mathematics

MDPI

Article

Considering Section Balance in an Integer Optimization Model for the Curriculum-Based Course Timetabling Problem

Cristian D. Palma * and **Patrick Bornhardt**

Facultad de Ingeniería, Universidad del Desarrollo, Concepción 4040418, Chile; pbornhardtd@ingenieros.udd.cl
* Correspondence: cristianpalma@ingenieros.udd.cl; Tel.: +56-41-268-6423

Received: 19 August 2020; Accepted: 6 October 2020; Published: 13 October 2020

Abstract: University course timetabling is a complex and time-consuming duty that every educational institution faces regularly. It consists of scheduling a set of lectures in predefined time slots so as to avoid student conflicts, meet teacher and room availability, and manage several institution-specific operational rules. In this paper, we schedule courses based on a curriculum, that is, before the students' registration. Unlike other curriculum-based models, the proposed model considers two practical aspects when managing the conflicts between lectures: (i) it schedules sections of subjects so that each section is evenly likely to be registered by the students, and (ii) it considers the failure rates and periodicity a subject is taught. We present a multi-objective integer programming model that maximizes the use of specific time slots, the symmetry in which the lectures of a course are scheduled during a week, and the flexibility for straggler students to take courses. The model is solved using commercial software, and it is applied to a real course-timetabling problem. We show the advantages of its use by comparing the model's solution with the actual solution obtained by the manual scheduling.

Keywords: course timetabling; integer programing; balanced scheduling; curriculum-based timetabling

1. Introduction

Educational institutions constantly face the problem of scheduling activities where students, professors, classrooms and other resources are involved. Examination timetabling [1], curriculum design [2] and course timetabling [3] are some of these problems. In the later, a number of courses need to be scheduled over a set of time slots, subject to both capacity and operational constraints. While the different considerations to take into account when dealing with these problems makes them very difficult to solve, many institutions still do it manually, usually consuming the working hours of many people [3]. The use of mathematical tools to assist in this type of problems has translated into a more efficient use of resources, a significant reduction in the time required to generate the timetable, and the possibility of quickly evaluating alternative scenarios [4].

The timetabling problem has been thoroughly addressed using operations research. While the different models available in the literature are tailored to the needs of specific institutions, they all divide, based on their own reality, the problem components into hard and soft constraints [3]. The former are requirements that need to be met, e.g., availability of rooms and professors, and specific operational/administrative rules; the later are wishes that should be met as well as possible, e.g., time slots preferences and the avoidance of spare periods. Just as the conditions of the problem greatly differ from one institution to another, so does the objective of the timetabling decision problem. Some of the most common indicators of good timetabling are the maximization of the students' and/or professors' preferences, the maximization of the compactness of the timetable (from the students' or from the professors' point of view), and the minimization of conflicts among the courses taken together. The variety of characteristics that a good timetable should possess has motivated the use of multi-objective approaches, and the NP-hard nature of the problem [5,6] has led to the search for different solution strategies. According to Sørensen and Dahms [6], heuristics are still the most popular techniques to solve timetabling problems, although the direct solution of integer programming formulations has also proved to be a good approach [7,8].

In this study, a multi-objective integer-programming model that addresses some new aspects of the university curriculum-based course-timetabling problem is formulated. Along with the basic requirements (i.e., room capacity, professors' availability, management of same-semester class conflicts, etc.), we considered (i) the failure rates and periodicity of delivery of subjects when avoiding conflicts between courses taken in different semesters within a program, and (ii) the possibility of handling conflicts between the sections of subjects that can be taken simultaneously. The later point avoids situations in which, for example, two sections of a subject (classes in which a subject is divided due to a large number of students) are scheduled to meet all the requirements, but only few students are allowed to register in one of the sections due to conflicts with classes of other subjects. In this case, an unbalanced number of students are allowed to take each section. To our knowledge, no model has addressed this important practical issue in a curriculum-based course timetabling approach. We formulate an integer programming model, solve the model using a commercial solver, and apply it to the course timetabling problem faces by the Faculty of Engineering (FE) at Universidad del Desarrollo in Chile. The benefits of using the model clearly demonstrate superiority when compared to the course timetable obtained manually.

2. Literature Review

A timetabling problem consists of coordinating different elements, e.g., workers, events, sport teams, referees, students, teachers, subjects, etc., within a period divided into time slots. In an educational frame, this type of problems recurrently arises when the lectures or the examinations of different subjects need to be scheduled to meet institutional rules, and students' and teachers' requirements, both in schools and universities. The literature in educational timetabling problems is vast, and includes model-based methods, (meta)heuristic approaches and the use of constraint logic programming, among other solution approaches. This review is not intended to be exhaustive, but to discuss the most relevant situations considered in the existing literature. For detailed surveys, the reader is referred to the works of Schaerf [9] and Pillay [10].

School and university timetabling problems mainly differ in that the former has disjoint sets of students, while the latter has common students. When courses have common students, managing the course conflicts becomes more complicated. Additionally, room availability seems to play a major role in the university problem since, at schools, each class has its own room [9]. Thus, solving university timetabling is considered to be more complicated to solve than school timetabling [11].

