

Research Article

Solving a Real Constraint Satisfaction Model for the University Course Timetabling Problem: A Case Study

Marco Antonio Cruz-Chávez,¹ Mireya Flores-Pichardo,¹ Alina Martínez-Oropeza,¹ Pedro Moreno-Bernal,¹ and Martín H. Cruz-Rosales²

¹Engineering and Applied Science Research Center, UAEM, Avenida Universidad 1001, Colonia Chamilpa, 62209 Cuernavaca, MOR, Mexico

²Faculty of Accounting, Administration & Informatics, UAEM, Avenida Universidad 1001, Colonia Chamilpa, 62209 Cuernavaca, MOR, Mexico

Correspondence should be addressed to Marco Antonio Cruz-Chávez; mcruz@uaem.mx

Received 10 February 2016; Revised 12 June 2016; Accepted 20 June 2016

Academic Editor: Miguel A. Salido

Copyright © 2016 Marco Antonio Cruz-Chávez et al. 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 paper proposes a real mathematical constraint satisfaction model which defines the timetabling problem in the Faculty of Chemical Sciences and Engineering (FCSE) at the Autonomous University of Morelos State, Mexico. A Constructive Approach Algorithm (CAA) is used to obtain solutions in the proposed model. A comparison is made between the CAA's results and the schedule generated by the FCSE administration. Using the constraint satisfaction model, it is possible to improve the allocation of class hours in the FCSE so that classroom use is more efficient.

1. Introduction

In the field of computer science there is an area known as algorithmic complexity, which is a theoretical measure available to compare different algorithmic solutions. It is possible to observe how they behave when attempting to solve a complex problem using these tools. This measure enables the selection of the most efficient algorithm for solving a problem. The complexity theory classifies the universe of problems according to the inherent complexity of solving them [1]. This depends on whether there is an exact algorithm in polynomial time behavior and whether there is proof of optimality for solving the problem in question. If there is such an exact algorithm, then the problem is easy to solve and it is called a P type polynomial problem. If the problem is not known to date to have exact polynomial time algorithm behavior, then the problem is difficult to solve and it is not polynomial; it is an NP type problem [2]. If this is the case, the best solution for dealing with an NP problem is to use computational heuristics which allow for constraints in polynomial time and yield a solution very close to the global optimum [3]. This

paper focuses specifically on the problem of the allocation of expensive resources (classrooms) in university scheduling. It is classified by complexity theory as an NP-hard problem, which is the most difficult type of problem in the NP set [3, 4]. It is important due to the widespread application in universities to manage class schedules for students and teachers.

The scheduling problem consists of generating schedules for defined tasks, searching for the best way to comply with specific conditions or user requirements, by both teachers and students. These problems are very common and are found in different types of educational environments, such as universities, colleges, institutes, departments [5], sports activities [6], transportation activities [7], and other activities involving people and the use of equipment [8, 9].

In the field of education, the efficient establishment and planning of schedules improve the use and organization of staff, schedules, classrooms, and equipment. The improved use of resources boosts academic performance for students and teachers.

The academic programming schedule is a specific problem within the general problem of resource allocation. This

scheduling problem is known in the scientific community as the University Course Timetabling Problem (UCTP) [10].

The staff is responsible for organizing the schedules of students and teachers during each academic period. It is important to provide the best possible solution, according to a number of restrictions which must be satisfied in order to ensure that the educational program meets quality measures; this implies a great effort.

The goal of the scheduling problem in education is to schedule the courses in an academic period (trimester, quarter, or semester). It is necessary to consider several variables that impact the problem, such as teachers required for all courses, which students need to take the courses, groups of students that take a set of courses, days and available periods of the individual classrooms, the number of classrooms available, the capacity of each room, and other relevant variables.

Resources that are considered critical or scarce are the classrooms, since they are not resources that are readily available. This is why the number, characteristics, and capacity of classrooms must be considered when scheduling, as one of the features and restrictions of the educational institution.

In the literature, there is information on theoretical models of UCTP, timetables, and algorithms to deal with these models. For example, in [11], a complete study of a theoretical model is presented which does not consider teachers as a variable. This model is treated with several computational heuristics such as genetic algorithms, ant colony, iterated local search, Simulated Annealing, and Tabu. Reference [12] proposes a new evolutionary algorithm-based timetabling tool for solving course scheduling problems. To evaluate the efficiency of this algorithm, they work with fourteen benchmarks of timetabling problems.

