Research Article

Linear Integer Model for the Course Timetabling Problem of a Faculty in Rio de Janeiro

Valdecy Pereira and Helder Gomes Costa

Department of Production Engineering, Fluminense Federal University, Rua Passo da Pátria 156, São Domingos, 24.210-240 Niterói, RJ, Brazil

Correspondence should be addressed to Valdecy Pereira; valdecypereira@yahoo.com.br

Received 5 September 2015; Revised 28 December 2015; Accepted 30 December 2015

Academic Editor: Ahmed Ghoniem

Copyright © 2016 V. Pereira and H. Gomes Costa. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This work presents a linear integer programming model that solves a timetabling problem of a faculty in Rio de Janeiro, Brazil. The model was designed to generate solutions that meet the preferences of the faculty's managers, namely, allocating the maximum number of lecturers with highest academic title and minimising costs by merging courses with equivalent syllabuses. The integer linear model also finds solutions that meet lecturers' scheduling preferences, thereby generating more practical and comfortable schedules for these professionals. Preferences were represented in the objective function, each with a specific weight. The model outperformed manual solutions in terms of response time and quality. The model was also able to demonstrate that lecturers' scheduling preferences are actually conflicting goals. The model was approved by the faculty's managers and has been used since the second semester of 2011.

1. Introduction

Tripathy [1] defines university course timetabling (UCT) as a programming problem that allocates a given number of meetings between students and lecturers for a given period. Each of these meetings is attended by specific groups of students and one lecturer. Each meeting may also require specific resources (projectors, computers, etc.). The allocation choices must take into consideration the availability of lecturers, resources, and, if necessary, other factors. Schaerf [2] defines UCT as scheduling a sequence of teaching sessions between lecturers and students in a predetermined period of time, normally within a week, while satisfying a set of constraints. The timetable may be created manually, often laboriously, and the solution thereby obtained is generally unsatisfactory in some sense. de Werra [3] believes that, because UCT problems can be very different, it seems unreasonable to develop universal programming models that might be used in all institutions.

The objective function of these problems usually consists of minimising costs and/or maximising preferences; that is, the objective function measures both the quality of solutions

and the validity of the generated timetables [4]. The constraints of the UCT problems are the core of these models and are defined by the true characteristics of each institution.

Constraints are generally divided into two categories, hard and soft constraints [5]. A solution to timetable programming that violates even a single hard constraint is not feasible, whereas a solution that violates any number of soft constraints remains feasible. Soft constraints are used to improve the solution's quality because satisfying one of them means the solution meets some particular preference; therefore, a feasible solution can be found satisfying or not a soft constraint. For instance, a lecturer is available to give lessons only on Mondays in the morning, afternoon, or night shifts, but he prefers the morning period. A schedule with lessons on any other given day, than Monday, is considered unfeasible, so we can conclude that this is a hard constraint. In the other hand a schedule with a lesson on Mondays in the afternoon or night shifts is considered feasible, and a schedule with a lesson on Mondays in the morning shift is considered feasible and satisfies the lecturer preference, therefore improving the quality of the schedule, so we can conclude that this is a soft constraint [6].

This work aims to propose a model that solves the UCT problem in a faculty of Rio de Janeiro. This paper is organized as follows. Section 2 provides a literature review to present the problem of using UCT programming for solving real-world cases. Section 3 details the characteristics of the educational institution in this study, and Section 4 details the modelling to solve the corresponding problems. Section 5 shows how the methodology was applied, and, lastly, Section 6 contains conclusions and a discussion of the main issues arising.

2. Literature Review

The main goal of this literature review is to identify differences and similarities between the various models that have been developed. If each institution possesses unique characteristics, it is expected that their respective mathematical models will also possess unique characteristics that may be represented by hard or soft constraints or even by the objective function itself, when applicable.

Daskalaki et al. [7] developed a binary linear integer programming model to which several constraints can be added to incorporate a large number of operational requirements and rules, as occurs in most academic institutions. The model was used in the Department of Electrical and Computer Engineering of the University of Patras in Greece. The objective of this model is to minimise a linear cost function while satisfying preferences related to the time, day of week, or even classroom for a specified course. The cost function consists of two terms: the first represents the cost of allocating a module (lecture) to a given time of a teaching session; the second represents the cost incurred in allocating these modules to a given day of the week. Hard constraints are as follows: modules are not allowed to overlap; all planned modules for a given semester should be offered; and the teaching session duration should meet lecturers' demands and should occur in the required rooms. Soft constraints are as follows: collisions in lecturers' availability schedules are not permitted; the timetable should be as compact as possible; and modules should be restricted to a single classroom. The model was used in a case study, and the authors reported that a feasible solution was generated.