While models to solve this university course timetabling are specific to the institution for which they were designed, some standard constraints are common to all university timetabling problems [11]. For instance, professors can teach only one course at a time, room availability is limited, courses require a specific number of time slots during a week, and groups of courses scheduled in the same semester of a study program cannot conflict with each other in order to allow students to take them simultaneously. Along with these requirements, other non-standard conditions are sometimes necessary. In several cases, some lessons of certain courses are to be scheduled in consecutive periods [5,11,12], while for other courses lessons are required in different days [11,13], or even strictly every other day [7]. The latter condition is essentially imposed to spread lessons out evenly throughout the week, a desirable attribute of a timetable that has been mostly considered a soft constraint and therefore attainable only when possible, as in our case. Symmetry in the course timetable, in which the lectures of a course are scheduled in the same time slot every other day, are usually implicitly considered when courses are assigned to patterns of time slots rather than to time slots [4,14]. The use of patterns to group decisions makes models simpler, at the cost of a reduction in the feasible space. In our work, we explicitly model the symmetry in the course timetables, and consider it to be a soft constraint. Another feature commonly considered is the compactness of the timetables, which can be explicitly modeled in integer programming formulations [13,15,16] or can be easily incorporated when patterns of courses are used [17].

Less common requirements have also been considered, mostly through soft constraints. Some authors prefer all sessions of a course to be scheduled in the same room [5], or a group of courses to be scheduled in the same building to reduce students moving between lectures [13]. In some cases, the registration of the students in a set of courses is used to assign the students to specific sections of the courses, and then to produce a timetable that minimizes the student time conflicts while keeping the section size limited [13,18,19]. In this type of approach, the courses are scheduled based on a pre-enrolment or after the students have enrolled, in a sort of demand driven system. Either in a pre-enrolment approach or in a curriculum-based approach, the proper manage of sections is an important step in real-life applications of course timetabling [20]. In this study, we consider the balance of the potential section sizes in a curriculum-based course-timetabling problem, that is, in which the courses are scheduled before enrolment. Given that section issues have been mainly ignored in curriculum-based approaches [20], this paper contributes a modeling approach to reduce this gap.

3. Materials and Methods

3.1. Problem Description

The academic year at FE is divided into two semesters, each containing compulsory and optional subjects that are connected through prerequisite relationships. If many students take a given subject $s \in S$, where S is the set of subjects, then s needs to be taught in more than one class, known as 'sections' of a subject.

Let's denote C as the set of all the classes that need to be scheduled, some of them being sections of the same subject. C is assumed to be known, as the number of expected students taking a subject can be well estimated based on historical data and failure rates. In some cases, a pre-registration survey can also be conducted before the previous term ends. Once the set C is known, professors are assigned to each class depending on their workload and expertise, therefore the professor-class assignment is not part of the timetabling problem. For each class $c \in C$, a given number of lectures, n_c, has to be scheduled during a week, in some of the available time slots $t \in T = \{1, \dots, T\}$. Each semester, about 70 classes have to be scheduled in 45 time slots (5 days × 9 time slots a day).

Each class needs a classroom of a certain type (e.g., regular classroom or laboratory) and size based on their teaching requirements and estimated number of students. Although most of the models consider the classroom allocation as a decision variable, we only considered the maximum availability of rooms, as in the works by Dimopoulou and Miliotis [17], since the subsequent class–classroom allocation is easy to solve manually.

The FE has two campuses, located in two different cities, for which the course timetable is independently scheduled. The only consideration for scheduling each of the two campuses timetable is that some professors teach in both campuses, so when visiting a campus their weekly sessions (usually two) needs to be scheduled during the same day, in contiguous time slots. Of course, during these days the professors are not available in their home campuses. Industry leading professionals also teach as part-time professors, in which case sessions are also scheduled in contiguous time slots during a day.

In this case, a good course timetable would be one that (i) schedules the lectures as early as possible during the day, hopefully reducing spare periods, (ii) enables straggler students some flexibility to take subjects and move forward in their study programs, and (iii) spreads lectures of a subject symmetrically within a week. That is, lectures should be scheduled in the same time slot every other day, e.g., in the same time slot Mon-Wed, Tue-Thu, Wed-Fri in case that $n_c = 2$, or Mon-Wed-Fri in case that $n_c = 3$.

Some practical considerations that can be observed in university timetabling distinguish this model from others available in the literature. In addition to the infrastructure capacity, professors' availability, and subjects conflict avoidance constraints, we considered issues that, to our knowledge, have not been previously considered. First, we considered the failure rates and the periodicity a subject is taught when managing the conflict of subjects programmed to be taken in different semesters within a program. Secondly, we balanced the conflicts between sections of subjects that can be taken simultaneously. This point is quite a significant practical issue. Sometimes two sections of a subject are perfectly scheduled to meet all the requirements, but only few students are actually able to take one of these sections due to conflicts with other classes. This unbalanced situation should be avoided in a good course timetable.

3.2. Model Formulation

Many models in the literature use some sort of clustering methods to group either lectures or time slots into predefined patterns [4,8,21,22]. While these techniques reduce the problem size, they also reduce the feasible space. Since our timetabling problem is solved in a very limiting setting, we opted not to group decisions into patterns. The following variables and parameters are used in our formulation.