The UCTP has also been addressed using real models applied to universities. For instance, in [13], an evolutionary algorithm is applied in the Italian School System with the hybridization of two heuristics based on local search. In [14], a genetic algorithm is applied based on sectors, reporting good results when applied to a real case. Reference [15] presents a study of exam scheduling in Pahang University of Malaysia which includes comparing a constructive heuristic with existing software. This case study considers four sets of hard constraints and three sets of soft constraints. The authors indicate that the solutions obtained by the heuristics outperform those obtained by the software that was implemented at the university to manage all the restrictions in the case. In [16], a Simulated Annealing metaheuristic was implemented, and preliminary work is presented that deals with the problem of programming courses in Tamhidi University of Malaysia. The instance they work with has 800 students, grouped into five study programs, and students that attend the same courses. The authors work with three sets of hard constraints and three sets of soft constraints. In [17], an initial solution is obtained by a mathematical programming approach based on Lagrangian relaxation. The proposed method has been tested on instances from a university in Indonesia. Eleven requirements of the problem are considered. They are divided into two different categories; four requirements represent a set of teacher assignment requirements, while the rest are related to the course scheduling problem. Computational

experiments were performed on two real problem datasets from a university in Indonesia. Reference [18] developed three models of the University Course Timetabling Problem as used at Mkwawa University College of Education and tested using real data from the stated university. It was possible to obtain the optimal solution for real problem instances through reformulations of models which involve a mixture of binary and time-indexed variables. In China, [19] uses the greedy heuristic approach for UCTP. Complex multiconstraint conditions are considered, based on the uniform teaching resources of the university, which include students, teachers, and classrooms with feasible time slots, respectively. In particular, with respect to most of the graduate schools in China, there are more than 400 courses in every academic semester. A timetable should satisfy 9 basic requirements. The characteristics of datasets of two semesters from the graduate school are considered in the implementation. In [20], a mathematical programming model with an approach called meta-model is proposed. It can include implicit restrictions and implicit objective functions for the assignment of a set of temporary classes at Toyama Prefectural University. Each class during a specific period of time has hard and soft constraints. Initially, scheduling was done manually and took about 3 weeks, with a preparation time of 5 to 6 days. To validate the proposed mathematical model, algorithms apply branch and bound. Reference [21] presents a linear integer programming model that solves a timetabling problem in a Rio de Janeiro, Brazil, department. The model was designed to generate solutions that meet the preferences of the faculty's managers. More specifically, this meant assigning the maximum number of lecturers with the highest academic title while minimizing costs by merging courses with equivalent syllabuses.

In the Research Center for Engineering and Applied Sciences (CIICAp) of the Autonomous University of Morelos State (UAEM), two projects have been carried out with respect to scheduling university courses. In [22], a scheduler was programmed which provided a theoretical model of schedule generation using Simulated Annealing by applying benchmarks of a Metaheuristic Network [11] (medium 01–05, hard 01, and hard 02). Feasible solutions were obtained with greater efficiency and effectiveness than with other algorithms reported in [11] for medium and large instances. The second project involved the development of a basic system of administrative schedules for the Faculty of Chemical Sciences and Engineering (FCSE) at the UAEM with web interface, using a three-layer model, client, web service, and database. Through this project, a system of consultations of academic schedules for teachers and students via the internet was developed. A basic constraint model was also proposed to avoid overlapping subjects for students in the FCSE, but without taking into account restrictions on the allocation of teachers for the subjects they teach [23].

The present work proposes a constraint satisfaction model for the problem of programming university courses for the Faculty of Chemical Sciences and Engineering (FCSE). It increases the number of actual model restrictions for teachers, aiming to create a complete model for the slot allocation in FCSE. The administration of the Faculty of Engineering creates a timetable of courses manually. Usually there are

overlaps in the classes taken by students, which leaves them with the dilemma of which course to take. In addition, the students usually attend classes in a discontinuous schedule. The considerations described above detract in several ways from the quality of student learning in the FCSE.

Given the problem of schedule assignment in the FCSE, a constraint model is proposed which considers the real case of allowing the FCSE to offer courses during each period, satisfying in its entirety the restrictions of the proposed model. The contributions of this paper are the mathematic formulation of a real problem, and the results related to a real application.

This paper is divided into six sections. Section 1 is an introduction to the University Course Timetabling Problem and its application in universities. Section 2 describes the problem of allocation of slots in the Faculty of Chemical Sciences and Engineering (FCSE), in the city of Cuernavaca, Morelos, Mexico. Section 3 presents the model of the constraint satisfaction problem for the FCSE. Section 4 defines the instance solved in the allocation problem for the FCSE schedules. It also explains why the decision was made to use a Constructive Approach Algorithm (CAA) for generating feasible solutions that meet the constraints of the total allocation model for the scheduling algorithm of the FCSE. Section 5 shows the experimental results of feasible solutions obtained by the CAA and compares the solution obtained by the administration of the FCSE. Finally, Section 6 presents the conclusions.

