Available online at www.sciencedirect.com
SCIENCE DIRECT®
European Journal of Operational Research 153 (2004) 117–135
EUROPEAN JOURNAL OF OPERATIONAL RESEARCH
www.elsevier.com/locate/dsw

An integer programming formulation for a case study in university timetabling

S. Daskalaki [a,*], T. Birbas [b], E. Housos [b]

[a] Department of Engineering Sciences, University of Patras, GR-26500 Rio Patras, Greece
[b] Department of Electrical and Computer Engineering, University of Patras, GR-26500 Rio Patras, Greece

Abstract

A novel 0–1 integer programming formulation of the university timetabling problem is presented. The model provides constraints for a great number of operational rules and requirements found in most academic institutions. Treated as an optimization problem, the objective is to minimize a linear cost function. With this objective, it is possible to consider the satisfaction of expressed preferences regarding teaching periods or days of the week or even classrooms for specified courses. Moreover, with suitable definition of the cost coefficients in the objective function it is possible to reduce the solution space and make the problem tractable. The model is solvable by existing software tools with IP solvers, even for large departments. The case of a five-year Engineering Department with a large number of courses and teachers is presented along with its solution as resulted from the presented IP formulation.

Keywords: Timetabling; Integer programming; University timetabling

1. Introduction

The construction of a timetable that satisfies all operational rules and needs in an academic institution, while at the same time fulfills as many of the wishes and requirements of the staff and the students is an important but extremely difficult task for the staff involved. In most institutions this task is left to administrative staff and the current practice is to replicate the timetables of previous years with minor changes to accommodate newly developed situations. However, in recent years, changes occur more frequently and patching of what has been developed historically is not always the best policy. Under these circumstances, and in light of the progress achieved both in the hardware and software technologies, the scientific community continues to work on the problem in order to develop formal and automated procedures for constructing efficient and desirable timetables.

Formally, the university timetabling problem is defined as the process of assigning university courses to specific time periods throughout the five working days of the week and to specific classrooms suitable for

* Corresponding author. Tel./fax: +30-2610-997810.
E-mail addresses: sdask@upatras.gr (S. Daskalaki), tbirbas@ee.upatras.gr (T. Birbas), housos@ee.upatras.gr (E. Housos).

doi:10.1016/S0377-2217(03)00103-6

the number of students registered and the needs of each course. For the problem we are modeling in this paper, courses are offered to well-defined groups of students that follow a semi-structured schedule and in that sense the problem belongs to the category of the class/teacher timetabling problem for university environments [10]. For every educational institution the objective is always the construction of effective and satisfactory weekly timetables. A timetable is considered to be effective when it is feasible and may be realised by the institution, while it is considered to be satisfactory when it carries certain quality characteristics that keep its users satisfied at least to a certain degree.

The university timetabling is the third stage of our effort to solve the timetabling problem for all three levels of education in Greece. The first step faced the problem of constructing timetables for the Greek high schools [7,8], the second solved the problem for the Greek Lyceums and this last effort gives a thorough model and an initial solution for the university environment. The timetabling problem is known to be an NP-complete problem in most all of its forms [4,34], meaning that if all combinations were to be examined, the time to solution for reasonable problems would rise dramatically.

The timetabling problem, like many others in the area of combinatorial optimization, has been approached by several well-known techniques of the operational research and the computer science fields. Several surveys on course timetabling [11] and automated timetabling [29], as well as others on more focused aspects of the problem, have managed to record this work in a systematic way, categorizing thus the different variations of the problem and solution approaches. Given the aforementioned work, this paper does not attempt to cover exhaustively the state-of-the-art in this area, except from pointing out the main contributions of our work with respect to other integer programming formulations.

The paper is organized as follows. Section 2 presents a brief historical review of the timetabling problems concentrating mainly on the integer programming (IP) formulations that appear in the literature. Then Section 3 gives the rationale behind our approach with IP. Section 4 describes the characteristics typically carried by courses offered in a university. Section 5 lists the rules that timetablers have to obey in order for their product to be acceptable (basic rules), as well as a number of proposed conventions that a timetable is desired to hold (quality rules). The IP formulation of the problem is given in Section 6, which comprises complete description of the constraints and the objective function of the model. Section 7 gives an approach to defining the cost coefficients involved in the formulation of the objective function. A real timetabling problem for the department of Electrical and Computer Engineering is finally discussed through in Section 8 and the solution provided from the proposed model is partially presented. Section 9 summarizes the presented work and concludes with remarks from our experience with the timetabling problem.

2. Review of similar problems and state-of-the-art

As early as the decade of sixties simulation approaches, other heuristics and graph colouring methods appeared in the literature while they were attempting to solve some variations of the problem in an effective way. Specifically, Welsh and Powell [35] use the graph colouring technique to solve a timetabling problem, while Schmidt and Strohlein [30] report on simulation techniques. More than 30 years later, there is still the need to approach the problem in a systematic way that facilitates the inclusion of real-world variants of the problem along with automation and solution of very big problems.

Among the first approaches in mathematical programming, Lawrie in [25] and Akkoyunlu in [1] present linear and integer programming models for some versions of the problem and thus succeed in computing optimal solutions for a school and a university timetabling problem, respectively. More specifically, in [25] the timetabling process starts with the generation of the so-called "layouts" for the different groups of students and an integer programming formulation attempts to find a set of optimal "arrangements", one for each group. Conversely, in [1] the problem is solved as an assignment problem between a set of courses and the time periods of a weekly timetable, with central focus on preventing conflicts throughout this assignment.

In [9] a solution is provided for the faculty assignment problem, a problem closely related to the timetabling problem, using linear programming models. The same problem was studied in [26] and a solution was attempted, again with the help of mathematical programming. More recently, in [22] the teacher assignment problem is formulated as a MIP problem and is solved as a special case of the fixed charge transportation problem. In this case the proposed specialized algorithm, perform better than publicly available packages. Using goal programming, in [3] the teacher assignment problem is combined with a form of the timetabling problem and solved through commercial software for goal programming. In a similar manner, in [20] a linear programming formulation is provided for the classroom allocation problem, a sub-problem of the university timetabling.

IP formulations for the school and the university timetabling problems as optimization problems are also given in [33], where the NP-completeness of both problems is shown even for simple versions. The author, however, chooses graph theory approaches for the solution of the problems under consideration. Extensions to this work, especially with regards to the so-called conflict matrix are presented in [31]. Given the difficulties of those days to solve large IP problems, the Lagrangean relaxation is proposed as a possible solution approach for the resulting model. However, in other real-world applications [32] a heuristic approach is proposed instead.

Along with the mathematical formulation, several authors divide requirements into two groups; the hard ones, which are included in the constraints and they define the search space, and the soft ones, which are included in some way in the objective function [18]. A similar strategy for a timetabling problem for universities is solved in [2] by grouping sub-problems. A solution approach for the same problem, however with lectures of different length is provided in [19]; in this formulation approach lectures may last one, two or three periods.

Apart from the classical mathematical programming approaches, several new and most efficient techniques for combinatorial problems have also being used for the timetabling problems. Amongst these, tabu search was used in [12,21] for the solution of the school and university timetabling problems; constraint logic programming are presented in [13,24]. Last but not least, genetic algorithms have been utilized as an effective tool for the solution of timetabling problems. For example in [5,6,10,16] as well as in [27,28] university timetabling problems are solved through these techniques.