3.2.1. Decision Variables

x_{ct} = 1 if a lecture of class c is scheduled in time slot t, 0 otherwise.

v_{ct} = 1 if a lecture of class c scheduled in time slot t is symmetric with another lecture, 0 otherwise.

z_{ct} = 1 if a lecture of class c is scheduled in time slot t and $t + 1$, 0 otherwise.

$y_{cc'}$ = 1 if a lecture of class c and a lecture of class c', charted in different semesters in the academic program, are scheduled in different time slots to allow straggle students to take them simultaneously, 0 otherwise.

$u_{cc'}$ = 0 if a lecture of class c and a lecture of class c', that can be taken simultaneously regardless of being charted in the same or in different semesters, are scheduled at the same time-slots. This variable can be 1 if a lecture of class c and a lecture of class c' are scheduled at different time-slots.

3.2.2. Parameters

m_t = desirability of scheduling a lecture in time slot t (priority of use).

n_c = number of weekly lectures of class c (usually $1 \leq n_c \leq 3$).

$sem(c)$ = semester within the academic program in which class c is charted.

$year(c)$ = year within the academic program in which class c is charted.

$p(c)$ = professor that teaches class c.

$subj(c)$ = subject $s \in S$ corresponding to class c.

f_c = historical failure rate of class c.

k_{ert} = number of rooms of type e and size r available in time slot t.

d_{ct} = 1 if the professor of class c is available in time slot t, 0 otherwise.

h_c = 1 if a class c is taught once a year (every other semester), 0 otherwise.

q_c = 1 if a class c is not a prerequisite for other classes, 0 otherwise.

l_{cs} = 1 if class c can be taken simultaneously with a class of subject s, 0 otherwise.

$g_{cc'}$ = 1 if classes c and c' belong to subjects that can be taken simultaneously, 0 otherwise.

b_{cs} = number of sections of a subject s that cannot conflict with class c.

α_i = weight associated with objective i.

3.2.3. Additional Sets

$\mathbf{A}(e, r)$ = set of classes that can be scheduled in a room type $e \in \mathbf{E}$ and size $r \in \mathbf{R}$ (includes classes smaller than the room capacity, that is, a class will be part of all sets A where r is larger than the minimum size required).

$\mathbf{C}^{'}$ = set of classes that need to be scheduled in consecutive time slots, $\mathbf{C}^{'} \subseteq \mathbf{C}$.
$\mathbf{T}^{'}$ = set of time slots where 1st, 2nd and 3rd year classes cannot be scheduled, $\mathbf{T}^{'} \subseteq \mathbf{T}$.

3.2.4. Constraints

1. Number of weekly lectures: we need to schedule the lectures of each class a fixed number of times a week.

$$\sum_{t \in \mathbf{T}} x_{ct} = n_c \quad \forall c \in \mathbf{C} \tag{1}$$

2. Professors' availability: classes with the same professor should not be scheduled at the same time. While defining a constraint for each professor and time slot would produce fewer constraints, doing it for each class resulted in shorter solution times.

$$x_{ct} + \sum_{c' \neq c | p(c) = p(c')} x_{c't} \leq d_{ct} \quad \forall c \in \mathbf{C},\ t \in \mathbf{T} \tag{2}$$

3. Daily scheduling: most lectures of a class should be scheduled in different days (3a), but some of them ($c \in \mathbf{C}^{'}$) should be scheduled in consecutive time slots (3b) and (3c)

$$\sum_{t'=t}^{t+8} x_{ct'} \leq 1 \quad \forall c \in \mathbf{C} \backslash \mathbf{C}^{'},\ t \in \{1,\ 10,\ 19,\ 28,\ 37\} \tag{3a}$$

$$x_{ct} + x_{c(t+1)} \geq 2 \cdot z_{ct} \quad \forall c \in \mathbf{C}^{'}, t \in \mathbf{T} \backslash \{9, 18, 27, 36, 45\} \tag{3b}$$

$$\sum_{t \in T \backslash \{9,\ 18, 27, 36, 45\}} z_{ct} = n_c - 1 \quad \forall c \in \mathbf{C}^{'} \tag{3c}$$

4. Room availability: the number of lectures scheduled at the same time slot must not exceed the room availability. Note that we assume that lectures can be scheduled in a room larger than needed, and that rooms are ordered from smallest to largest.

$$\sum_{c \in A(e,r)} x_{ct} \leq \sum_{r' \leq r} k_{er't} \quad \forall\ e \in \mathbf{E}, r \in \mathbf{R},\ t \in \mathbf{T} \tag{4}$$

5. Blocked time slots: In our case, some time slots are not available for first, second and third-year students since other interfaculty-activities at a university level are scheduled.

$$\sum_{c | year(c) \in \{1,2,3\}} x_{ct} = 0 \quad \forall\ t \in \mathbf{T}' \tag{5}$$

6. Variables of symmetry: the model should promote the symmetry of the different lectures of a class (i.e., same time slot every other day). This symmetry is captured through the decision variable v_{ct}, and then used in the objective function.