2. University Course Timetabling Problem in the Faculty of Chemical Sciences and Engineering

The Faculty of Chemical Sciences and Engineering at the Autonomous University of Morelos State started in 1977 as part of the solution to growing needs of industries in the Morelos area. Currently, the FCSE offers majors in industrial chemistry, chemical engineering, industrial engineering, mechanical engineering, and electrical engineering.

The FCSE has access to 41 spaces including the following:

- (i) 21 classrooms located in the main building of the FCSE.
- (ii) 1 academic computing center (located on the top floor of the FCSE building).
- (iii) 2 industrial analysis laboratories (Chemistry Laboratories 3 and 5, both located on the top floor of the FCSE building).
- (iv) 2 Chemistry Laboratories (Chemistry Labs 1 and 2, both located on the ground floor of the FCSE building).
- (v) Basic multidisciplinary workshop (TAMULBA), which includes 1 physics lab and 10 classrooms.
- (vi) Unit Operations Laboratory (LOU): 1 Unit Operations Lab and 3 classrooms.

The completion of the degrees takes 10 semesters; students are permitted a maximum of 12 semesters. Students attend

classes either in the morning or in the afternoon shift. During the first two semesters, the students are in rigid groups and the classes are determined by the department administration. After those two semesters, the groups are flexible and students advance based on their ability. From the third semester on, each student can choose the subjects they wish to take, in which semester they take them, and with the professor of their choice. Students are admitted to the FCSE each year and begin study in September or February.

The enrollment of students in the FCSE is 1,500 on average but varies slightly each semester. To teach the courses offered every semester, the FCSE uses an average of 150 tenured teachers, plus others that are not tenured.

The FCSE performs manual scheduling, attempting to take into account the capacity of the classrooms and facilities, availability of full time professors, days and existing periods, the classes that students can choose, and the restrictions on choice of subjects.

The FCSE administration considers the following when generating class schedules:

- (i) A subject is formed by a subset of 4 or 6 events. The number of events that forms a subject depends on the subject itself. The events that are scheduled in a classroom must have the necessary infrastructure and the required facilities provided by the classroom (television, projectors, and other necessary electrical connections).
- (ii) A search is done to find the subjects and their events that can be scheduled in the periods and classrooms available for the week.
- (iii) An attempt is made to prevent classrooms from being underutilized. Classrooms are assigned according to the size of the student group. A small group of students needs a small space, not a large lecture hall. The classroom assigned must have the necessary capacity for the event.
- (iv) Tenured teachers can decide the day and period they teach their subjects. Because the assignment of tenured teachers is known and is chosen by them, their class schedules are known. The only assignment for tenured teachers is the classroom where they will teach. The use of the word tenured at the Autonomous University of Morelos State warrants definition, as it varies from its most common usage. In this university, tenure means that a particular teacher/professor is a permanent member of the FCSE. This person has also earned the rights to a certain course or courses, similar to being the owner of the course. No other person can teach the subject unless the tenured teacher requests a temporary leave, allowing someone else to teach the course only during the tenured teacher's absence. Upon return, the tenured teacher reclaims rights to the course.
- (v) Students should have continuous schedules; they should not attend classes in the morning and late in the day. The events of the same subject must be presented in consecutive periods if they appear on the

same day. The subjects taught are composed of 4 or 6 events. These events should be scheduled considering that half of the events that correspond to a subject should be taught on the same day in consecutive periods. For example, if a subject is composed of 4 events and event 1 is scheduled on day 1 in period 1, then event 2 must be scheduled consecutively in period 2 of day 1. If event 3 is scheduled in period 4, then event 4 must be scheduled consecutively in period 5 of the same day.

- (vi) A search checks for classroom allocation in which overlap of classes for students is avoided.
- (vii) The search for a classroom allocation attempts to avoid schedule overlap for tenured teachers. The subjects of nontenured teachers must be scheduled on days and periods that are free after programming the tenured teachers.

It has never been possible to generate an adequate allocation schedule in the FCSE which satisfies all the considerations defined in the allocation problem. This paper proposes a model of constraint satisfaction with a proposed solution using a Constructive Approach Algorithm (CAA) for scheduling resources in the FCSE. This allows class schedules to be generated which satisfy the toughest considerations currently taken into account, which the model calls restrictions. The schedules generated are proposed solutions to the timetabling problem in the FCSE.

The results of this work improve the process of generating class schedules and attempt to eliminate the current problems with compliance to restrictions. It also aims to decrease the process time it takes to schedule resources; manual scheduling requires up to two weeks to complete. In addition, there is an attempt to utilize the resources of the FCSE more efficiently, which in turn promotes student achievement.