3. Main principles in our approach

In recent years, because of the advancements in computer software and hardware, IP and MIP formulations have started being again an acceptable approach for many combinatorial problems [23]. The new technologies in information systems, the availability of reliable software and the ability to solve relatively large problems in relatively short time are the main reasons for making this traditional modeling approach attractive for the solution of realistic problems. Two decades back the problems that were solvable by classical IP techniques, mainly branch-and-bound, carried tens of integer variables. Now a problem with many thousands and in special occasions millions of binary variables is not necessarily a problem. In regards with the timetabling problems, IP models have been presented in [14] for the university timetabling problem and in [7,8] for the school timetabling problem; the solutions produced with commercial software presented no real problem in terms of computation times.

In the sections that follow, a university timetabling problem is modeled as an optimization problem using 0–1 variables. The model provides constraints for a large number of different rules and regulations that exist in academic environments. More specifically, the model succeeds in creating timetables that are free from collisions between courses, teachers and classrooms and they are complete from all aspects;

moreover, it supports the scheduling of courses that require consecutive time periods, as well as courses that require sessions that are repeated several times to accommodate different groups of students. Lastly, pre-assignments can be easily introduced into the model, as well as free-of-teaching time during the week.

The IP formulation followed in this paper is a novel one and the model carries certain unique features compared to others. The binary variables are defined in such a way that the structural elements: courses, students, teachers, days and periods are preserved in the model in a distinct way. This choice was made knowing that in reality the assignment is performed between the set of triplets (course, teacher, group of students) and the set of pairs (period, day). Of course this choice leads to a very large number of possible variables, however with the introduction of suitable subsets of the basic sets of days, time periods, groups of students, teachers, courses and classrooms, this number reduces to manageable sizes. On the other side, the proposed definition of our binary variables leads to very flexible elements as far as the modeling is concerned. Carrying all the important information along with the variables provided great flexibility in the modeling process, so that almost every functional rule that the university timetabling process presented was easily turned into mathematical equations.

Moreover, the introduction of the auxiliary variables for the consecutiveness and repetitiveness constraints was also quite successful in managing the complexity of these issues. It is known that requests for consecutive time slots in timetabling problems increase complexity and make the problem NP-Hard [17]. The emphasis of the paper is on modeling the problem using IP, however, the models that resulted from the real-world problem in our case study were still solvable by commercial packages using the classical branch-and-bound approach. The solutions found through the presented model are optimal in the sense that they minimize a cost function, while the objective function was chosen to be the means of introducing preferences for certain time periods, days and classrooms for all courses involved.

4. Characteristics of university timetabling

The university timetabling problem carry special features that highly depend on the characteristics that the courses taught in a university carry along, as well as on the arrangements that each university provides for the utilization of its resources.

4.1. Structure of university courses

A course offered by any department in the university may comprise of just lectures, or lectures and recitations, or lectures, recitations, and/or work in a lab.

The *lectures* are delivered by professors, lecturers or other teaching staff and they choose to carry out the weekly requirement in single- or multi-period sessions. Occasionally two or more professors may be assigned the lectures for a given course, in which case they have to decide the type of sharing for the teaching load.

Time slots designated for *recitations* quite often are treated just like time for lectures, covered by the same person that does the lectures, while in other cases they may be considered as time for exercises and better understanding of the material covered during lectures. In this last case recitations are assigned to one or more different persons that take responsibility for the recitations, while the class of the students is split in several small groups.

Lab work is usually part of a given course or sometimes a course by itself and the group of students is split into several sub-groups for training. Lab work takes place in specially equipped classrooms and is assigned to the professor that teaches the lectures or to lab assistants.

4.2. Types of university courses

In many universities courses are characterized as either *mandatory* or *electives*. Mandatory courses are those that a given department considers as basic for the training of its students. During lower-grade years, most courses are mandatory; therefore they are designed for all students of the same year. On the contrary, electives are less in number, however in the timetable they should never overlap with any mandatory or other elective course.

In higher-grade years, respectively, mandatory courses are less in number, while electives increase exponentially. In addition, during higher-grade years it is very common for a number of departments to streamline their students to several divisions and each division has its own set of mandatory and elective courses. Then, timetabling becomes even more complicated, given that some courses may overlap because there is no common interest between divisions, while other courses are shared between divisions and their schedule should be convenient for all interested students.

The characteristics of university courses just described, along with the fact that different students may belong to different recitation or lab groups, pose many difficulties in controlling student schedules. As we shall see further down quality rules placed by schedulers to make timetables easier for students have limited usage, since each student has his/her own schedule. However, in our model some considerations for improving attendance of the students during lower-grade courses are possible by giving preferential treatment to those that are generally accepted as "difficult" courses.

4.3. Availability of resources

For the university timetabling problem resources refer to *human resources* and *classrooms*.

Availability of human resources for teaching is defined by the people themselves. Duties of administrative nature consume a number of periods during the week from each individual in the teaching staff. To make things more controllable, in Greek universities the meetings for the general assembly of each department are scheduled for a fixed day and time interval, so that courses for all participants are scheduled around those time periods. In addition, each individual in the pool of the teaching staff should provide all other periods that may be unavailable for teaching.

Availability of classrooms is defined by the system followed by the university. Some classrooms may be considered as available during all periods, some others however have limited availability. For example, in Greek universities it is common for several departments to share some classrooms, so that one department has access to a given classroom some days of the week and other departments the rest of the days. The sharing of classrooms is particularly common for big auditoriums, a scarce commodity in most institutions.

5. Rules for timetabling in the university

Building a weekly timetable for an institution is a tedious process that administrators usually undertake after spending several man-hours. In this effort timetablers follow several rules, some of which are so important that they may never be violated (hard constraints) and others that are not as important and usually are obeyed only if all hard constraints are satisfied and there is still room for better solutions under some objectives for improved quality (soft constraints).

Hard constraints for the university timetabling are regulated by the following basic rules:

1. *Collisions may not be permitted*. In a timetable a collision occurs when two or more courses are scheduled at the same period for the same teaching person, for the same group of students or for the same classroom. Also, when two or more teaching persons are assigned to the same group of students to teach two

different courses; and lastly when two or more classrooms are assigned to the same course and to the same group of students.
2. *A timetable has to be complete*. A timetable is complete when all courses planned for every group of students appear in the timetable, with the right amount of time periods for every course and every portion of each course. Also, when every person of the teaching staff is assigned the total number of teaching periods that the institution requires.
3. *A timetable should accommodate requests for sessions of consecutive teaching periods*. Depending on the course and the number of teaching periods assigned per week, a professor may choose to give the course in single-period or multi-period sessions. The timetable has to be able to schedule a given course in any scheme that the responsible professor may choose to follow.
4. *A timetable should accommodate requests for repetitive sessions of a given course or part of a course*. This rule refers to the need presented by some courses for sessions of recitations or lab work that are designed for small groups. In this case the same session has to be repeated several times to accommodate the total number of registered students.