$$x_{ct} + x_{c(t+18)} \geq 2{\cdot}v_{ct} \quad \forall\ c \in \mathbf{C} \backslash C',\ t \in \mathbf{T},\ t \leq 27 \tag{6}$$

7. Strict avoidance of conflicts: Subjects charted in the same semester within the study program cannot be scheduled in the same time slot. However, when subjects are taught in more than one section, there is no problem if some sections of one subject conflict with some sections of the other subject. We therefore allow the model to schedule some of these classes in the same time slot ($u_{cc'} = 0$, Equation (7a)), but ensuring that at least a minimum number of sections (b_{cs}) of a subject s cannot conflict with a class c (Equation (7b)).

$$x_{ct} + x_{c't} \leq 2 - u_{cc'} \quad \forall\ c, c' \in \mathbf{C}, c \neq c', Subj(c) \neq Subj(c'), sem_c = sem_{c'},\ t \in \mathbf{T} \tag{7a}$$

$$\sum_{c'|Subj(c')=s} u_{cc'} \geq b_{cs} \ \ \forall\, c \in \mathbf{C},\ s \in \mathbf{S},\ l_{cs} = 1 \tag{7b}$$

As already mentioned, the b_{cs} parameter is used to balance the number of students able to register in a section of a subject, and it is obtained as follows:

$$b_{cs} = \left\lceil \frac{N^{\circ}\ sections\ of\ subject\ s}{N^{\circ}\ of\ sections\ of\ a\ subject\ s'such\ Subj(c) = s'} \right\rceil$$

For example, suppose that a subject $s1$ is taught in three different sections ($s1$-1, $s1$-2 and $s1$-3), and a subject $s2$ is taught in two sections ($s2$-1 and $s2$-2). It would be great if none of the lectures of sections $s1$ and $s2$ conflict with each other, but this situation may over-constrain the solution, and in some cases, make it infeasible. Instead, for each section associated with s1, let's say $s1$-1, we force that at least 1 ($\lceil 2/3 \rceil$) section of $s2$ has no conflict. That is, $u_{s1-1,s2-1} + u_{s1-1,s2-2} \geq 1$. For each section associated with $s2$, let's say $s2$-2, at least 2 ($\lceil 3/2 \rceil$) sections of s1 need to have no conflict, that is, $u_{s2-2,s1-1} + u_{s2-2,s1-2} + u_{s2-2,s1-3} \geq 2$. These constraints work together with constraints (7a), that actually determines the values of x and u.

Suppose now that a subject $s3$ has two sections. Each of them cannot conflict with two sections of $s1$ and one section of $s2$, as we already discussed. However, it can happen that $s2$ and $s3$ are scheduled in the same pair of time slots as two sections of $s1$ (Figure 1a), forcing most of the students to register in the third section of $s1$.

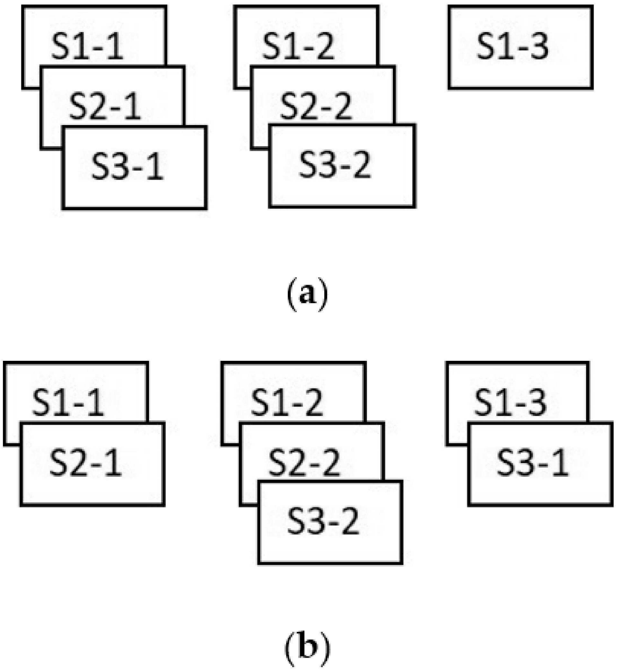

Figure 1. (**a**) A class scheduling that meets constraints (7a) and (7b) but produces an unbalanced number of students in subject $s1$. Only few students are allowed to register in $s1$-1 and $s1$-2. (**b**) A more balanced scheduling is obtained when constraints (7c) are considered.