3. Constraint Satisfaction Model for the Faculty of Chemical Sciences and Engineering

In the constraint model, several sets of variables are taken into account, such as the set of events $E = \{1, 2, 3, \dots, n_E\}$, the set of events that require laboratory equipment $E_l = \{1, 2, 3, \dots, n_{E_l}\}$, the set of time slots $T = \{1, 2, 3, \dots, n_T\}$, the days $D = \{1, 2, 3, 4, \dots, n_D\}$, the set of periods $P = \{1, 2, 3, \dots, n_P\}$, the set of classrooms $R = \{1, 2, 3, \dots, n_R\}$, the set of classrooms that have laboratory equipment $R_l = \{1, 2, 3, \dots, n_{R_l}\}$, student enrollment $S = \{1, 2, 3, \dots, n_S\}$, teacher enrollment $M = \{M_T, M_N\}$ which is composed of two subsets, the enrollment $M_T, M_T = \{1, 2, 3, \dots, m_T\}$ of tenured teachers and $M_N; M_N = \{1, 2, 3, \dots, m_N\}$ of nontenured teachers, the set of classes $C = \{1, 2, 3, \dots, n_C\}$, the subset of events in each class $c_i, c_i = \{e_i, e_j, \dots\}$, and finally the set of facilities $F = \{1, 2, 3, \dots, n_F\}$.

The constraint satisfaction model is as follows:

$$\sum_{t=1}^{n_T} \sum_{r=1}^{n_R} X(t, R(r, e)) = 1 \quad \forall e, \quad (1)$$

$$\sum_{t=1}^{n_T} \sum_{r=1}^{n_R} X(e, \Phi(t, r), R(r, e)) = 2 \quad \forall e, \quad (2)$$

$$\sum_{e=1}^{n_E} X(e, \Phi(t, r)) \leq 1 \quad \forall (t, r), \quad (3)$$

$$\sum_{s=1}^{n_S} Z(s, X(e, \Phi(t, r))) \leq \text{Capacity}(r) \quad (4)$$

$$\forall (e, t, r),$$

$$\sum_{t=1}^{n_T} \sum_{r=1}^{n_R} X(e_{tt}, \Phi(t, r), R(r, e_{tt})) = 2 \quad \forall e_{tt}, \quad (5)$$

$$\sum_{e=1}^{n_E} \sum_{r=1}^{n_R} Z(s, X(e, \Phi(t, r))) \leq 1 \quad \forall (s, t), \quad (6)$$

$$\sum_{e=1}^{n_E} \sum_{r=1}^{n_R} Z(m, X(e, \Phi(t, r))) \leq 1 \quad \forall (m, t), \quad (7)$$

$$\sum_{t=1}^{n_T} \sum_{r_l=1}^{n_{R_l}} X(e_l, \Phi(t, r_l), R_l(r_l, e_l)) = 2 \quad \forall e_l, \quad (8)$$

$$(t, r, e, s, m) \in N.$$

The model of constraint satisfaction is subject to eight sets of hard constraints, which have to be satisfied so the instance solution obtained is feasible.

A feasible solution is found when events are assigned to time slots. Each available room has a set of time slots, in this case n_T time slots. A time slot t is defined in a 2D space as a function of coordinates (d, p) . Therefore, $T = \{t \in N^2 \mid t = f(d, p), t = 1, 2, 3, \dots, n_T\}$. The full assignment of events takes place in a 3D space, as a function of $\Phi(t, r)$, where $r \in R$. Then, for each $\Phi(t, r)$,

$$\Phi(t, r) = \begin{cases} 1, & \text{If assigned to an event} \\ 0, & \text{Otherwise.} \end{cases} \quad (9)$$

Classroom $r \in R$ must satisfy the needs of event $e \in E$ that occurs in the time slot $t \in T$.

The set of constraints in (1) implies that classroom $r \in R$ must satisfy event facilities $e \in E$ that occur in the time slot $t \in T$. For each classroom $r \in R$ and event $e \in E$, the following is defined:

$$R(r, e) = \begin{cases} 1, & \text{If } r \text{ satisfies facilities of } e \\ 0, & \text{Otherwise.} \end{cases} \quad (10)$$

Each and every one of the events $e \in E$ has to be scheduled. Each event must be scheduled in a time slot $t \in T$ in a classroom.

The set of constraints in (2) require that each and every one of the events must be programmed. Each event must be scheduled in a time slot of a classroom with adequate facilities.

$$X(e, \Phi(t, r), R(r, e)) = \begin{cases} 1, & \text{If } \Phi(t, r) + R(r, e) = 2 \\ 0, & \text{Otherwise.} \end{cases} \quad (11)$$

Classroom $r \in R$ may allow only one event $e \in E$ for each time slot $t \in T$. This implies that, in time slot $t \in T$ for classroom $r \in R$, a maximum of one event may be programmed.