Thompson [8] investigated the construction of a UCT with information on students' preferences on course allocation. The objective function measures the solution's quality based on how it satisfies students' preferences. The model was tested with 3 data sets of the School of Hotel Administration of Cornell University in the USA. Its objective function consisted of the following: minimising the number of modules allocated to lecturers; minimising the number of modules not in the timetable; minimising collisions of modules in time; minimising violations regarding modules that should be scheduled in the same days; and minimising cases in which modules should be scheduled for the same time but are not. Its hard constraints are the following: lecturers schedules are not allowed to collide; the distance between the location of one class and the next should be as small as possible; a lecturer must have his or her work hours fully allocated; classroom schedules are not allowed to collide; and module schedules are not allowed to collide. Soft constraints are as follows: some

modules should not be offered on the same days as some other modules, while some modules should be offered on the same day as some other modules. The author believes that his model is applicable to other institutions.

Schimmelpfeng and Helber [9] described a binary integer programming approach that was used in the School of Economics and Administration of the University of Hannover in Germany. The model solves the UCT problem and allocates classrooms. Its objective function consists of minimising violations of each one of the problem's soft constraints. Hard constraints are as follows: lecturers' availability schedules must be respected; collisions in lecturers' availability schedules are not allowed, nor are collisions between modules; and modules' sequences must be respected. Soft constraints are as follows: some classes' scheduling preferences must be respected, and some lecturers' scheduling preferences must be respected. After applying the model, the authors conducted a survey to assess the model's results from the lecturers' perspective. Only one of the 100 respondents was dissatisfied with the new system.

Bakir and Aksop [10] formulated a binary integer model for timetable programming in the Department of Statistics of Gazi University in Turkey. The model aims to minimise the dissatisfaction of students and lecturers, and thus the objective function assesses these quantities. Hard constraints are as follows: modules and classrooms are not allowed to collide; students may only enrol in courses corresponding to their current semester; some courses must obey a logical sequence in time; and teaching sessions allocated to modules must respect the required duration of each module. Soft constraints are as follows: the timetable should be as compact as possible, and failing students may only enrol in courses from their current and previous semesters, if possible. The authors claim the optimal solution was found; however, the size of the problem required a very long time to find this solution.

Adewumi et al. [11] developed a genetic algorithm to address a UCT in an unidentified university in Nigeria. The hard constraints are that lecturers' and classrooms' schedules are not allowed to collide and classrooms' capacities must be respected. The authors claim that the heuristic found good results and that their model may easily be adapted to satisfy soft constraints, if necessary.

Razak et al. [12] present an approach that uses a bipartite graph and graph colouring to solve a UCT problem for the Department of Information Technology of a private university. The hard constraints are that modules and lecturers' availabilities are not allowed to collide. The soft constraints are as follows: the timetable should accommodate lunch breaks if possible; teaching sessions should have a limited maximum duration; the total number of modules should not exceed the lecturers' available working hours; lecturers should not teach courses for more than three consecutive hours; students should not attend classes for more than four consecutive hours; older students should have preferential schedules; a class of students should not have only one course in any given day; and a module should not be taught for more than two consecutive hours. The authors reported that the results of experimental studies demonstrated that the

proposed model solves UCT problems more easily than other methods.

Al-Tarawneh and Ayob [13] applied a tabu search and a multineighbourhood structure to solve the UCT problem of the School of Engineering of Kebangsaan University in Malaysia. The objective function sums the weights of penalties associated with the model's soft constraints, which were determined by specialists. The hard constraints are as follows: collisions between classrooms and in lecturers' availability schedules are not allowed; classroom capacities must be respected; and modules that take longer than two hours cannot be split into more than one teaching session. The soft constraints are as follows: each class of students must have one module in isolation every day; no class of students may have two classes in sequence; intervals between modules must not be long; no lecturer should teach two consecutive sessions; and no female lecturer should be allocated for modules in the evening. The authors claim that the results they obtained are better than those obtained manually.

This literature review shows that the core of hard constraints is quite coherent in the analysed works because these are, in fact, logical constraints that cannot be left unsatisfied. By contrast, the set of soft constraints may be viewed as a type of "institutional fingerprint" that makes the modelling extremely specific for each case and thus cannot be generalised.

3. Description of the Institution to Be Analysed

The private university that is the object of study of this work is located in São Gonçalo, state of Rio de Janeiro, Brazil. The period of analysis is the year 2011 because this was the period for which data were fully available to the authors. Data gathering on the institution was all-encompassing, and the main sources were the institution page on Internet, institutional Intranet, documental research, and key staff interviews. In the first semester of 2011, the institution had a total of 1,297 students distributed over nine undergraduate courses and 92 lecturers. Its infrastructure consists of 47 classrooms and five laboratories, which seat a total of 1,857 students. There are two distinct shifts for classes, namely, morning and evening shifts. The morning shift is divided into two teaching sessions: the first session starts at 7:20 a.m. and ends at 9:00 a.m.; after a 20-minute interval, the second session starts at 9:20 a.m. and ends at 10:20 a.m. The evening shift also has two teaching sessions. The first starts at 7:00 p.m. and ends at 8:20 p.m.; after a 20-minute break, the second session starts at 8:40 p.m. and ends at 10:00 p.m.