To avoid this situation and 'balance' the number of students across the sections of $s1$, we impose the following constraint that acts as a transitivity relationship among sections.

$$u_{cc'} + u_{c'c''} \leq 1 + u_{cc''} \ \ \forall\, c, c', c'' \in \mathbf{C},\ g_{cc'} = g_{c'c''} = g_{cc''} = 1 \tag{7c}$$

Given that for some c and c' with no conflict $u_{cc'}$ can still be zero (7a), only some of the $u_{cc'}$ will be required to be one to meet constraint (7b). This condition will produce that some transitivity relationships hold and balance the sections. For example, if $u_{s1-1,\, s2-2} = u_{s2-2,\, s3-1} = 1$, then $u_{s1-1,\, s3-1}$ will be forced to be one (7c), and a more balanced scheduling can be obtained (Figure 1b),

8. Flexible avoidance of conflicts (classes of different semesters): For classes charted in different semesters, the previous consideration (balance of sections) also holds, but the need to avoid conflicts is less important as mainly straggler students fall in this situation. We allow the model to decide if classes of different semesters are scheduled without conflict ($y_{cc'} = 1$) or not, based on their failure rates, the frequency a subject is taught and whether a subject is a requisite for other courses (see objective function).

$$x_{ct} + x_{c't} \leq 3 - u_{cc'} - y_{cc'} \;\; \forall\, c, c' \in \mathbf{C} \big| g_{cc'} = 1,\; sem_c \neq sem_{c'}, f_c > 0 \tag{8a}$$

$$y_{cc'} \leq u_{cc'} \;\; \forall\, c, c' \in \mathbf{C} \tag{8b}$$

If historically nobody fails a subject s ($f_c = 0$), then lectures of a class $c \in \mathbf{C}$ can be allowed to conflict with lectures of another class c' charted in a different semester, and the set of constraints (8) do not apply.

9. Fixed variables: To avoid some variables to take on inappropriate values that artificially affect the objective function, we included the following constraints:

$$v_{ct} = 0 \;\; \forall\, c \in \mathbf{C},\; t > 27 \tag{9a}$$

$$y_{cc'} = 0 \;\; \forall\, c, c' \in \mathbf{C}, sem_c = sem_{c'} \tag{9b}$$

$$y_{cc'} = 0 \;\; \forall\, c, c' \in \mathbf{C}, f_c = 0 \tag{9c}$$

$$u_{cc'} \leq g_{cc'} \;\; \forall\, c, c' \in \mathbf{C} \tag{9d}$$

10. Objective function: a broad range of different objectives have been considered in course timetabling problems [3], and most of the models in the literature consider a weighted combination of different objectives. In the same direction, we maximized the following three objectives:

Objective 1: the schedule of the lectures in preferred time slots. m_t represents a measure of the preference of using time slot t. In our case, earlier time slots were preferred so m_t was defined as a decreasing number throughout the day, but another type of utility function can be used. We want this objective to be maximized.

$$O_1 = \sum_{c \in \mathbf{C}} \sum_{t \in \mathbf{T}} m_t x_{ct} \tag{10a}$$

Objective 2: the flexibility for straggler students to taking subjects and progress in their programs. The timetable should allow students that fail in a subject (class c) to take subjects charted in another semester (class c'), by making them not to conflict ($y_{cc'} = 1$). The relevance to avoid a conflict comes from: (i) the failure rates (f_c) of both subjects, i.e., the higher the failure rates, the more the need to avoid the conflict; (ii) the frequency that a subject is taught, i.e., if a subject is taught only once a year ($h_{c'} = 1$), then it is more relevant to avoid the conflict; and (iii) whether a subject is requisite to take another subject. i.e., if a subject is not requisite for other subjects ($q_{c'} = 1$), then they could conflict since the subjects can be taken in the future without producing delays to the students.

$$O_2 = \sum_{c \in \mathbf{C}} \sum_{c' \in \mathbf{C} | c' \neq c} y_{cc'} (1 + f_c + f_{c'})(1 + h_{c'} - q_{c'}) \tag{10b}$$

Objective 3: the time slot symmetry of the lectures of a class.

$$O_3 = \sum_{c \in \mathbf{C}} \sum_{t \in \mathbf{T}} v_{ct} \tag{10c}$$

These objectives were combined in a weighted sum approach as previously done by others [8,23,24]. Unlike other models, we maximized the relative improvement over the worst result for each objective as a way to overcome the problem of the magnitudes of the different objectives. Let O_i^+ and O_i^- be the best and the worst possible value of objective i (i = 1, 2, 3), respectively. These values, also known as ideal and non-ideal solutions [25], were determined by solving each objective separately. For each objective, O_i^+ is directly obtained when maximizing objective i. O_i^- is obtained from picking the worst level of objective i observed after running the model for all the objectives.

We therefore maximized the following objective function, where α_i is the weight of objective i:

$$Max \sum_{i=1}^{3} \alpha_i \left(\frac{O_i - O_i^-}{O_i^+ - O_i^-} \right) \tag{11}$$

4. Results

The model was solved using the commercial solver Lingo in connection with MS Excel, for which a user-friendly VBA application was designed to input the model's parameters and sets described in Section 3.2. To evaluate the performance of the model, we scheduled the classes of the first semester of 2017 in the Faculty of Engineering at Universidad del Desarrollo, Chile. We compared the timetable obtained using our model with the actual timetable, which was manually designed by two persons during a period of about two weeks. The instance we run had almost 22,000 decisions variables (all binaries) and 214,000 constraints, and was solved in less than an hour in a 3.4 GHz i5 processor with 8 GB RAM. For all runs, a 5% gap was defined.

Initial runs. To implement the objective function, each objective was first independently optimized, as described above. From the results of these runs, the best and worst possible values of each objective were obtained (Table 1).