Similarly, soft constraints for the university timetabling are regulated by the following quality rules:

1. *Preferences for teaching in specific time intervals are obeyed if possible.* Each person from the teaching staff may express an opinion regarding the preferred period for teaching his/her courses. For example, one may prefer to teach a given course during morning periods, while another during afternoon or evening periods.
2. *Student schedules should be as compact as possible*, *however allow for breaks during lunch hours.* Student timetables are easier to follow if timetablers take advantage of those periods that students are more alert and schedule the so-called "difficult" courses during those periods. The other courses may then fill the rest of a given day in a way to minimize the empty slots between attendances. However, in most cases, students prefer an empty slot around lunch time, so they can rest before they go on with their classes.
3. *Minimize classroom changeovers especially for students of lower-grade years.* In lower-grade years students attend classes in big auditoriums and it is much more preferred to stay in the same place for a few time periods before they make a change for a laboratory or for a class with smaller audience.

6. Modeling the university timetabling problem

Following similar approach as in [7,8], we use IP to build the model that may construct university timetables. In this modeling philosophy the equations for the actual IP model may alter from institution to institution to reflect the special requirements imposed by each one of them. However, the underlying structure of the model remains the same and is described in the next section. Further down the equations for modeling the timetabling problem under specific assumptions given by engineering departments in our university is presented.

6.1. General features of the model

Six parameters are considered as the basic structural elements for this approach. These are:

- The *day* of the week, on which a course or part of a course may be scheduled. The set of all days possible for scheduling is denoted with the letter I, e.g. $I = \{1, 2, 3, 4, 5\}$.
- The *time period* of a day, on which a period of a course may be scheduled. In our approach, a time period is any one-hour period from 8:00 a.m. till 9:00 p.m. Traditionally in Greek universities the teaching

period is 45 minutes, thus allowing 15 minutes for a short break and change of teaching staff. The set of all time periods in a given day available for scheduling is denoted with the letter J, e.g. $J = \{1, 2, \dots, 14\}$.
- The *group of students* for which a course in the timetable is designed. In our approach, examples of groups of students may be the first year students of a given department or the fourth year students of a given division in a given department. The set of all different groups of students that may be used for a given timetable is denoted with the letter K, e.g. $K = \{\text{group\#1}, \text{group\#2}, \dots, \text{group\#}|K|\}$.
- The *professor*, *lecturer or other teaching staff*, from now on called *teacher* who is going to teach a course in the timetable. For our problem, it is assumed that the assignment of courses to teachers precedes the timetabling process, and the teaching load of each teacher is an input to the timetablers. The set of all teachers to be utilized in a given timetable is denoted with the letter L, e.g. $L = \{\text{teach\#1}, \text{teach\#2}, \dots, \text{teach\#}|L|\}$.
- The *courses* to be scheduled for a given set of groups of students. As explained earlier all different parts of a given course have to appear in a weekly timetable and in a desired scheme, different for each course. The set of all courses to be scheduled in a given timetable is denoted with the letter M, e.g. $M = \{\text{course\#1}, \text{course\#2}, \dots, \text{course\#}|M|\}$.
- The *classrooms* that may be available for scheduling courses for a given set of group of students. The set of all classrooms to be utilized in a given timetable is denoted with the letter N, e.g. $N = \{\text{classroom\#1}, \text{classroom\#2}, \dots, \text{classroom\#}|N|\}$.

Two different sets of binary variables are adopted. Throughout this paper, the first one is called the *basic set of variables* and is denoted by $x_{i,j,k,l,m,n}$, where $i \in I$, $j \in J$, $k \in K$, $l \in L$, $m \in M$, and $n \in N$. The variable $x_{i,j,k,l,m,n}$ takes the value of 1, when course m, taught by teacher l to the group of students k, is scheduled for the jth period of day i in classroom n. The second set of variables will be called *auxiliary variables* and will be denoted by $y_{i,p_v,k,h_v,m,n}$, where $i \in I$, $k \in K$, $m \in M$, and $n \in N$, while p_v and h_v are natural numbers. The variable $y_{i,p_v,k,h_v,m,n}$ will take the value of 1, when course m, which requires a session of h_v consecutive periods, is scheduled for day i for the group of students k in classroom n. Index p_v takes the values $1, 2, 3, \dots, p\max_m$ when the session has to be repeated $p\max_m$ times. When the variable $y_{i,p_v,k,h_v,m,n}$ refers to lecture sessions, which are given just once, then p_v takes only the value of 1.

6.2. Special features of the model

The two sets of variables just described are quite detailed, therefore flexible enough and with their help all basic rules for the timetabling problem are easily described. However, in order to keep the number of the variables manageable several subsets of the original sets I, J, K, L, M, and N are defined. In addition, a number of new sets are defined a priori so that certain indices use those as domains instead of the original ones. Thus the number of variables is further reduced, while modeling of certain constraints becomes much easier. These new sets are the following:

$K^1 = \{k \in K$: $k =$ group of students of lower-grade years, usually students from the first two or three years, following general education within a department$\}$.
$K^2 = \{k \in K$: $k =$ group of students of higher-grade years (registered to a specific division)$\}$.
It is apparent that K^1 and K^2 are disjoint sets, while $K = K^1 \cup K^2$.
$K_l = \{k \in K$: $k =$ group of students (from either K^1 or K^2) for which teacher l offers some course$\}$.

$L_i = \{l \in L$: $l =$ teacher available on day $i\}$.
$L_k = \{l \in L$: $l =$ teacher teaching at least one course for the group of students k, $k \in K\}$.
$L_m = \{l \in L$: $l =$ teacher teaching course $m\}$.
$L_{km} = L_k \cap L_m$ and $L_{ki} = L_k \cap L_i$.

$M_k = \{m \in M$: $m =$ course designed for the kth group of students or offered by the kth division for its students, $k \in K\}$.
$M_l = \{m \in M$: $m =$ course taught by teacher $l\}$.
$M_n = \{m \in M$: $m =$ course designed for a group of students that fits in classroom $n \in N\}$.
$M_{kl} = M_k \cap M_l$, $M_{kn} = M_k \cap M_n$, and $M_{kln} = M_k \cap M_l \cap M_n$.
$M_k^{\mathrm{com}} = \{(k_a, l, m) \in K^2 \times L \times M$: $m =$ course taught by teacher l for the student group $k_a \in K^2$, offered additionally as elective course for the student group $k \in K^2\}$.
$M_{\mathrm{lab}} = \{m \in M$: $m =$ course requiring lab work$\}$.

$N_{mk} = \{n \in N$: $n =$ classroom that fits the group of students $k \in K$ for the course $m \in M\}$.

$I_n = \{i \in I$: $i =$ day on which classroom n is available for use$\}$.
$I_l = \{i \in I$: $i =$ day on which teacher $l \in L$ is available for teaching assignments$\}$.
$I_{ln} = I_l \cap I_n$.

$J_{iln} = \{j \in J$: $j =$ time period of day i on which teacher $l \in L$ and classroom $n \in N$ are available for assignments$\}$.
$JL_{iln} = \{(j_a, j_b) \in J_{iln} \times J_{iln}$: $(j_a, j_b) =$ time interval from period j_a to j_b of day i on which teacher $l \in L$ and classroom $n \in N$ are available for assignments$\}$.
$FJL_{iln} = \{j_a \in J_{iln}$: $j_a =$ the starting period of an interval of day i on which teacher $l \in L$ and classroom $n \in N$ are available for assignments$\}$.