Because each teaching shift is split into two teaching sessions, modules are classified into two categories: double-session (DS) and single-session (SS) modules. The first corresponds to modules that comprise two teaching sessions (from 7:20 a.m. to 10:20 a.m. or from 7:00 p.m. to 10:00 p.m.), and the second corresponds to modules that comprise only one teaching session (from 7:20 a.m. to 9:00 a.m.; from 9:20 a.m. to 10:20 a.m.; from 7:00 p.m. to 8:20 p.m.; or from 8:40 p.m. to 10 p.m.). Each module has strictly one class per week,

meaning a regular student may be enrolled in five to 10 modules in a single shift.

Some modules may have joint classes; that is, students from different undergraduate programs may attend the same class because the modules have the same syllabuses; Foulds and Johnson [14] call this practice *piggybacking*.

The institution offers nine undergraduate programmes, with different durations each: Administration (ADM) in eight semesters, Law School (LAW) in ten semesters, Information Systems (INF) in eight semesters, Pedagogy (PED) in six semesters, Languages (LAN) in six semesters, Tourism (TOU) in seven semesters, Logistics Technology (LOG) in five semesters, Quality Management Technology (QUA) in five semesters, and Financial Management (FIN) Technology in five semesters.

The technology group is a new step in Brazil's higher education system towards professional education. These undergraduate courses grant technology degrees to students that want to fulfil specific demands of the job market and can be rapidly allocated as specialised professionals. The "technologist" title is granted upon the completion of these courses, which are authorised by the Brazilian Ministry of Education.

Table 1 shows the total distribution of students among the 9 undergraduate programmes in the first semester of 2011.

There are 17 receiver modules in total, that is, modules that have the same syllabuses as modules from other undergraduate programmes and that have a larger number of students and are thus able to "receive" student classes from other programmes. This set of receiver modules can hold a total of 29 joint modules, that is, modules with matching syllabuses but with fewer students that cannot, therefore, "receive" other students. The institution prefers that all such fusions take place (if the required physical space is available). In total, for planning the second semester of 2011, there were 302 modules to be offered, of which 198 were DS and 104 were SS. Of these, 73 required new hires, as there were no or insufficient lecturers on staff qualified to teach their contents. Further hiring was needed due to losses in staff from 92 lecturers in the first semester of 2011 to 77 in the second semester of the same year.

Having ascertained the institution's characteristics and the planned offer of modules for the semester, the following preferential needs for the timetable were identified: to aggregate modules with the same syllabuses (*piggybacking*) and preferably allocate as many modules as possible to lecturers with higher academic titles. This latter preference is a strategic alignment of the institution to increase the quality of its education services and to foster research groups within the faculty.

Two other preferences arose from lecturers' schedule preferences: avoiding SS class dispersion (thus also avoiding gaps in lecturer's schedules) and, should there be any gaps, scheduling modules in the first teaching session so that lecturers may leave earlier. The total time estimated by the coordinators of each course to manually solve this problem was approximately two months.

4. Problem Modelling

The modelling developed herein considered only the planning of modules offered in the second semester of 2011

TABLE 1: Student distribution.

2011.1	ADM	LAW	INF	PED	LAN	TOU	LOG	QUA	FIN	Total
Students	711	161	123	93	78	76	30	13	12	1297
%	54.82%	12.41%	9.48%	7.17%	6.01%	5.86%	2.31%	1.00%	0.93%	100%

in the evening shift because the morning shift has a minimal amount of students and therefore did not require mathematical modelling. A total of 77 lecturers were to be allocated to 302 different modules distributed over the nine undergraduate programmes.

The set of available lecturers will be called I , and each element of this set will be referred to as i . Within I there are lecturers that can teach SS modules in the first and/or second teaching session. The lecturers that teach modules in the first teaching session are form the I_g subset, and those who teach modules in the second teaching session are form the I_h subset.

The set of planned modules will be called K , and each element in this set will be referred to as k . Within K there are receiving and joint modules, which taken together form subsets K_r and K_c , respectively.

4.1. Objective Function. The objective of the proposed modelling is to maximise the allocation of lecturers as a function of a weight or score attributed to each of them. This score may be generated via one or more criteria; in the present work, only one criterion was used: academic title. This is a general indicator of the benefit of allocating a given lecturer to a given module: the higher the score, the greater the chance of this lecturer being allocated. Should the lecturer be unavailable on some day of the week, this day will be penalised with a negative value sufficiently large that the model avoids allocating him outside of his availability. The objective function is represented by the following:

$$\max \sum_i \sum_j \sum_k \sum_l \sum_m \sum_n \sum_o x_{i-j.k.l.m.n.o} \cdot T_{ij} \cdot \alpha_i, \quad (1)$$

where i is elements of I which represents all available lecturers for the second semester of 2011. j represents the days of the week in which modules are taught, with values 1 through 5 corresponding to Monday through Friday. k is elements of K which represents all available modules for the second semester of 2011. l indicates the class period to which a class/lecturer is allocated: if allocated to the first teaching session, then $l = 1$; if allocated to the second, then $l = 2$; if allocated to both, then $l = 3$. m indicates the semester within undergraduate program n . n indicates the degree program, where Administration = 1, Law = 2, Language = 3, Pedagogy = 4, Tourism = 5, Information Systems = 6, Logistics Technology = 7, Quality Management Technology = 8, and Financial Management Technology = 9. o indicates the class in semester m of undergraduate program n . $x_{i-j.k.l.m.n.o}$ is a binary decision variable indicating which lecturer was allocated; the day of allocation; the module to be taught; the teaching session; the course semester; the undergraduate program; and the class. Value 1 means that the allocation was made, while 0 means that it was not. These variables are defined in this work as binary allocation variables. T_{ij} is a binary parameter that denotes the day(s) j in which lecturer i is available. Value