The set of constraints in (3) requires that a room admit only one event per time slot. This implies that, for time slot $t \in T$ in classroom $r \in R$, a maximum of one event can be programmed.

Classroom $r \in R$ must have enough capacity to hold the group of students $s \in S$ participating in the event $e \in E$. The capacity of each classroom $r \in R$ is input data.

To find when student s attends an event, the function presented in (12) is defined. This function indicates that student s attends the event that occurs in time slot t in classroom r . The restrictions in (4) must be met; the classroom must have sufficient capacity to serve the group of students who attend the event $e \in E$. The capacity of each classroom r is input data:

$$Z(s, X(e, \Phi(t, r))) = \begin{cases} 1, & \text{If } X(e, \Phi(t, r)) = 1 \text{ Then } s \text{ serves } e \\ 0, & \text{Otherwise.} \end{cases} \quad (12)$$

Each and every one of the events given by MT teachers will be scheduled in the first instance, as a partial assignment in the requested periods.

In constraint set (5), the events of the tenured teachers (tt) are programmed. The assigned classroom must have the facilities required by the teacher. This gives a partial scheduling.

A student $s \in S$ must not have overlapping events.

The set of constraints (6) avoids the overlap of events attended by the student $s \in S$. As indicated by function (12), if the student attends event e in classroom r , then this student will not attend another event e' during the same time slot t . Therefore, $Z(s, X(e', \Phi(t, r))) = 0, \forall e' \in E$ with $e' \neq e$. Restriction (6) indicates that student s may not have overlapping events:

$$Z(m, X(e, \Phi(t, r))) = \begin{cases} 1, & \text{Si } X(e, \Phi(t, r)) = 1 \text{ Then } m \text{ attends } e \\ 0, & \text{Otherwise.} \end{cases} \quad (13)$$

A teacher $m \in M$ must not have overlapping events.

The set of constraints (7) prohibits overlap of events taught by teacher $m \in M$. Function (13) ensures that if teacher m attends event e in classroom r , then the teacher cannot attend another event e' in the same time slot t . This means

TABLE 1: Symbolic representation of a timetable.

(d, p)	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
1	$t1$	$t2$	$t3$	$t4$	$t5$	$t6$
2	$t7$	$t8$	$t9$	$t10$	$t11$	$t12$
.
.
.
15	$t85$	$t86$	$t87$	$t88$	$t89$	$t90$

that $Z(m, X(e', \Phi(t, r))) = 0, \forall e' \in E$ with $e' \neq e$. Restriction (7) indicates that teacher m cannot be present in overlapping events.

In the set of constraints (8), events e_l which correspond to lab classes must meet in classrooms with laboratory equipment; the assigned classroom must meet the needs of the teacher (10).

The values that the variables t, r, e, s , and m can take must be natural numbers.

The set of teachers is divided into two subsets, $M = M_T \cup M_N$. The set of time slots is also divided into two subsets: $T = T_T \cup T_N$. The time slots t that form part of T_T are assigned to tenured teachers M_T , who teach classes at their request. Each tenured teacher requests a specific time slot t in which to teach his or her class, $M_T = \{m \mid \forall m \exists t \in T_T, t = \text{Assigned}(m)\}$. The assignment of t to m is input data. For all time slots T_N , which will be assigned to nontenured teachers M_N , the specific assignment of t for each nontenured teacher m_n is not known. Given the above, the model of the FCSE begins with a feasible partial solution, because the allocation of T_T is known and is selected by all tenured teachers M_T in the timetable of classes for the semester. All that remains to assign to M_T is the classroom r where the event will be held.

The objective is to find an instance solution of the constraint satisfaction model of the Faculty of Chemical Sciences and Engineering (FCSE) that minimizes the value of the objective function. An instance of the problem occurs in each semester of the school year. The instance is generated based on the supply of subjects, subjects selected by each student, and the enrollment of students and teachers.

3.1. Symbolic Representation of the Problem Instance for the FCSE. A representation of the *University Course Timetabling Problem* is constructed using a 3D matrix. The process begins with a 2D matrix, which represents an available classroom and is comprised of a set of time slots, 90 slots in this case. A time slot t is defined as a 2D space based on the coordinates (day, period), as seen in Table 1.

The scheduling of events for a classroom should utilize one of the 90 time slots available, so there is a 2D array for each of the 41 classrooms (including laboratories) available in the FCSE. Table 1 shows the symbolic representation for each classroom.

In Table 1, it is possible to observe the periods of time when there should be events programmed in a classroom. The two dimensions observed are made up of days d and periods

of time p in which the FCSE works. Since the instance has 15 periods and 6 days, there are 90 time slots in each classroom, numbered t_1 to t_{90} , as shown in Table 1.