$P_m = \{p_v \in \{1, 2, \ldots, p\max_m\}$, where $p\max_m$ is the number of repetitions required for a multi-period session of course $m \in M\}$.
$H_m = \{h_v \in N^+$: $h_v =$ length (number of periods) of a multi-period session requested for course $m \in M\}$.
$\mathrm{PRA} = \{(i, j, k, l, m, n) \in I \times J \times K \times L \times M \times N / (i, j, k, l, m, n) =$ an a priori assignment in the timetable, that is, course m taught by teacher l for the group of students k is pre-assigned to period j of day $i\}$.

6.3. Constraints for the IP model

In this section the constraints for the IP model along with the corresponding rule that each one of them models are presented. For presentation purposes constraints are grouped into five groups, each consisting of a certain number of equations.

6.3.1. Uniqueness constraints

This set of constraints ensure that there are no conflicts in the timetable (basic rule #1). More specifically:

1. Every member of the teaching staff shall be assigned at most one course, one group of students and one classroom at a time:

$$\forall i \in I, \quad \forall j \in J, \quad \forall l \in L_i, \quad \sum_{k \in K_l} \sum_{m \in M_{kl}} \sum_{n \in N_{mk}} x_{i,j,k,l,m,n} \leqslant 1. \tag{1}$$

2. For every group of students at most one course, one teaching person and one classroom shall be assigned to every teaching period. This requirement further ensures that for every group of students there will be no conflicts between mandatory courses but also between mandatory and elective courses, so that students may choose to attend them:

$$\forall k \in K^1, \quad \forall i \in I, \quad \forall j \in J, \quad \sum_{l \in L_{ki}} \sum_{m \in M_{kl}} \sum_{n \in N_{mk}} x_{i,j,k,l,m,n} \leqslant 1, \tag{2}$$

$$\forall k \in K^2, \quad \forall i \in I, \quad \forall j \in J, \quad \sum_{l \in L_{ki}} \sum_{m \in M_{kl}} \sum_{n \in N_{mk}} x_{i,j,k,l,m,n} + \sum_{(k_a,l,m) \in M_k^{\mathrm{com}}} \sum_{n \in N_{mk_a}} x_{i,j,k_a,l,m,n} \leqslant 1. \tag{3}$$

Eq. (2) refers to lower-grade students, who attend courses designed for the total number of students in a given student year. Similarly, Eq. (3) refers to higher-grade students, who may attend the courses offered by the division of their choice and a number of additionally suggested courses offered by other divisions. This complication for student groups in K^2 is reflected by the two terms of the left hand side in the inequality.

3. Every classroom may be assigned to at most one course, one teacher and one group of students at a time:

$$\forall n \in N, \quad \forall i \in I_n, \quad \forall j \in J_{in}, \quad \sum_{k \in K} \sum_{l \in L_{ki}} \sum_{m \in M_{kln}} x_{i,j,k,l,m,n} \leqslant 1. \tag{4}$$

6.3.2. Completeness constraints

This set of constraints ensure that the timetable is complete (basic rule #2)

4. All courses in the curriculum of each student year should be in the timetable and in the right amount of teaching periods:

$$\forall k \in K^1, \quad \sum_{l \in L_k} \sum_{m \in M_{kl}} \sum_{n \in N_{mk}} \sum_{i \in I_{ln}} \sum_{j \in J_{iln}} x_{i,j,k,l,m,n} = \boldsymbol{a}_k, \tag{5}$$

$$\forall k \in K^2, \quad \sum_{l \in L_k} \sum_{m \in M_{kl}} \sum_{n \in N_{mk}} \sum_{i \in I_{ln}} \sum_{j \in J_{iln}} x_{i,j,k,l,m,n} + \sum_{(k_a,l,m) \in M_k^{\mathrm{com}}} \sum_{n \in N_{mk_a}} \sum_{i \in I_{ln}} \sum_{j \in J_{iln}} x_{i,j,k,l,m,n} = \boldsymbol{a}_k, \tag{6}$$

where $\boldsymbol{a}_k$ is the total number of teaching periods planned for the kth group of students.

5. Each course should be scheduled for as many teaching periods as the curriculum of each group of students requires

$$\forall k \in K, \quad \forall l \in L_k, \quad \forall m \in M_{kl}, \quad \sum_{n \in N_{mk}} \sum_{i \in I_{ln}} \sum_{j \in J_{iln}} x_{i,j,k,l,m,n} = \boldsymbol{b}_m, \tag{7}$$

where $\boldsymbol{b}_m$ is the total number of teaching periods required for course m.

6. Each person in the teaching staff should be assigned to so many teaching periods as his/her weekly teaching load requires

$$\forall l \in L, \quad \sum_{k \in K} \sum_{m \in M_{kl}} \sum_{n \in N_{mk}} \sum_{i \in I_{ln}} \sum_{j \in J_{iln}} x_{i,j,k,l,m,n} = \boldsymbol{s}_l, \tag{8}$$

where $\boldsymbol{s}_l$ is the total number of teaching periods required from teacher l every week according to the departmental assignments.

6.3.3. Consecutiveness constraints

This set of constraints ensures that the timetable may manage requests for multi-period sessions in some courses (basic rule #3). Requests for multi-period sessions are quite common for university courses and may concern either part of a course, that is the lectures, recitations, or lab work. In our model it is assumed that the professor in charge of a course provides the desired split for the course.

In order to facilitate the modeling of this need, the auxiliary variables $y_{i,p_v,k,h_v,m,n}$ are introduced, for each course m for which there is request for at least one multi-period session. As mentioned previously, the indices i, k, m, and n are defined just like for the $x_{i,j,k,l,m,n}$ variables, while index h_v gives the number of consecutive periods requested for the course (length of the session). For example, if course m requires totally 5 teaching periods per week for lectures, the professor may choose to split them in one session of

3 periods and one session of 2 periods or in two sessions of 2 periods each and one session of 1 period, or any other possible split. In the first case index h_v takes the values 2 and 3, while in the second case h_v takes the values 1 and 2.

In terms of the constraints in the IP model, consecutiveness is obtained in two steps, as follows:

7. A course m requiring a session of $h_v \in H_m$ consecutive periods should be assigned exactly h_v periods on a given day:

$$\forall i \in I, \quad \forall k \in K, \quad \forall l \in L_{ki}, \quad \forall m \in M_{kl}, \quad \forall n \in N_{mk},$$
$$\sum_{j \in JL_{iln}} x_{i,j,k,l,m,n} - \sum_{h_v \in H_m} \sum_{p_v \in P_m} (y_{i,p_v,k,h_v,m,n} * h_v) = 0. \tag{9}$$

8. If h_v periods of a given day have been assigned to course m, they should also be consecutive:

$$\forall i \in I, \quad \forall k \in K, \quad \forall l \in L_{ki}, \quad \forall m \in M_{kl}, \quad \forall n \in N_{mk}, \quad \forall j_a \in FJL_{iln}, \quad \forall h_v \in H_m \wedge h_v > 1,$$
$$\forall t \in \{1, \ldots, h_v - 1\}, \quad x_{i,j_a,k,l,m,n} - x_{i,j_a+t,k,l,m,n} \leqslant 0; \tag{10}$$

$$\forall i \in I, \quad \forall k \in K, \quad \forall l \in L_{ki}, \quad \forall m \in M_{kl}, \quad \forall n \in N_{mk}, \quad \forall j \in J_{iln}, \quad \forall h_v \in H_m \wedge h_v > 1,$$
$$\forall t \in \{2, \ldots, h_v\}, \quad -x_{i,j,k,l,m,n} + x_{i,j+1,k,l,m,n} - x_{i,j+t,k,l,m,n} \leqslant 0. \tag{11}$$

Eqs. (10) and (11) express the logic that if a given period of a day is assigned course m, then the following $h_v - 1$ periods should be assigned the same course. Eq. (10) takes care of the request for consecutiveness in case that the first period of an available interval is assigned course m, while Eq. (11) does the same in all other cases.