1 indicates availability, while 0 indicates unavailability. α_i is weight or score attributed to each lecturer i ; this weight or score may be the result of 1 or more criteria, and the higher its value, the greater the chances of lecturer i being allocated.

The objective function is constructed in such a way that every time a lecturer is allocated on one of his or her available days, he or she contributes with an increment in points to the objective function; the higher his title is, the larger this increment is. The objective function is decreased (penalty) if the lecturer is allocated to some day on which he or she is not available (the penalty is the same regardless of his title).

4.2. First Group of Hard Constraints: Completeness Constraints. The problem's first group of constraints is of the hard type and imposes that the time table for all undergraduate programs be completed (completeness constraint), that is, once the planned course semesters and modules for a given calendar semester are defined, all of these planned modules should be offered:

$$\sum_i \sum_j x_{i-j.k.3.m.n.o} = 1, \quad (2)$$

$\forall k, \forall m, \forall n, \forall o,$

$$\sum_i \sum_j x_{i-j.k.1.m.n.o} + \sum_i \sum_j x_{i-j.k.2.m.n.o} = 1, \quad (3)$$

$\forall k, \forall m, \forall n, \forall o.$

Equation (2) imposes that all planned DS modules pertaining to a term m from program n and class o should be offered on a weekday j by a lecturer i . Equation (3) imposes the same constraint for all planned SS modules, which must be offered either in the first or in the second session.

4.3. Second Group of Hard Constraints: Collisions between Courses. The second group of constraints is of the hard type and imposes that two modules of a single term m from undergraduate program n and class o cannot be offered on the same weekday j . This constraint avoids collisions between courses. Consider

$$\sum_i x_{i-j.k.1.m.n.o} + \sum_i x_{i-j.k.3.m.n.o} \leq 1, \quad (4)$$

$\forall j, \forall k, \forall m, \forall n, \forall o,$

$$\sum_i x_{i-j.k.2.m.n.o} + \sum_i x_{i-j.k.3.m.n.o} \leq 1, \quad (5)$$

$\forall j, \forall k, \forall m, \forall n, \forall o,$

$$\sum_i x_{i-j.k.1.m.n.o} + \sum_i x_{i-j.k.2.m.n.o} \leq 1, \quad (6)$$

$\forall j, \forall k, \forall m, \forall n, \forall o.$

Constraints (4) and (5) guarantee that any DS module will not be offered on the same weekday j as any SS module. Constraint (6) guarantees that the same SS module will not be offered in both the first and the second teaching sessions.

4.4. Third Group of Hard Constraints: Collisions in Lecturers' Schedules. The third and last group of constraints, which is also of the hard type, imposes that it is not possible for a lecturer i to teach k different modules at the same time on the same weekday. This constraint avoids collisions in lecturer's schedules. Consider

$$\sum_i \sum_m \sum_n \sum_o x_{i-j,k,1,m,n,o} + \sum_i \sum_m \sum_n \sum_o x_{i-j,k,3,m,n,o} \leq 1, \quad (7)$$

$\forall j, \forall k,$

$$\sum_i \sum_m \sum_n \sum_o x_{i-j,k,2,m,n,o} + \sum_i \sum_m \sum_n \sum_o x_{i-j,k,3,m,n,o} \leq 1, \quad (8)$$

$\forall j, \forall k,$

$$\sum_i \sum_m \sum_n \sum_o x_{i-j,k,1,m,n,o} + \sum_i \sum_m \sum_n \sum_o x_{i-j,k,2,m,n,o} \leq 1, \quad (9)$$

$\forall j, \forall k.$

Constraints (7) and (8) ensure that, on day j , some DS module of any semester m of any undergraduate program n of any class o will be taught by some lecturer i along with any SS module. Constraint (9) ensures that 2 SS modules of any semester m of any undergraduate program n and any class o will be taught by some lecturer i at the same time. One might add to these another constraint preventing lecturers from having a null workload, but such a constraint might render the problem unfeasible.

To ensure that the model and its variations generate solutions that are always feasible, an artefact must be introduced; namely, an element is added to I that has full availability in its schedule, cannot suffer from scheduling collisions, and is allowed to "teach" any course. The element is called HIRING NEW LECTURER or HNL. Table 2 exemplifies this artefact:

Assume that five courses must be offered in a given week, labelled as C01 through C05. Table 2 gives the lecturer's availability and the modules that he/she is qualified to teach. Lecturer T01 is available every day of the week and can teach courses C01 and C02; lecturer T02 is available every day except Friday and can teach courses C03 and C04; lecturer T03 is available only Mondays and Wednesdays and can teach courses C01 and C03. Element HNL is available every weekday and can "teach" any course.