Table 1. Results of the initial runs.

Objective Function	Levels of Different Objectives		
	Time-Slot Preference	**Flexibility**	**Symmetry**
Time-slot preference	806	136.090	12
Flexibility	609	151.917	3
Symmetry	642	144.929	54

Final run. Based on the results of the initial runs, the multi-objective function (Equation (11)) was formulated and the model solved, without loss of generality using the same preferences for each objective, i.e., $\alpha_i = 1/3\ \forall i$. The relative improvements over the worst possible values were 54%, 81% and 76% for the three objectives, respectively.

The use of the model improved the three objectives in relation with the manual timetabling (Table 2). It is worth noting that the manual timetabling failed to meet two constraints of the model. It scheduled one of the classes in a room smaller than needed and two lectures of a subject in the same day. The most significant improvements over the manual timetabling was observed in the symmetry of the resulting timetable, where the indicator improved by about 280%. This improvement translated into 31 more lectures that were now scheduled in the same time-slot every other day.

Table 2. Improvements in the three objectives were observed with the model timetabling.

Objective	Solution Approach	
	Model	**Manual**
Time-slot preference	716	633
Flexibility	149.079	144.776
Symmetry	42	11

The indicator of time slot preferences rose 13%. In our case, the earlier a lecture was scheduled better, which led to an increase of lectures scheduled in the morning time slots (Figure 2). Finally, the flexibility indicator increases by about 3% (Table 2).

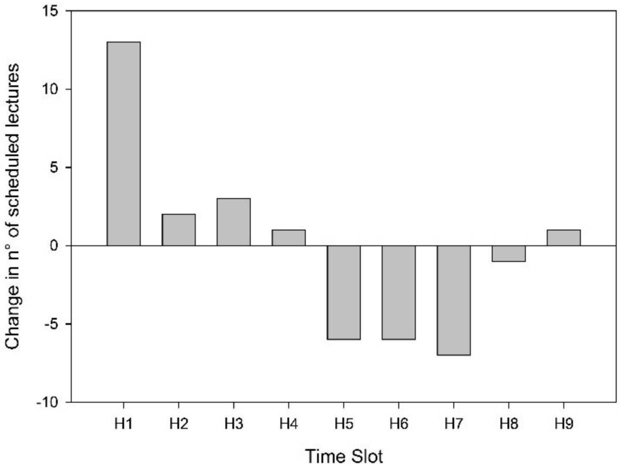

Figure 2. An increase in the lectures scheduled in the morning time slots was observed using the model.

To observe the effect of the set of constraints (7), which balance the availability of sections of a course, look at the solution for the seventh semester of our case study, shown in Figure 3. Five courses must be registered during this semester, and all courses except Finance (Fin) are taught in two sections. Dotted lines represent the conflicts between two course-sections obtained running the model without the set of constraints (7). While this solution allows students to take all courses simultaneously, some unbalances are observed. The most evident unbalance occurs with Optim II-S1, which has a conflict with both sections of MechFluid. That is, any student taking MechFluid will not be able to take Optim II-S1, which will produce an over registration of students in Optim II-S2. A similar situation is observed with MicroEcon-S1, which will receive many students' registrations because most of them will probably be taking Finance. The conflicts obtained when including the set of constraints (7) in our model are shown with solid lines. In this case, conflicts between the two sections of Optimization II and Mechanic of Fluids were obtained. These conflicts mean, for example, that students taking Optimization II-S1 cannot take Optimization II-S2 because at least one lecture of the courses is scheduled at the same time. These conflicts are not a problem, given that no student will register in both sections of the same course.

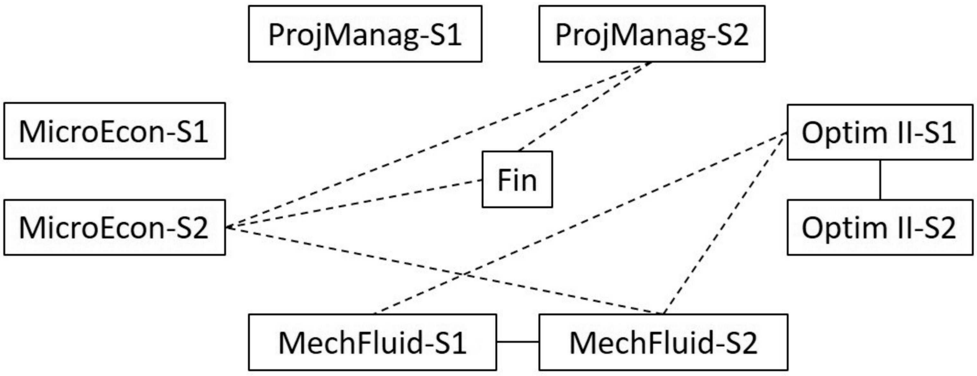

Figure 3. Conflicts between course-sections when set of constraints (7) is not considered in our model (dotted lines) produced an unbalanced schedule. This situation leads to sections with only few students while other sections concentrate most of the students. The conflicts when set of constraints (7) is considered (solid lines) produced a more balanced schedule.

