	6,189	171	51	45.2	**11.3**	5,005.7	2,230,870
sciences3	157,634	8,100	37	-	4.0	0	157,998	13,420	1,752	153	153.0	**-**	576.5	5,984
sciences4	119,170	7,166	552	-	**18.0**	1	119,482	11,774	13,898	870	870.0	**-**	3,811.1	8,499
chemical2	167,124	9,997	7	-	**8.2**	1	167,384	15,786	133	17	10.0	**41.2**	5,000.5	10,463
chemical3	159,159	10,943	311	-	**4.5**	0	159,497	17,287	6,675	885	645.2	27.1	5,000.7	5,477
chemical4	168,325	8,236	6	-	**15.5**	1	168,650	14,222	135	12	12.0	**-**	3,405.5	10,906

Table 3 Performance of the two-phase solution scheme under four scenarios. Values in boldface indicate the best running times (first phase) and optimality gaps (second phase) obtained for each instance.

Scenario IV: Cuts, presolve, lecture + room sym. ineqs.							
Phase 1							
Instance	Vars.	Cons.	UB^1	$\%gap^1$	T^1	N^1	nL^1
sciences	210,645	10,256	656	-	1,134.9	1,218	630
chemical	199,289	12,155	9	-	187.9	0	150
sciences2	49,110	3,790	6	-	14.7	0	135
sciences3	157,634	8,100	37	-	59.1	0	196
sciences4	119,170	7,166	552	-	219.8	2,015	283
chemical2	167,124	9,997	7	-	72.3	0	62
chemical3	159,159	10,943	311	-	44.0	0	79
chemical4	168,325	8,236	6	-	790.1	1,288	695

Phase 2									
Instance	Vars.	Cons.	UB_0^2	UB^2	LB^2	$\%gap^2$	T^2	N^2	nL^2
sciences	211,074	16,412	37,271	23,460	980.6	95.8	5,013	5,257	721
chemical	199,718	20,545	260	260	12.6	95.1	5,000	1,309	347
sciences2	49,292	6,189	172	51	36.4	28.7	5,000	1,900,152	395
sciences3	157,998	13,420	1,790	191	69.6	63.5	5,000	25,674	773
sciences4	119,482	11,774	17,462	1,547	678.1	56.2	5,000	5,848	792
chemical2	167,384	15,786	105	51	9.0	82.4	5,006	5,445	226
chemical3	159,497	17,287	6,623	3,762	477.6	87.3	5,008	5,608	993
chemical4	168,650	14,222	131	21	11.8	43.8	5,000	3,500	892

Table 3 (cont'd) Performance of the two-phase solution scheme under four scenarios. Values in boldface indicate the best running times (first phase) and optimality gaps (second phase) obtained for each instance.

Burke EK, Petrovic S (2002) Recent research directions in automated timetabling. European Journal of Operational Research 140(2):266–280, DOI 10.1016/S0377-2217(02)00069-3

Burke EK, Jackson KS, Kingston JH, Weare R (1997) Automated Timetabling: The State of the Art. The Computer Journal 40:565–571, DOI 10.1093/comjnl/40.9.565

Burke EK, Mareček J, Parkes AJ, Rudová H (2011) Decomposition, reformulation, and diving in university course timetabling. Computers and Operations Research 37(3):582–597, DOI 10.1016/j.cor.2009.02.023

Carter MW, Laporte G (1996) Recent developments in practical examination timetabling. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Springer, Berlin, Heidelberg, vol 1153, pp 3–21, DOI 10.1007/3-540-61794-9_49, URL http://link.springer.com/10.1007/3-540-61794-9_49

Gurobi Optimization L (2019) Gurobi optimizer reference manual. URL http://www.gurobi.com

Kristiansen S, Stidsen TR (2013) A Comprehensive Study of Educational Timetabling, a Survey. Tech. Rep. November, DTU

Lewis R (2008) A Survey of Metaheuristic-based Techniques for University Timetabling Problems. OR Spectrum 30(1):167–190

McCollum B (2006) University Timetabling: Bridging the Gap Between Research and Practice. In: Burke EK, Rudová H (eds) PATAT 2006, May 2005, pp 15–35, DOI 10.1097/00024665-200603000-00007

McCollum B (2007) A Perspective on Bridging the Gap Between Theory and Practice in University Timetabling. In: Practice and Theory of Automated Timetabling VI, pp 3–23, DOI 10.1007/978-3-540-77345-0-1

McCollum B, Schaerf A, Paechter B, McMullan P, Lewis R, Parkes AJ, Di Gaspero L, Qu R, Burke EK (2010) Setting the Research Agenda in Automated Timetabling : The Second International Timetabling Competition. INFORMS Journal on Computing 22(1):120–130, DOI 10.1287/ijoc.1090.0320

Mirhassani SA, Habibi F (2013) Solution approaches to the course timetabling problem. Artificial Intelligence Review 39(2):133–149, DOI 10.1007/s10462-011-9262-6

Müller T, Rudová H (2016) Real-life curriculum-based timetabling with elective courses and course sections. Annals of Operations Research 239(1):153–170, DOI 10.1007/s10479-014-1643-1

Müller T, Rudová H, Müllerová Z (2018) University course timetabling and International Timetabling Competition 2019. In: Proceedings of the 12th International Conference on the Practice ant Theory of Automated Timetabling (PATAT 2018), pp 5–31, URL http://www.itc2019.org.

Paechter B, Gambardella L, Rossi-Doria O (2003) International Timetabling Competition. Metaheuristics Network 31

Petrovic S, Burke EK (2004) University timetabling. Handbook of Scheduling: Algorithms, Models, and Performance Analysis, chapter 45 pp 1–34, DOI doi:10.1201/9780203489802.ch45

Pillay N (2014) A survey of school timetabling research. Annals of Operations Research 218(1):261–293, DOI 10.1007/s10479-013-1321-8

Post G, Di Gaspero L, Kingston JH, McCollum B, Schaerf A (2013) The Third International Timetabling Competition. Annals of Operations Research 239(1):69–75, DOI 10.1007/s10479-013-1340-5

Qu R, Burke EK, McCollum B, Merlot LT, Lee SY (2009) A survey of search methodologies and automated system development for examination timetabling. Journal of Scheduling 12(1):55–89, DOI 10.1007/s10951-008-0077-5

Schaerf A (1999) A Survey of Automated Timetabling. Artificial Intelligence Review 13(2):87–127, DOI 10.1023/A:1006576209967

Wren A (1996) Scheduling, timetabling and rostering - A special relationship? In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Springer, Berlin, Heidelberg, vol 1153, pp 46–75, DOI 10.1007/3-540-61794-9_51, URL http://link.springer.com/10.1007/3-540-61794-9_51