The weight corresponding to this element in the weight vector should be negative: as this is a maximisation model, element HNL should be the last option of allocation because it brings the objective function's total value down. This weight should, however, inflict a lighter penalty than allocating a lecturer outside of his days of availability, thus making it evident that, if no adjustment can be made, a new lecturer must be hired.

4.5. First Group of Soft Constraints. In the first group of soft constraints or the S1 category there are the preferences joining classes from different undergraduate programmes whose modules share the same syllabus. For this group, having two or more modules sharing the same lecturer, classroom, and time (*piggybacking*) is desirable. This preference requires adding the following term to the objective function:

$$\beta \cdot \sum_{k_r} \sum_{k_c} S1_{k_r,k_c}, \quad (10)$$

where k_r represents receiver courses belonging to subset K_r . k_c represents joint courses belonging to subset K_c . $S1_{k_r,k_c}$ are binary variables for which value 1 means that two courses were joined and value 0 means otherwise. Each binary variable is connected by a (k_r, k_c) pair. β is value by which the objective function should be increased when $S1_{k_r,k_c}$ is 1.

The soft constraints are then as follows:

$$2 \cdot S1_{k_r,k_c} - x_{i-j,k_r,3,m,n,o} - x_{i-j,k_c,3,m,n,o} \leq 0 \quad (11)$$

$\forall i, \forall j, \forall k_r, \forall k_c, \forall m, \forall n, \forall o,$

$$2 \cdot S1_{k_r,k_c} - x_{i-j,k_r,1,m,n,o} - x_{i-j,k_c,1,m,n,o} \leq 0 \quad (12)$$

$\forall i, \forall j, \forall k_r, \forall k_c, \forall m, \forall n, \forall o,$

$$2 \cdot S1_{k_r,k_c} - x_{i-j,k_r,2,m,n,o} - x_{i-j,k_c,2,m,n,o} \leq 0 \quad (13)$$

$\forall i, \forall j, \forall k_r, \forall k_c, \forall m, \forall n, \forall o.$

Constraint (11) indicates that when DS receiver and joint modules are joined, the objective function is increased by β units. Constraints (12) and (13) indicate that when SS receiver and joint modules (occurring in the same teaching session) are joined, the objective function is increased by β units. In all cases, the objective function is not increased if it is not possible to join the modules.

Tables 3 and 4 illustrate the behaviour of the binary variable $S1$.

In this example, one desires that DS modules C01 and C19 be taught on the same day. In Table 3, course C01 is being allocated to Tuesday, and course C19 is allocated to Thursday. The binary variable $S1$ cannot take value 1 in any case because this would break constraint (11) and therefore not increase the objective function. On the days where there is no allocation, the binary variable's only possible value is 0. A case in which the binary variable $S1$ can take value 1 is illustrated in Table 4.

The binary variable $S1$ may take value 0 or 1 on Tuesday (because courses C01 and C19 are both taught on this day); however, because the objective function is a maximisation, this is the only chance this variable has to take value 1, thus increasing the objective function by β units. The same occurs with SS joint and receiver modules.

4.6. Second Group of Soft Constraints. The second group of soft constraints or the S2 category contains preferences indicating that lecturers who teach more than one SS module may allocate these in the same weekday j . This avoids

TABLE 2: Availabilities.

Lecturer	Monday	Tuesday	Wednesday	Thursday	Friday	Courses
T01	1	1	1	1	1	C01, C02
T02	1	1	1	1		C03, C04
T03	1		1			C01, C03
HNL	1	1	1	1	1	ALL

TABLE 3: S1: variable equals 0.

	S1	Module TI: C01	Module TI: C19	Equation
Monday	0	0	0	$2 \times 0 - 0 - 0 = 0$
Tuesday	0	1	0	$2 \times 0 - 1 - 0 = -1$
Wednesday	0	0	0	$2 \times 0 - 0 - 0 = 0$
Thursday	0	0	1	$2 \times 0 - 0 - 1 = -1$
Friday	0	0	0	$2 \times 0 - 0 - 0 = 0$

TABLE 4: S1: variable equals 1.

	S1	Module: TI-01	Module: TI-19	Equation
Monday	0	0	0	$2 \times 0 - 0 - 0 = 0$
Tuesday	1	1	1	$2 \times 1 - 1 - 1 = 0$
Wednesday	0	0	0	$2 \times 0 - 0 - 0 = 0$
Thursday	0	0	0	$2 \times 0 - 0 - 0 = 0$
Friday	0	0	0	$2 \times 0 - 0 - 0 = 0$

“broken” days with only one teaching session. This preference requires adding the following term to the objective function:

$$\gamma \sum_{i_g} \sum_{i_h} S2_{i_g-i_h}, \quad (14)$$

where i_g represent lecturers who teach SS modules in the first teaching session, thus belonging to I_g , i_h represent lecturers who teach SS modules in the second teaching session, thus belonging to I_h . $S2_{i_g-i_h}$ are binary variables indicating that both a module for the first teaching session and a module for the second teaching session are allocated to the same weekday if its value is 1; otherwise, its value is 0. Each binary variable is connected to a (i_g, i_h) pair. γ is value to increase the objective function should the value of $S2_{i_g-i_h}$ be 1.