5. Discussion

Our results show that the solution obtained by the proposed formulation of the course-timetabling problem outperformed, in all aspects, the current solution of a real instance of the problem, and successfully managed some practical issues not previously considered in the literature.

We considered the three criteria used to manually solve the problem as our objectives in a multi-objective maximization model, namely the use of preferred time slots, the flexibility for straggler students to take courses, and the symmetry of the lectures schedules within a week. Decisions are associated with a reward for each of these objectives, on the same sense as a penalty function in a minimization model [11]. While a multi-objective approach is common in course timetabling, it is less common to translate the objectives measure to a relative increase as we did. This approach helped us to avoid the difficulties associated with the different magnitudes of the objectives and their unit measures. The only disadvantage of doing so is that multiple runs, one for each objective, are required before the final run to generate the best and worst possible results.

Regarding our specific results, the significant increase in the symmetry of the lecture schedule is explained because, when the timetable is manually designed, the symmetry is certainly the less important criteria. Moreover, the difficulty of meeting the non-conflict requirements and the professors and room availability makes that when solving the problem manually, the symmetry is, although kept in mind, rarely considered. This leads to a very significant increase in this attribute when using the model. On the other hand, the flexibility of the timetable to enable struggling students to take courses is probably the most important criteria used in the manual solution approach of the problem. Due to the many interactions among passed and failed courses, and the different requisite relationships, finding a conflict-free and flexible timetable is a highly time-consuming duty. In our case, the manual process of course scheduling takes a couple of weeks, and two people to check and review the different possibilities that students may have to register in the different courses based on their requisites. A conservative estimation of the time required for a manual generation of a timetable is at least a hundred working hours. Despite all the effort in this regard, the model solution improved the flexibility indicator. Including the time required to gather and enter de input data, and the individual and the final runs of model, the time required to produce the timetable using the model did not exceed ten hours. Finally, the model outperformed the greedy allocation of lectures to preferred time slots that predominate in the manual approach.

Commonly, the subjects with too many students are taught in two or more sections, which are usually scheduled at different time slots to give more flexibility to students and because sometimes the same professor teaches several sections. As a result of the interaction with other subjects taken by the same students, it happens that some sections are potentially only available to a few students. This unbalance in the potential number of students registered in a section produces a practical issue that has to be avoided, but that to our knowledge has not been considered in previous models. Solving this issue with the manual solution approach takes hours of work, and we suspect that with current models implies that optimal solutions have to be revised. Our model successfully represents this situation and balances the potential number of students by ensuring that a minimum number of non-conflicts schedules has to occur between each section of a subject and the rest of the subjects. The minimum number of non-conflicts schedules required is explicitly estimated based on the number of sections of the subjects.

Students that are up to date in their study program must have no conflict among the subjects they need to take, and this is guaranteed in a good timetable. Straggler students face more difficulties when taking courses, as the courses they are able to register may have conflicts. Non-conflict constraints used in the former case can also be considered in the later [15], but doing so may produce an over-constrained model. Other authors have minimized the number of conflicts [7,26], which turns out to be similar to what it is done in this study. We favor flexible timetables for straggler students by maximizing an indicator related with the number of non-conflicts among courses that, although charted in different semesters, can be registered simultaneously. In our case, this indicator is heavier than those subjects with higher failure rates since they have to be taken by more students and are therefore more demanded, and subjects that are taught only once a year. This makes sense as failing a subject taught every other semester may limit the chances to register in another subject in the next semester. Those subjects that are not requisite for other subjects are considered to be lighter as failing them does not limit the possibilities to take another subject.

While the research effort to tackle the timetabling problem is vast, there is room for future work as every institution may have particular operational rules. Research efforts should address the combination of different curricula or study programs into a unique timetabling problem since only a few papers have tackled this problem (e.g., Reference [15]). In this case, subjects of different curricula need to be scheduled simultaneously as some subjects are common to more than one curriculum and have different requisites; therefore, more intricate conflict relationships need to be considered. Additionally, we expect to include, in our model, the possibility to split one of the weekly sessions into smaller groups to schedule special sessions of some courses. These smaller groups require a different type of room, and need to be scheduled in different time slots in a way that professors are available and students in each group have no conflict with other subjects. Finally, since models are expected to become more complex as more practical issues are considered, the study of new solution approaches for the timetabling problem is very fruitful for further research.