Table 2 shows an example of an event map for a classroom that includes the variables in the FCSE case with 6 days (Monday, Tuesday, Wednesday, Thursday, Friday, and Saturday), 15 periods ($p = 1$ to $p = 15$), and the assignment of 18 events corresponding to eight subjects identified with ID (15, 29, 49, 50, 120, 212, 332, and 335). As the events are scheduled in the time slots, the identifier numbers ID of the events are placed in the fields corresponding to the assigned periods. For example, in Table 2, the field with ID 120 consists of 4 events. The time slot spaces which are blank in Table 2 represent the time periods available for scheduling other events.

When making the event schedule for the different classrooms, the variable r (classrooms) creates a third dimension to form the set of tables corresponding to the set of available classrooms, as shown in Figure 1. Therefore, with the set of classrooms, a tridimensional structure is obtained with coordinates for each time slot $t = (d, p, r)$. To allocate the total set of FCSE events E in slots available t in the 3D matrix, all of the hard constraints are satisfied. The hard constraints (1 to 8) were presented in this section in the constraint satisfaction model. Then a solution for the UCTP for FCSE is generated.

Upon characterizing the variables of the case study FCSE, there are 6 days (M, TU, W, TH, F, and SA), 15 periods (Period 1 to Period 15), and 41 classrooms ($r = 1$ to $r = 41$). This yields an area of 3060 ($6 \times 15 \times 34$) time slots available in classrooms and 630 ($6 \times 15 \times 7$) time slots available in laboratories, where only classes requiring laboratory equipment are taught.

Figure 1 presents an instance of 3 events and 2 classrooms for the University Course Timetabling Problem, using a bipartite graph model. The graph consists of the set of events on one side and the assignments $t = (r, d, p)$ which can occupy events on the other side. In this instance, the problem only has 2 days d and two periods p . Each classroom r has a capacity $A(k)$ and can be assigned a single event in each t . The event e_i that is assigned a t indicates that the event will be held in classroom r on day d and during period p . The objective is to determine t for each e_i in the bipartite graph, taking into account the capacity $A(k)$ of classroom r . Figure 2 presents a solution to the problem presented in Figure 1. For example, event e_1 is assigned on day d_1 during period p_1 , to classroom r_1 , which has a capacity $A(1)$. Naturally, to assign events to classrooms, the constraints present in the constraint satisfaction model presented in (1) to (8) must be taken into account.

4. Definition of the Problem Instance for the Faculty of Chemical Science and Engineering (FCSE)

The parameter values for the instance of the FCSE problem are defined according to a scheduling assignment by the FCSE administration. They are applied as a solution in an academic semester. Naturally, this applied solution is not feasible for that semester given the amount of overlap found in the schedules of the students who attended. The considered instance

TABLE 2: Abstraction of a timetable. FCSE case.

(d, p)	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
1	49			332	50	
2	49		15	332	50	
3			15		29	335
4			15			335
:						
12		120			212	
13		120			212	
14		120				
15		120				

TABLE 3: Parameter instance for fall semester (August to December) 2015 in the FCSE.

Parameters	Value
Events ($n_E + n_I$)	1514
Classrooms ($n_R + n_{R_i}$)	41 (classrooms and laboratories)
Students	1426
Days	6
Periods	15
Time slots	90
Facilities	7
Tenured teachers	156

TABLE 4: Classification of instances.

Concept	Small	Medium	Large
Events	100	400	400
Classrooms	5	10	10
Features	5	5	10
Students	80	200	400
MNSE	20	50	100

corresponds to the schedule for the August–December 2015 semester. The referred values were obtained from the field study in the FCSE with information provided by the person responsible for the administrative scheduling of the FCSE. The definition of the instance is presented in Table 3.

Based on their research and theoretical evidence of UTTP instances, the Metaheuristic Network group, [11], provided a ranking of the instances or benchmarks collected in three classes depending on the degree of difficulty and size. According to the number of resources in the program, the instances are classified as small, medium, and large [11]. The characteristics of the three classes of instances are shown in Table 4, where MNSE is the maximum number of students per event.