The soft constraint is as follows:

$$2 \cdot S2_{i_g-i_h} - x_{i_g-j.k.1.m.n.o} - x_{i_h-j.k.2.m.n.o} \leq 0 \quad (15)$$

$$\forall j, \forall i_g, \forall i_h, \forall m, \forall n, \forall o.$$

Constraint (15) indicates that when two modules taught by a single lecturer are offered in the same day, one in the first and other in the second teaching session, there should be an increase of γ units in the objective function. Should it not be possible to offer the two modules in the same day, there will be no increase in the objective function. Tables 5 and 6 illustrate the behaviour of the binary variable S2.

TABLE 5: S2: variable equals 0.

	S2	1st session	2nd session	Equation
Monday	0	0	0	$2 \times 0 - 0 - 0 = 0$
Tuesday	0	1	0	$2 \times 0 - 1 - 0 = -1$
Wednesday	0	0	0	$2 \times 0 - 0 - 0 = 0$
Thursday	0	0	1	$2 \times 0 - 0 - 1 = -1$
Friday	0	0	0	$2 \times 0 - 0 - 0 = 0$

TABLE 6: S2: variable equals 1.

	S2	1st session	2nd session	Equation
Monday	0	0	0	$2 \times 0 - 0 - 0 = 0$
Tuesday	1	1	1	$2 \times 1 - 1 - 1 = 0$
Wednesday	0	0	0	$2 \times 0 - 0 - 0 = 0$
Thursday	0	0	0	$2 \times 0 - 0 - 0 = 0$
Friday	0	0	0	$2 \times 0 - 0 - 0 = 0$

In the example in Table 5, one wishes that any module of the first session and another module of the second session be taught in the same day. The module in the first teaching session is allocated to Tuesday, and the module in the second teaching session is allocated to Thursday. The binary variable S2 cannot take value 1 in any case as this would violate constraint (15), and thus there will be no increase in the objective function. On days for which there is no allocation, the only possible value for the binary variable is 0. Table 6 illustrates a case in which the binary variable S2 can take value 1.

The binary variable S2 can be either 0 or 1 on Tuesday (because the modules are taught on this same day); however, because the objective function is a maximisation, this is the only chance the variable has to take value 1 and thus increase the objective function by γ units.

4.7. Third Group of Soft Constraints. The third group of soft constraints or the S3 category indicates that if a lecturer teaches only one SS module in a given day, this module should be preferably allocated to the first teaching session. This allows the lecturer to avoid leaving the university at a later time. This preference requires that the following term be added to the objective function:

$$\delta \sum_{i_g} \sum_{i_h} S3_{i_g-i_h}, \quad (16)$$

where $S3_{i_g-i_h}$ is binary variables that, when valued 1, mean that a SS module was allocated in the first teaching session;

TABLE 7: S3: variable equals 0 or 1.

	S3	1st session	2nd session	Equation
Monday	0	0	0	$2 \times 0 - 0 + 0 = 0$
Tuesday	1	1	0	$2 \times 1 - 1 + 0 = 1$
Wednesday	0	0	0	$2 \times 0 - 0 + 0 = 0$
Thursday	0	0	1	$2 \times 0 - 0 + 1 = 1$
Friday	0	0	0	$2 \times 0 - 0 + 0 = 0$

these variables are valued 0 otherwise. Each binary variable is connected to a pair of modules from the first and second teaching sessions. δ is value by which the objective function should be increased if $S3_{i_g i_h}$ is 1.

The soft constraint is as follows:

$$2 \cdot S3_{i_g i_h} - x_{i_g-j.k.1.m.n.o} + x_{i_h-j.k.2.m.n.o} \leq 1 \quad (17)$$

$$\forall j, \forall k, \forall i_g, \forall i_h, \forall m, \forall n, \forall o.$$

Constraint (17) indicates that, when a SS module is allocated to the first teaching session, the objective function is increased by δ units. Should this allocation not be possible, no increment is added to the objective function. Table 7 illustrates the behaviour of the binary variable S3.

In the example given in Table 7, one wishes to have a SS module allocated in the first teaching session. The module allocated to the first teaching session of Tuesday is the only opportunity for the binary variable to take value 1 and thus increase the objective function by δ units. All other allocations violate the soft constraint given by constraint (17) in the case of the binary variable taking a value other than 0.

5. Application of the Single-Criterion Model to Planning the Second Semester of 2011

The criterion adopted for scoring lecturers in modelling the second semester of 2011 was their academic title, that is, the higher the title a lecturer has, the more likely he is to be allocated. A lecturer with only a bachelor's degree or its equivalent receives a score of four, a lecturer with a specialisation receives a score of six, a lecture with a master's degree receives a score of eight, and a lecturer with a doctoral degree receives a score of ten. The higher title determines the score. Penalties for the HNL element and for scheduling unavailability were -1 and -10 , respectively.