Author Contributions: Conceptualization, C.D.P. and P.B.; methodology, C.D.P. and P.B.; software, P.B.; validation, P.B.; formal analysis, C.D.P. and P.B.; investigation, C.D.P.; resources, C.D.P.; data curation, P.B.; writing—original draft preparation, C.D.P.; writing—review and editing, C.D.P.; visualization, C.D.P.; supervision, C.D.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Al-Yakoob, S.M.; Sherali, H.D.; Al-Jazzaf, M. A Mixed-Integer Mathematical Modeling Approach to Exam Timetabling. *Comput. Manag. Sci.* **2010**, *7*, 19. [CrossRef]
2. Ünal, Y.Z.; Uysal, Ö. A New Mixed Integer Programming Model for Curriculum Balancing: Application to a Turkish University. *Eur. J. Oper. Res.* **2014**, *238*, 339–347. [CrossRef]
3. MirHassani, S.A.; Habibi, F. Solution Approaches to the Course Timetabling Problem. *Artif. Intell. Rev.* **2013**, *39*, 133–149. [CrossRef]
4. Miranda, J. EClasSkeduler: A Course Scheduling System for the Executive Education Unit at the Universidad de Chile. *Interfaces* **2010**, *40*, 196–207. [CrossRef]
5. Lach, G.; Lübbecke, M.E. Curriculum Based Course Timetabling: New Solutions to Udine Benchmark Instances. *Ann. Oper. Res.* **2012**, *194*, 255–272. [CrossRef]
6. Sørensen, M.; Dahms, F.H.W. A Two-Stage Decomposition of High School Timetabling Applied to Cases in Denmark. *Comput. Oper. Res.* **2014**, *43*, 36–49. [CrossRef]
7. MirHassani, S.A. A Computational Approach to Enhancing Course Timetabling with Integer Programming. *Appl. Math. Comput.* **2006**, *175*, 814–822. [CrossRef]
8. Boland, N.; Hughes, B.D.; Merlot, L.T.G.; Stuckey, P.J. New Integer Linear Programming Approaches for Course Timetabling. *Comput. Oper. Res.* **2008**, *35*, 2209–2233. [CrossRef]
9. Schaerf, A. A Survey of Automated Timetabling. *Artif. Intell. Rev.* **1999**, *13*, 87–127. [CrossRef]
10. Pillay, N. A Survey of School Timetabling Research. *Ann. Oper. Res.* **2014**, *218*, 261–293. [CrossRef]
11. Birbas, T.; Daskalaki, S.; Housos, E. School Timetabling for Quality Student and Teacher Schedules. *J. Sched.* **2009**, *12*, 177–197. [CrossRef]
12. Sánchez-Partida, D.; Martínez-Flores, J.L.; Olivares-Benítez, E. An integer linear programming model for a university timetabling problem considering time windows and consecutive periods. *J. Appl. Oper. Res.* **2014**, *6*, 158–173.
13. Alvarez-Valdes, R.; Crespo, E.; Tamarit, J.M. Design and Implementation of a Course Scheduling System Using Tabu Search. *Eur. J. Oper. Res.* **2002**, *137*, 512–523. [CrossRef]

14. Stallaert, J. Automated Timetabling Improves Course Scheduling at UCLA. *Interfaces* **1997**, *27*, 67–81. [CrossRef]
15. Daskalaki, S.; Birbas, T.; Housos, E. An Integer Programming Formulation for a Case Study in University Timetabling. *Eur. J. Oper. Res.* **2004**, *153*, 117–135. [CrossRef]
16. Dorneles, Á.P.; de Araújo, O.C.B.; Buriol, L.S. A Fix-and-Optimize Heuristic for the High School Timetabling Problem. *Comput. Oper. Res.* **2014**, *52 Pt A*, 29–38. [CrossRef]
17. Dimopoulou, M.; Miliotis, P. Implementation of a University Course and Examination Timetabling System. *Eur. J. Oper. Res.* **2001**, *130*, 202–213. [CrossRef]
18. Van den Broek, J.; Hurkens, C.; Woeginger, G. Timetabling Problems at the TU Eindhoven. *Eur. J. Oper. Res.* **2009**, *196*, 877–885. [CrossRef]
19. Müller, T.; Keith, M. Comprehensive approach to student sectioning. *Ann. Oper. Res.* **2010**, *181*, 249–269. [CrossRef]
20. Bettinelli, A.; Cacchiani, V.; Roberti, R.; Toth, P. An overview of curriculum-based course timetabling. *Top* **2015**, *23*, 313–349. [CrossRef]
21. Badri, M.A. A Two-Stage Multiobjective Scheduling Model for [Faculty-Course-Time] Assignments. *Eur. J. Oper. Res.* **1996**, *94*, 16–28. [CrossRef]
22. Hinkin, T.R.; Thompson, G.M. SchedulExpert: Scheduling Courses in the Cornell University School of Hotel Administration. *Interfaces* **2002**, *32*, 45–57. [CrossRef]
23. Ismayilova, N.A.; Sağir, M.; Gasimov, R.N. A Multiobjective Faculty–Course–Time Slot Assignment Problem with Preferences. *Math. Comput. Model.* **2007**, *46*, 1017–1029. [CrossRef]
24. Miranda, J.; Rey, P.A.; Robles, J.M. UdpSkeduler: A Web Architecture Based Decision Support System for Course and Classroom Scheduling. *Decis. Support. Syst.* **2012**, *52*, 505–513. [CrossRef]
25. Martinson, F.K. Fuzzy vs. Minmax Weighted Multiobjective Linear Programming Illustrative Comparisons. *Decis. Sci.* **1993**, *24*, 809–824. [CrossRef]
26. Schimmelpfeng, K.; Helber, S. Application of a Real-World University-Course Timetabling Model Solved by Integer Programming. *Oper. Res. Spectrum* **2006**, *29*, 783–803. [CrossRef]