6.3.4. Repetitiveness constraints

This set of constraints is also connected with the idea of consecutiveness in the sense that they secure the existence of the right amount of sessions of a certain type as well as the right amount of repetitions of a given session in case of repetitive recitations or lab work.

9. The h_v-period sessions should be as many as required during the week and for the non-repetitive parts of courses there should be at most one session per day:

$$\forall k \in K, \quad \forall m \in M_k - M_{lab}, \quad \forall h_v \in H_m, \quad \forall n \in N_{mk}, \quad \sum_{i \in I_{ln}} \sum_{p_v \in P_m} y_{i,p_v,k,h_v,m,n} = b_{m,h_v}, \tag{12}$$

where b_{mh_v} is the total number of the h_v-period sessions required for course m during a week. The constraint expressed by Eq. (12) refers to courses that require non-repetitive sessions, that is lectures or recitations that are delivered just once to their audience. For this reason, the next constraint additionally secures the existence of at most one of these sessions per day:

$$\forall k \in K, \quad \forall m \in M_k - M_{lab}, \quad \forall l \in L_{km}, \quad \forall h_v \in H_m, \quad \forall n \in N_{mk}, \quad \forall i \in I,$$
$$\sum_{h_v \in H_m} \sum_{p_v \in P_m} y_{i,p_v,k,h_v,m,n} \leqslant 1. \tag{13}$$

10. In case of repetitive parts of courses, like the lab work for a given course, the scheduling of the h_v-period sessions does not require different days for different sessions, since each session is delivered to a different sub-group of the kth group of students and the previous two constraints are replaced by Eq. (14).

$$\forall k \in K, \quad \forall m \in M_{lab}, \quad \forall h_v \in H_m, \quad \sum_{n \in N_{mk}} \sum_{i \in I_n} \sum_{p_v \in P_m} y_{i,p_v,k,h_v,m,n} = p\max{}_m, \tag{14}$$

where $p\max{}_m$ is the number of student sub-groups and therefore the number of repetitions required for the h_v-period session of course m.

6.3.5. Pre-assignment constraints

This last set of constraints ensures that certain courses will be assigned at a given period in a given day and could be used either for the exact pre-allocation of courses or for the facilitation and better handling of computational difficulties.

11. Course m taught by teacher l to student group k should be assigned to a given period in a given day:

$$\forall (i,j,k,l,m,n) \in \mathrm{PRA}, \quad x_{i,j,k,l,m,n} = 1, \tag{15}$$

where PRA is the set with all required pre-assignments.

6.4. Objective function for the IP model

The constraints just presented, when placed in an integer programming model, are capable of returning feasible solutions, i.e. assignments that do not violate any of the basic rules. However, certain assignments are preferable than others and the improvement of the suggested solutions is the responsibility of the objective function, which in this model is defined to be the following:

$$\begin{aligned}\text{Minimize}\Bigg\{ & \sum_{k\in K}\sum_{l\in L_k}\sum_{m\in M_{kl}}\sum_{n\in N_{mk}}\sum_{i\in I_l}\sum_{j\in J_{iln}} c_{i,j,k,l,m,n} * x_{i,j,k,l,m,n} \\ & + \sum_{k\in K}\sum_{i\in I}\sum_{m\in M_k}\sum_{n\in N_{mk}}\sum_{h_v\in H_m}\sum_{p_v\in P_m} a_{i,p_v,k,h_v,m,n} * y_{i,p_v,k,h_v,m,n}\Bigg\}.\end{aligned} \tag{16}$$

As can be seen in Eq. (16), the objective is to minimize a cost function consisting of two terms. The first term of the objective function refers to the cost of assigning course m to the jth period of day i, while the second term refers to the cost incurred from the assignment of those courses that require sessions of more than one consecutive hours, on a given day of the week.

7. Determination of cost coefficients

In any given timetabling problem the cost coefficients in the objective function shown in Eq. (16) may take any value. In fact, if all cost coefficients take the same value, then the problem becomes degenerate and all feasible solutions will be optimal. In all practical situations this, of course, is not possible and a more specific analysis for the correct assignment of the cost coefficients is needed. While rules for assigning costs to time periods may be the subject of further research, in our model the cost coefficients are assigned values in such a way as to reflect preferences for specific time periods of the day for all courses and for specific days of the week for the courses with multiple time periods sessions. Preferences for certain classrooms are also incorporated in the $c_{i,j,k,l,m,n}$ cost coefficients of the objective function. In order to make this more understandable, each case is discussed separately.

7.1. Assignment of values to $c_{i,j,k,l,m,n}$ coefficients according to requests from teachers, students or the department chairman for specific time periods of the day

It is well accepted among teachers, students and the people that handle the timetabling process that certain courses should have preferential treatment in terms of the time period that they are assigned. For each department there is a set of courses that are considered more difficult than others and require "prime time", while other courses are less demanding and can be taught at any time of the day. In order to

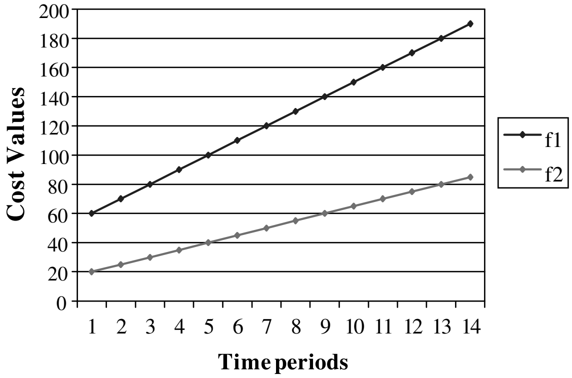

Fig. 1. Penalty functions for assigning values to the $c_{i,j,k,l,m,n}$ cost coefficients: f_2 is used for the time periods mostly preferred and f_1 for those that are less preferred.

influence the assignment of courses to their respective preferred time periods, the $c_{i,j,k,l,m,n}$ coefficients are assigned values derived from predefined functions, called *penalty functions*.

Examples of such penalty functions are shown in Fig. 1, even though in other cases they may take quite different and more complicated shapes. The figure shows two linear functions with ascending slope as the day progresses. The $c_{i,j,k,l,m,n}$ coefficients that correspond to any given course take values from either one of the two functions for each time period according to the preferences of the department. This assignment may be repeated for each day of the week. An example of such an assignment for a given day and a given course is shown in Fig. 2. It is shown that for this particular course the time periods 9–10, 10–11, 11–12, 12–13, 13–14 are the preferred time periods, followed by the time periods 16–17, 8–9, 17–18, 18–19, 19–20, 20–21, 14–15, 15–16. This approach of defining the cost coefficients may be considered as an extension to the V-shaped function suggested in [15] and is very beneficial to the optimization process.

The slope given to functions f_1 and f_2 serves a significant additional purpose: the students should end up with a schedule that is as compact as possible meaning there exist a minimal number of empty time periods between lectures. This quality rule is important, however it cannot always be satisfied given that there are also elective courses, as well as lab work and recitations that are performed in groups. These activities are performed in smaller groups and as a result individual students of the same year of study may indeed have different daily and weekly schedules.