The models were solved using *Microsoft Excel* 2010 with the freeware plug-in *OpenSolver* [15] version 1.9; a computer running *Windows 7* was used that had a 64-bit architecture, a Quad Core processor at 2.40 GHz, and 8 GB DDR 2 memory. Table 8 gives the quantitative data detailed below:

- (i) *Number of binary allocation variables* indicates the total number of binary allocation variables.
- (ii) *Number of S1 binary variables* indicates the number of binary variables of the first group of soft constraints.
- (iii) *Number of S2 binary variables* indicates the number of binary variables of the second group of soft constraints.

TABLE 8: Modelling information.

Number of binary allocation variables	3,745
Number of S1, S2, and S3 binary variables	145, 145, and 145
Total number of integer variables	4,180
Number of hard constraints in the first, second, and third groups	302, 205, and 234
Number of soft constraints in the first, second, and third groups	145, 145, and 145
Total number of constraints	1,176
Weights (β, γ, δ)	10; 100; 1
Satisfaction of soft constraints in the first group (from a total of 29 modules)	28 of 29 (96.55%)
Satisfaction of soft constraints in the second group (in pairs of SS modules)	29 of 34 (85.29%)
Satisfaction of soft constraints in the third group (for each nonpaired SS module)	6 of 10 (60.00%)
Lecturers allocated outside of their availability schedule	0 of 77 (0.00%)
Lecturers with no modules allocated	0 of 77 (0.00%)
Computational time	261.41 seconds
Number of unallocated modules (estimated 73 unallocated modules)	80 (8.75% above estimate)

- (iv) *Number of S3 binary variables* indicates the number of binary variables of the third group of soft constraints.
- (v) *Total number of integer variables* indicates the total of integer variables in each given model.
- (vi) *Number of hard constraints in the first group* indicates the number of hard constraints in the first group.
- (vii) *Number of hard constraints in the second group* indicates the number of hard constraints in the second group.
- (viii) *Number of hard constraints in the third group* indicates the number of hard constraints in the third group.
- (ix) *Number of soft constraints in the first group* indicates the number of soft constraints in the first group.
- (x) *Number of soft constraints in the second group* indicates the number of soft constraints in the second group.
- (xi) *Number of soft constraints in the third group* indicates the number of soft constraints in the third group.
- (xii) *Total number of constraints* indicates the total number of constraints of each given model.
- (xiii) *Weights (β, γ, δ)* indicate the weight (β, γ, δ) used for each group of soft constraints. In models two, three, and four, weights were gradually increased to find the maximum number of soft constraints from a given group that could be satisfied. For models two, three, and four, value 100 is a sufficiently large weight to obtain the maximum constraint satisfaction.

- (xiv) *Satisfaction of soft constraints from the first group* indicates how many soft constraints from the first group were satisfied. For the problem presented herein, there are 29 grouping possibilities.
- (xv) *Satisfaction of soft constraints from the second group* indicates how many soft constraints of the second group were satisfied. For every two SS modules (module pairs) allocated, one second-group constraint may be satisfied; therefore, the total number of satisfied constraints varies according to the solution found for the problem. To compare the models, the result will be presented as a percentage.
- (xvi) *Satisfaction of soft constraints from the third group* indicates how many soft constraints from the third group were satisfied. For each second-group constraint satisfied, one third-group constraint was removed. This number of constraints also varies according to the solution found for the problem: to compare the models, the result will be presented as a percentage.
- (xvii) *Lecturers allocated outside of their available schedule* indicate how many lecturers, out of a total of 77, were allocated outside of the available days as informed by them.
- (xviii) *Lecturers with no modules allocated* indicate how many lecturers, out of a total of 77, were not allocated.
- (xix) *Computational time* indicates the total time to find a solution, in seconds.
- (xx) *Number of unallocated modules* indicates how many modules, from an initial estimate of 73, were not allocated.

The model contains 4,180 binary allocation variables and a total of 1,176 constraints. The value of the allocation objective function was 1,318; the model managed to satisfy 96.55% of the soft constraints in the first group, 85.29% of the soft constraints in the second group, and 60.00% of the soft constraints in the third group. No lecturer remained with no module allocated to him/her or otherwise allocated outside of the provided availability schedule. The solution was reached in 261.41 seconds, and a total of 80 modules was left unallocated. The complete model used the following weights: ten for the first group of soft constraints, 100 for the second group of soft constraints, and one for the third group of soft constraints. These weights were chosen after several tests with different values, which revealed that the soft constraints of the second and third groups compete with each other. While the constraints of the second group attempt to allocate SS modules to the same day, the constraints of the third group attempt to allocate all SS modules in the first teaching session. To ensure that timetables are as coherent as possible, the second-group constraints should have larger weights than the constraints of the third group. The number of unallocated modules remained quite close to the original estimate; therefore, the number of new lecturers to be hired remained practically unchanged. The institution's managers and course coordinators approved the timetable generated by

the complete model, which was implemented in the second semester of 2011.