The instance of the FCSE has 1514 events, 41 classrooms, 7 features, and 1426 students. In addition, it has 156 tenured teachers who teach events that should be scheduled in the first instance. Given this information, it can be observed that the instance of the FCSE is a size that exceeds the largest classification of the Metaheuristic Network. The number of

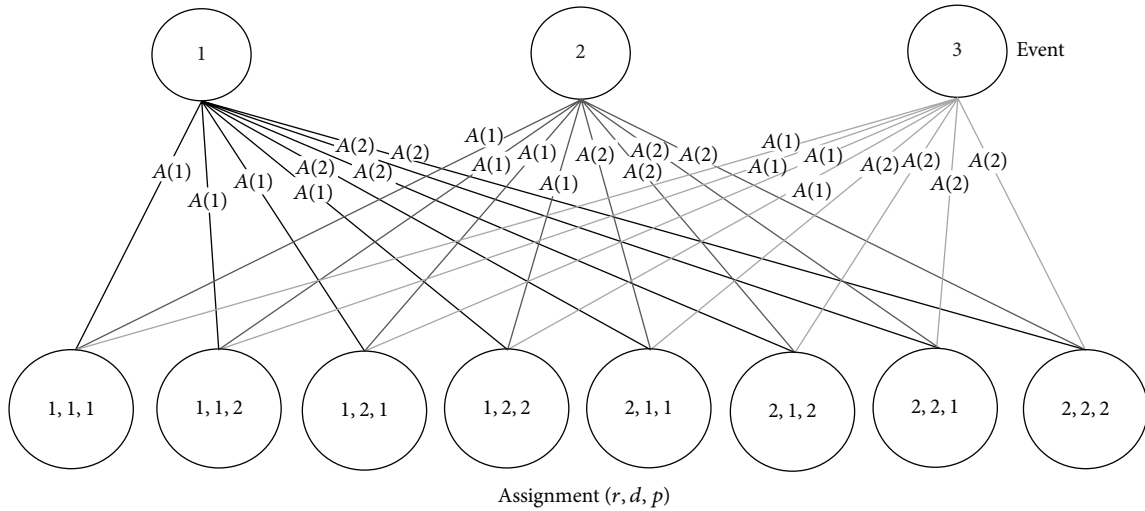


FIGURE 1: Bipartite graph model for University Course Timetabling Problem.

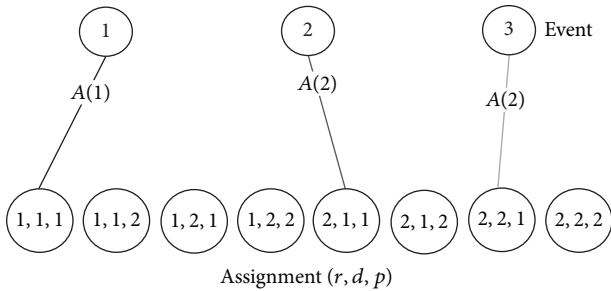


FIGURE 2: A solution to the University Course Timetabling Problem presented in Figure 1.

hard constraints in the case study is greater than the number of hard constraints contemplated by [11].

4.1. Format of the Problem Instance. For the execution of the FCSE problem instance, use of the CAA (or any other heuristic or metaheuristic) that obtains feasible solutions requires that the input data, or instance of the problem, must be stored in text files in the proper format so they are readable. The same format is generated and then applied to the benchmarks presented in [11]. Since the number of instance data is very large, the instance must be stored in more than one file. The instance data of the problem is organized into 6 files with the format shown below.

Text File 1 (Defining the Problem)

- (i) *First line:* for the number of events, number of classrooms, number of features, number of students, all numbers are integers.
- (ii) *One line for each classroom:* each line indicates the capacity of a classroom.

Table 5 shows an example of the FCSE instance. In this example, it can be observed that the first line shows the number of events, number of classrooms,

number of features, and number of students, in that order. From line 2 to line 42, the capacity of each of the 41 classrooms of the FCSE is presented. The only information that appears in text file 1 is the coding.

Text File 2 (Assigning Events per Student)

- (i) *One row for each student/event:* coding is done with a zero or a one. A zero indicates that the student does not attend the event; a one indicates that the student attends the event. Table 6 shows an example for 2 students and 4 scheduled events. In this example, it can be seen that the first student attends event number two and the second student attends events one and two. All that appears in text file 2 is the coding.

Text File 3 (Assigning Events per Teacher)

- (i) *One line for each teacher/event.* Coding is done with a zero or a one. Zero indicates that the teacher does not teach the event; one indicates that the teacher does teach the event. Table 7 presents an example with 2 teachers and 4 scheduled events. In this example, it can be observed that the first teacher attends event number two and the second teacher attends events one and two. All that appears in text file 3 is the coding.

Text File 4 (Tenured Teachers' Hours)

- (i) *One line for each schedule/event.* This coding is four columns containing integers. Table 8 presents the coding for assigning events to each tenured teacher. The first column indicates the ID of the tenured teacher who requests to teach a specific class in the schedule. The second and third columns indicate the day (Monday = 1, ..., Friday = 5) and period (period = 1, ..., period = 15) in which a tenured teacher requests to teach the event. Numbers other than zero indicate that a tenured teacher has requested to teach the

TABLE 5: Coding to define the problem of the FCSE.