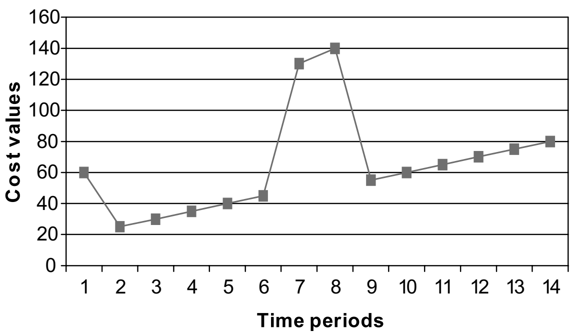

Fig. 2. Example of an assignment of values to $c_{i,j,k,l,m,n}$ cost coefficients that correspond to a given course at a given day.

7.2. Assignment of values to $c_{i,j,k,l,m,n}$ coefficients to achieve minimal classroom changeovers between courses

Another quality rule that may be partially satisfied through the assignment of suitable values to the $c_{i,j,k,l,m,n}$ *coefficients* is that of minimal classroom changeovers between lectures. This is achieved through an agreement under which each student group is assigned its so-called "prime classroom". With this assumption it is possible to assign smaller values to those $c_{i,j,k,l,m,n}$ that relate to this classroom and for all courses taught to this specific group of students.

7.3. Assignment of values to $a_{i,p_v,k,h_v,m,n}$ coefficients according to requests from teachers, students or the department in general for specific days of the week

In the objective function of the IP model the $a_{i,p_v,k,h_v,m,n}$'s are the coefficients for the auxiliary variables, which are used for the scheduling of those courses that require sessions of multiple time periods to specific days. By giving values to the $a_{i,p_v,k,h_v,m,n}$ coefficients it is thus possible to influence assignments for each one of these courses to preferred days. For example, it is possible to avoid assigning lab work on Fridays, by assigning higher values to the $a_{i,p_v,k,h_v,m,n}$ coefficients that correspond to $i = 5$ and to the lab sessions.

The tri-fold way of defining the cost coefficients in the objective function of the IP model just discussed is followed in the example presented in the next section and proved to provide very satisfactory solutions for the required timetable.

8. Case study

In order to demonstrate the capabilities of the proposed model for the solution of the university timetabling problem, the department of Electrical and Computer Engineering at the University of Patras was chosen. The department, which is one of the two largest departments in the Engineering School, has a five-year programme for its students. During the first three years the students obtain a general education in Electrical and Computer Engineering and during the last two years each student chooses one of the four divisions of the department in order to gain a certain specialization.

For the fall semester, the department offers a total of 72 courses, summing up to 211 teaching periods for lectures and recitations. As shown in Table 1(Panels A and B) the 72 courses are split unevenly among the five years of study. In addition, 26 of these courses require certain training hours of the students in

Table 1
Number of courses offered for the students of the first, second, and third year (Panel A) and for the students of the fourth and fifth year (Panel B)

	Student year			Total
	First	Second	Third	
Panel A				
# of courses	10 (six mandatory, four electives)	8	7	25
Panel B				
Division	Fourth	Fifth		
A	4	8		12
B	6	4		10
C	6	4		10
D	7	8		15
Total	23	24		47

laboratories, summing up to 110 periods for lab work. The department assigns these courses to 55 professors and lecturers. There is an additional pool of assistants that take responsibility of specific tasks, mainly supervision of the students during training in the laboratories.

The department disposes six regular classrooms (suitable for lectures and recitations) and 12 specialized classrooms (suitable for lab work). The regular classrooms belong to the department and are available for use at any time during the week. This is not always the case with the other departments and sharing of a classroom with a pre-agreed availability schedule for each one of them is the current practice. Classrooms have variable capacities and are suitable either for large or for small audiences. Respectively, most of the lab rooms are specialized and are utilized for a single course, however computer rooms are shared between courses of the department. Moreover, a few courses share specialized rooms (e.g. rooms with tables for practice in design) with courses from other departments and in this case the timetabler must schedule the involved courses only during pre-agreed time periods.

An IP model based on the formulation described in this paper was developed for the construction of the fall semester timetable. The cost coefficients were assigned values according to the strategy discussed in Section 5, using functions similar to the one in Fig. 2, but possibly different for different courses and days. The mixed integer programming (MIP) solver by CPLEX 5.1 was used in an HP J7000 Workstation and the solution found is a proposed timetable for the department. For presentation purposes, we selectively present the timetables for the first and fourth years.

8.1. Discussion on the resulting timetables

The first year students have six mandatory and four elective courses for the fall semester. The course names, code numbers, structure (lectures/recitations/lab work), and required split for the assigned time

Table 2
The curriculum and timetable for the fall semester of 1st year students

No	Courses	Code	Periods			
			Lect.	Reci.	Required split in sessions	Lab/#groups
1	Mathematics I	AF1	3	2	2 + 2 + 1	–
2	Physics I	AF2	3	2	3 + 2	2/3
3	Introduction to computers I	AF3	2	1	3	2/3
4	Linear algebra	AF4	2	1	2 + 1	–
5	Introduction to digital logic	AF5	2	1	3	–
6	Engineering drawing	AF6	–	–	–	4/2
	Electives					
7	Philosophy I	AF7	3	–	3	–
8	History of Greek nation I	AF8	3	–	2 + 1	–
9	New Greek literature I	AF9	3	–	3	–
10	Foreign language I	AF10	3	–	1 + 1 + 1	–

<table>
<tr><th>Periods / DAYS</th><th colspan="13">First Year (fall semester)</th></tr>
<tr><th></th><th>PER1</th><th>PER 2</th><th>PER3</th><th>PER4</th><th>PER5</th><th>PER6</th><th>PER7</th><th>PER8</th><th>PER9</th><th>PER10</th><th>PER11</th><th>PER12</th><th>PER13</th></tr>
<tr><td>Monday</td><td>AF10
Rm 0</td><td colspan="3">AF2
Rm 0</td><td colspan="2">AF1
Rm 0</td><td colspan="4">AF6 (G1)
LR 8</td><td>AF8
Rm 0</td><td></td><td></td></tr>
<tr><td>Tuesday</td><td colspan="2">AF8
Rm 0</td><td colspan="2">AF1
Rm 0</td><td colspan="3">AF5
Rm 0</td><td colspan="4">AF6 (G2)
LR 8</td><td>AF4
Rm 0</td><td></td></tr>
<tr><td>Wednesday</td><td colspan="3">AF3
Rm 0</td><td>AF10
Rm 0</td><td colspan="2">AF2 (G1)
LR 3</td><td colspan="2">AF3 (G1)
LR 4</td><td colspan="2">AF2 (G2)
LR 3</td><td></td><td>AF1
Rm 0</td><td></td></tr>
<tr><td>Thursday</td><td>AF10
Rm 0</td><td colspan="2">AF4
Rm 0</td><td colspan="3">AF7
Rm 0</td><td colspan="2">AF3 (G2)
LR 4</td><td></td><td colspan="2">AF2
Rm 0</td><td></td><td></td></tr>
<tr><td>Friday</td><td colspan="3">AF9
Rm 0</td><td colspan="2">AF2 (G3)
LR 3</td><td colspan="2">AF3 (G3)
LR 4</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>

periods, as well as the # of groups required for the lab work are all presented in Table 2. Judging the resulting timetable for the first year, one should note the following:

(a) All courses required by the curriculum appear in the timetable with no conflicts among them. Since all students should have the chance to choose two of the elective courses, there should be absolutely no overlapping between courses and this is reflected in the presented solution.
(b) Each course appears in so many periods as required from the curriculum, assigning lectures and recitations in regular classrooms (indicated as Rm#) and lab work in specialized rooms (indicated as LR#). Also the session length follows the requirements for each course. The lab sessions are scheduled in a repetitive manner to handle the total number of groups required by the corresponding course.
(c) All lectures and recitations for the mandatory and elective courses are scheduled during morning sessions (8:00–15:00) or afternoon sessions (17:00–20:00), following the preferences for certain time periods given as input from the timetabler.