6. Final Remarks

The general goal of this study was reached: to build a mathematical model that enables the identification of timetables that best meet the specific needs of lecturer allocation in the institution being studied. Data gathering on the institution was all-encompassing; the main sources were institution page on Internet, institutional Intranet, documental research, and key staff interviews. From the information obtained from the gathered data, it was possible to clearly identify and classify the main problems involved in this timetabling, namely, employing *piggybacking* to minimise costs, scheduling SS modules on the same day, and, in the case of broken schedules, filling the first teaching session rather than the second. The literature labels these problems or challenges as soft constraints. The literature further labels another existing category of constraints as hard; and these constraints should be satisfied in order to generate feasible solutions. For the case study approached in this work, the hard constraints completely avoided problems such as collisions in lecturers' schedules, colliding modules, and the inability to offer modules planned for a given period. Soft constraints were separated into groups and solved by introducing binary variables. When each binary variable took the value 1, a soft constraint of some given group was satisfied. The model's solution improved as soft constraints were satisfied. It was always possible to satisfy hard constraints because an artefact was introduced to allow feasible timetables to be created even if the lecturer staff was not complete; when a module was allocated to this artefact, this indicated that a new lecturer must be hired. Once the hard and soft constraints were clearly defined, a model that allocates lecturers according to their academic title (the higher the title, the greater the possibility of allocation), minimises costs by joining classes whose modules' syllabuses are equivalent, and meets lecturers' scheduling preferences (generating timetables that are more practical and comfortable for these professionals) was developed. The model was accepted by the institution's managers and by the courses' coordinators and has been in use at the institution since the second semester of 2011. However, one can note that our approach used a case study as a basis, and because of the nature of the problem, the model cannot be fully generalised, to solve all types of problems.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

- [1] A. Tripathy, "Computerised decision aid for timetabling—a case analysis," *Discrete Applied Mathematics*, vol. 35, no. 3, pp. 313–323, 1992.
- [2] A. Schaerf, "A survey of automated timetabling," *Artificial Intelligence Review*, vol. 13, no. 2, pp. 87–127, 1999.

- [3] D. de Werra, "An introduction to timetabling," *European Journal of Operational Research*, vol. 19, no. 2, pp. 151–162, 1985.
- [4] M. J. F. Souza, N. Maculan, and L. S. Ochi, "Melhorando quadro de horários de escolas através de caminhos mínimos," *Tendências em Matemática Aplicada e Computacional*, vol. 1, no. 2, pp. 515–524, 2000.
- [5] E. K. Burke, S. Petrovic, and R. Qu, "Case-based heuristic selection for timetabling problems," *Journal of Scheduling*, vol. 9, no. 2, pp. 115–132, 2006.
- [6] E. Burke, K. Jackson, J. H. Kingston, and R. Weare, "Automated university timetabling: the state of the art," *The Computer Journal*, vol. 40, no. 9, pp. 565–571, 1997.
- [7] S. Daskalaki, T. Birbas, and E. Housos, "An integer programming formulation for a case study in university timetabling," *European Journal of Operational Research*, vol. 153, no. 1, pp. 117–135, 2004.
- [8] G. M. Thompson, "Using information on unconstrained student demand to improve university course schedules," *Journal of Operations Management*, vol. 23, no. 2, pp. 197–208, 2005.
- [9] K. Schimmelpfeng and S. Helber, "Application of a real-world university-course timetabling model solved by integer programming," *OR Spectrum*, vol. 29, no. 4, pp. 783–803, 2007.
- [10] M. A. Bakir and C. Aksop, "A 0-1 integer programming approach to a university timetabling problem," *Hacettepe Journal of Mathematics and Statistics*, vol. 37, no. 1, pp. 41–55, 2008.
- [11] A. O. Adewumi, B. A. Sawyerr, and M. M. Ali, "A heuristic solution to the university timetabling problem," *Engineering Computations*, vol. 26, no. 8, pp. 972–984, 2009.
- [12] H. A. Razak, Z. Ibrahim, and N. M. Hussin, "Bipartite graph edge coloring approach to course timetabling," in *Proceedings of the International Conference on Information Retrieval & Knowledge Management (CAMP '10)*, pp. 229–234, IEEE, March 2010.
- [13] H. Y. Al-Tarawneh and M. Ayob, "Using Tabu search with multi-neighborhood structures to solve University Course Timetable UKM case study (faculty of engineering)," in *Proceedings of the 3rd Conference on Data Mining and Optimization (DMO '11)*, pp. 208–212, Putrajaya, Malaysia, June 2011.
- [14] L. R. Foulds and D. G. Johnson, "SlotManager: a micro-computer-based decision support system for university timetabling," *Decision Support Systems*, vol. 27, no. 4, pp. 367–381, 2000.
- [15] A. J. Mason and I. Dunning, "OpenSolver: open source optimization for excel," in *Proceedings of the 45th Annual Conference of the ORSNZ*, pp. 181–190, Operations Research Society of New Zealand, Auckland, New Zealand, November 2010.