Table 3
Curriculum for the fourth year for all divisions of the department

No	Courses	Code	Periods			
			Lect.	Reci.	Required split in sessions	Lab/#group
Division of Electric Power Systems—fall semester						
1	Power systems analysis I	DFA1	2	1	2 + 1	3/1
2	High voltages	DFA2	3		3	–
3	Power electronics I	DFA3	3		3	3/1
4	Elecrical installations	DFA4	3		3	–
Division of Electronics & Computers—fall semester						
1	Algorithms & data structures	DFB1	2	1	2 + 1	2/1
2	Advanced programming techniques	DFB2	2	1	2 + 1	1/1
3	Microprocessors & microsystems I	DFB3	2	1	2 + 1	3/2
4	Advanced analog/digital circuits	DFB4	2	1	2 + 1	–
5	VLSI design I	DFB5	2	1	2 + 1	–
6	Digital signal processing	DFB6	2	1	2 + 1	–
Division of Systems Automatic Control—fall semester						
1	Lab. for analogue and digital control	DFC1	–	–	–	3/1
2	Design of dynamic systems I	DFC2	3	–	2 + 1	–
3	Industrial automation I	DFC3	3	-	2 + 1	–
4	Applied optimization	DFC4	3	–	3	–
5	Industrial information systems	DFC5	3	–	3	–
6	Applied computational methods	DFC6	3	3	3	–
Division of Telecommunications & Information Technology—fall semester						
1	Telecommunications systems I	DFD1	2	1	2 + 1	3/1
2	Information theory	DFD2	3	–	2 + 1	–
3	Telephone systems I	DFD3	3	–	2 + 1	–
4	Information systems	DFD4	2	1	2 + 1	–
5	Wave propagation and antenna design	DFD5	2	1	2 + 1	3/1
6	Artificial intelligence	DFD6	2	1	2 + 1	–
6	Physics of photovoltaic cells	DFD7	2	1	2 + 1	–

(d) On the contrary, lab work is scheduled around noontime, for various reasons that the timetabler thought would be beneficial to the students.
(e) Due to the assignment of suitable values to cost coefficients in the objective function it is possible to satisfy the preference given to classroom Rm0, which was named by the timetabler to be the designated room for the first year students. Thus the objective for minimal classroom changes is fully satisfied.
(f) The elective courses (indicated by the darker coloured boxes in the timetable) are scheduled at the beginning of the day or after all mandatory lectures have come to an end so that schedules for individual students are more compact.

Similarly, in Tables 3 and 4, one may observe the curriculum and the timetable for the students of the fourth year. Many comments that were put forth in the previous discussion could be repeated for this timetable too. The main difference here is the distinction between students of different divisions. In fact, each division provides a separate timetable for its own students, with the exception of a number of courses shared between divisions. As a result, overlapping (in time) between courses of different divisions is allowed in general; however, certain courses are offered by one division and are recommended to students of other divisions. In those cases there should be no overlapping with any of the courses of either division. This fact is reflected in the timetable presented with courses like DFB3 (offered by division B and attended from students of the divisions B and C) or DFA1 (offered by division A and attended from students of the divisions A and C).

Table 4
Timetables for the fourth year students

		Fourth Year (fall semester)												
	Division	PER1	PER 2	PER3	PER4	PER5	PER6	PER7	PER8	PER9	PER10	PER11	PER12	PER13
MONDAY	A				DFA4 Rm 3									
	B		DFB2 LR 2	DFB1 LR 2				DFB3 Rm 3						
	C	DFC5 Rm 3				DFD1 Rm 4		DFB3 Rm 3		DFC2 Rm 4				
	D	DFD7 Rm 4	DFD5 Rm 4		DFD4 Rm 4	DFD1 Rm 4		DFD3 Rm 4						
TUESDAY	A					DFA1 Rm 3								
	B	DFB5 Rm 3		DFB2 Rm 3				DFB3 Rm 3						
	C	DFC1 LR 4				DFA1 Rm 3		DFB3 Rm 3	DFD1 LR 8					
	D	DFD7 Rm 4		DFD5 Rm 4	DFD3 Rm 4				DFD1 LR 8					
WED / DAY	A	DFA1 LR 7				DFA2 Rm 3								
	B	DFB4 Rm 3		DFB1 Rm 3										
	C	DFA1 LR 7			DFD2 Rm 4	DFD6 Rm 4		DFD1 Rm 4	DFC4 Rm 4					
	D	DFD4 Rm 4			DFD2 Rm 4	DFD6 Rm 4		DFD1 Rm 4						
THURSDAY	A	DFA3 Rm 3												
	B					DFB6 Rm 3		DFB1 Rm 3	DFB5 Rm 3	DFB4 Rm 3	DFB3 (G1) LR 5			
	C	DFD2 Rm 4		DFD6 Rm 4		DFC3 Rm 4		DFC6 Rm 4			DFB3 (G1) LR 5			
	D	DFD2 Rm 4		DFD6 Rm 4										
FRI DAY	A	DFA1 Rm 3	DFA3 LR 3											
	B	DFB6 Rm 4	DFB2 Rm 3	DFB3 (G2) LR 5										
	C	DFA1 Rm 3		DFB3 (G2) LR 5			DFC3 Rm 4	DFC2 Rm 4						
	D				DFD5 LR 9									

Table 5
Size of problems solved and resulting models

	Problem size				Model size		
	No. of courses	No. of lab courses	Required teach. per.	No. of rooms/labs	No. of rows	No. of columns	No. of non-zeros
Problem #1	25	8	139	3/6	7,543	4,100	35,685
Problem #2	47	19	187	4/10	12,734	13,527	78,523
Problem #3	92	27	326	6/12	17,159	19,295	92,358

8.2. Computational results

In order to evaluate the proposed IP model, three problems of different size were solved and are exposed in Table 5. The number of the courses varied from 25 to 92 in addition to the lab courses that varied from 8 to 27, totaling the requirements for teaching periods from 139 to 326. We should note that these teaching periods are scheduled within the 70 available time periods during each week.

The models that resulted following the suggested IP formulation carried 7,543–17,159 equations and 4,100–19,295 binary variables, while the non-zeros of the IP model varied from 35,685 to 92,358. Computing the optimal timetables required 2.5 minutes for the first problem, 18.5 minutes for the second and 95 minutes for the last one.

For large problems like problem #3, it is almost always possible to break the problem into smaller ones, at least for our engineering school. This is because there is no real competition of the lower-grade students with those of the higher grades for teaching rooms. Lower-grade students are large audiences and require auditoriums and higher-grade students are usually small audiences and request small classrooms.

Lastly, it is important to mention the role of the values chosen for the cost coefficients. According to our experience, by changing the penalty functions it is possible to change computation time by a large factor, meaning that the optimization process may be guided faster to the optimal solution.

9. Summary and conclusions

In this paper we presented a new IP formulation of a timetabling problem as it appears in many universities, adding however many features that may be distinct in Engineering Schools of Greek universities.

The problem is a hard one and very complex, however, the choices made through the modeling process result to solvable and flexible models. The flexibility offered is due to the multi-dimensional variables, which allow low details of the educational system to be modeled as constraints of the IP model. A variety of rules may be represented in the model with suitable constraints provided by this formulation. Moreover, the choice made for the cost function allows the introduction of certain preferences regarding time periods, days and classrooms, so that timetables can be improved according to well-accepted quality measures.

The timetable for the Electrical and Computer Engineering Department in our university was used as a case study and was solved very successfully. The timetable construction required scheduling of lectures, recitations and lab courses, each type carrying different characteristics and requests. Sessions with consecutive time periods and/or repetitions of the same course in different sessions are among the rules that require satisfaction and have turned into hard constraints.

Creating timetables for academic institutions is a tedious process, however automation is now possible.

References

[1] E.A. Akkoyunlu, A linear algorithm for computing the optimum university timetable, The Computer Journal 16 (4) (1973) 347–350.
[2] J. Aubin, J.A. Ferland, A large scale timetabling problem, Computers and Operational Research 18 (1989) 67–77.
[3] M.A. Badri, D.L. Davis, D.F. Davis, J. Hollingsworth, A multi-objective course scheduling model: Combining faculty preferences for courses and times, Computers and Operations Research 25 (4) (1998) 303–316.
[4] A.V. Bardadym, Computer-aided school and university timetabling: The new wave, in: E. Burke, P. Ross (Eds.), Lecture Notes in Computer Science, vol. 1153, Springer-Verlag, 1995, pp. 22–45.
[5] E.K. Burke, D.G. Elliman, R.F. Weare, A genetic algorithm for university timetabling, in: AISB Workshop on Evolutionary Computing, University of Leeds, UK, April 1994, Society for the Study of Artificial Intelligence and Simulation of Behaviour.
[6] E.K. Burke, D.G. Elliman, R.F. Weare, A genetic algorithm based university timetabling system, in: 22nd East–West International Conference on Computer Technologies in Education, Crimea, Ukraine, September 1994, vol. 1, pp. 35–40.
[7] T. Birbas, S. Daskalaki, E. Housos, Timetabling for Greek high schools, Journal of Operational Research Society 48 (1997) 1191–1200.
[8] T. Birbas, S. Daskalaki, E. Housos, Course and teacher scheduling in Hellenic high schools, in: 4th Balkan Conference on Operational Research, Thessaloniki, Greece, October 1997.
[9] J.A. Breslaw, A linear programming solution to the faculty assignment problem, Socio-Economic Planning Science 10 (1976) 227–230.
[10] M.P. Carrasco, M.V. Rato, A multiobjective genetic algorithm for the class/teacher timetabling problem, in: E. Burke, W. Erben (Eds.), Practice and Theory of Timetabling III, Lecture Notes in Computer Science, vol. 2079, Springer-Verlag, 2001, pp. 3–17.
[11] M.W. Carter, G. Laporte, Recent developments in practical course scheduling, in: E. Burke, M. Carter (Eds.), Practice and Theory of Timetabling II, Springer-Verlag, 1998, pp. 3–19.
[12] D. Costa, A tabu search algorithm for computing an operational timetable, European Journal of Operational Research 76 (1994) 98–110.
[13] S.B. Deris, S. Omatu, H. Ohta, P. Samat, University timetabling by constraint-based reasoning: A case study, Journal of the Operational Research Society 48 (1997) 1178–1190.
[14] M. Dimopoulou, P. Miliotis, Implementation of a university course and examination timetabling system, European Journal of Operational Research 130 (2001) 202–213.
[15] J.J. Dinkel, J. Mote, M.A. Venkataramanan, An efficient decision support system for academic course scheduling, Operations Research 37 (6) (1989) 853–864.
[16] A. Drexl, F. Salewski, Distribution requirements and compactness constraints in school timetabling, European Journal of Operational Research 102 (1997) 193–214.
[17] H.M.M. ten Eikelder, R.J. Willemen, Some complexity aspects of secondary school timetabling problems, in: E. Burke, W. Erben (Eds.), Practice and Theory of Timetabling III, Lecture Notes in Computer Science, vol. 2079, Springer-Verlag, 2001, pp. 3–17.
[18] H.A. Eiselt, G. Laporte, Combinatorial optimization problems with soft and hard requirements, Journal of Operational Research Society 38 (9) (1987) 785–795.
[19] J. Ferland, S. Roy, Timetabling problem for university as assignment of activities to resources, Computers and Operations Research 12 (2) (1985) 207–218.
[20] K. Gosselin, M. Truchon, Allocation of classrooms by linear programming, Journal of Operational Research Society 37 (6) (1986) 561–569.
[21] A. Hertz, Find a feasible course schedule using tabu search, Discrete Applied Mathematics 35 (1992) 255–270.
[22] T.H. Hultberg, D.M. Cardoso, The teacher assignment problem: A special case of the fixed charge transportation problem, European Journal of Operational Research 101 (1997) 463–473.
[23] E.L. Johnson, G.L. Nemhauser, M.W.P. Savelsbergh, Progress in linear programming based branch-and-bound algorithms: An exposition, INFORMS Journal on Computing 12 (2000).
[24] L. Kang, G.M. White, A logic approach to the resolution of constraints in timetabling, European Journal of Operational Research 61 (1992) 306–317.
[25] N.L. Lawrie, An integer linear programming model of a school timetabling problem, The Computer Journal 12 (1969) 307–316.
[26] R.H. McClure, C.E. Wells, A mathematical programming model for faculty course assignment, Decision Science 15 (1984) 409–420.
[27] B. Paechter, Optimising a presentation timetable using evolutionary algorithms, in: AISB Workshop on Evolutionary Computation, Lecture Notes in Computer Science, vol. 865, 1994, pp. 264–276.
[28] B. Paechter, A. Cumming, H. Luchian, M. Petriuc, Two solutions to the general timetable problem using evolutionary methods, in: The Proceedings of the IEEE Conference on Evolutionary Computing, 1994.
[29] A. Schaerf, A survey of automated timetabling, Artificial Intelligence Review 13 (2) (1999) 87–127.
[30] G. Schmidt, T. Strohlein, Timetable construction—An annotated bibliography, The Computer Journal 23 (4) (1979) 307–316.

[31] A. Tripathy, School timetabling—A case in large binary integer linear programming, Management Science 30 (12) (1984) 1473–1489.
[32] A. Tripathy, Computerized decision aid for timetabling—A case analysis, Discrete Applied Mathematics 35 (3) (1992) 313–323.
[33] D. de Werra, An introduction to timetabling, European Journal of Operational Research 19 (1985) 151–162.
[34] D. de Werra, The combinatorics of timetabling, European Journal of Operational Research 96 (1997) 504–513.
[35] D.J.A Welsh, M.B. Powell, An upper bound to the chromatic number of a graph and its application to timetabling problem, The Computer Journal 10 (1967) 85–86.