Line	Classroom capacity	Event number	Classroom number	Feature number	Student number
1	—	1514	41	7	1426
2	10	—	—	—	—
:	:	—	—	—	—
42	60	—	—	—	—

TABLE 6: Coding for assigning events per student.

Student	Coding	Event number
1	0	1
1	1	2
1	0	3
1	0	4
2	1	1
2	1	2
2	0	3
2	0	4

TABLE 7: Coding for assigning events per teacher.

Teacher	Coding	Event number
1	0	1
1	1	2
1	0	3
1	0	4
2	1	1
2	1	2
2	0	3
2	0	4

TABLE 8: Coding for assigning events to each tenured teacher.

ID teacher	Day	Coding		Subject ID
		Period		
1	2	1		1
	2	2		1
	4	1		1
	4	2		1
	0	0		2
	0	0		2
	0	0		2
	0	0		2

event at the scheduled time. A zero indicates that the tenured teacher has not requested to teach the event. The fourth column shows the ID of the field to which the event belongs. Table 8 presents an example of eight events scheduled for the tenured teacher with ID Teacher = 1. The teacher asks to teach two subjects, one with subject ID = 1 and the other with subject ID = 2; each subject has four events. The teacher asks

TABLE 9: Coding for the assignment of characteristics for each classroom.

Classroom ID	Coding Characteristic
1	0
1	1
1	0
1	0
2	1
2	1
2	0
2	0

that two events of subject ID = 1 be scheduled on Tuesday in the first and second period. The teacher also requests that the other two events of subject ID = 1 be on Thursday in the first and second period. For subject ID = 2, the teacher does not require a special schedule and adjusts to the schedule assigned. Only the coding appears in text file 4.

Text File 5 (Facilities by Classroom)

- (i) *One line for each class/feature:* Table 9 presents the coding for the assignment of features for each classroom. The first column indicates the ID of the classroom. The second column shows the characteristics of a classroom. A zero means the classroom does not satisfy the characteristic; a one means it does.

For example, in Table 9, column one has three classrooms (ID = 1, 2). In column two, it is shown that there are four characteristics that the classroom could satisfy. In the case of classroom ID = 1, it only satisfies characteristic number 2. Classroom ID = 2 satisfies characteristics 1 and 2. Only the coding appears in text file 5.

Text File 6 (Requirements by Event)

- (i) *One line for each event/need:* Table 10 presents the coding required to allocate for each subject based on the events that comprise the subject. The first column indicates the ID of the subject. The second column indicates what the class events need as related to the characteristics of a classroom. A zero is coded if the event does not need the characteristic, a one is coded if the event does need the characteristic. For example,

TABLE 10: Coding for allocating needs for each class.

Class ID	Coding Need
1	0
1	1
1	0
1	0
2	1
2	1
2	0
2	0

in Table 10, column one has two subjects (ID = 1, 2). In column two, it is shown that there are 4 characteristics which may be required of a subject. For the events that comprise subject ID = 1, only characteristic number two is needed. For the events that comprise subject ID = 2, characteristics 1 and 2 are needed. Only the coding appears in text file 6.

The Constructive Approach Algorithm was programmed to generate feasible solutions which satisfy the hard constraints (presented in Section 3) in the model of the Faculty of Chemical Science and Engineering. The constraint satisfaction model can be solved with academic tools (solvers), including adapted solvers designed to solve optimization models. In this case, the proposed model does not include the objective function that permits optimization. For simplicity's sake, in this work, the Constructive Approach Algorithm was used to obtain solutions to the proposed model and to compare with proposed solutions to the FSCE administration. In Appendix, the Constructive Approach Algorithm for generating feasible solutions is presented.

5. Experimental Results

Experimental tests of the Constructive Approach Algorithm were performed on a personal computer with Intel Core i7 2.2 GHz CPU, 8 GB RAM, SO Windows 7 Home Premium, Microsoft Visual Studio 2008 Ver. 9.0.21022.8. For each experimental trial, 30 executions of the CAA algorithm feasible solutions generator were performed.

The efficiency analysis of the algorithm to generate feasible solutions is shown in Figure 3. The times are shown that it took the algorithm to obtain each of the 12 feasible solutions and run 30 tests. The shortest time required to find a feasible solution was 3 hours and 33 minutes and the longest was 6 hours and 22 minutes. There was a feasible solution obtained in 40% of the executions. In the cases where no feasible solution was obtained, the time it took to perform the total number of iterations before arriving at the stop criteria for the algorithm ranged from 4 to 8 hours. The path of the CAA is always different because it is a nondeterministic algorithm. Therefore, the time it takes to perform operations to find partial solutions will be different, mainly due to the iterative search which checks for overlaps of students and teachers.

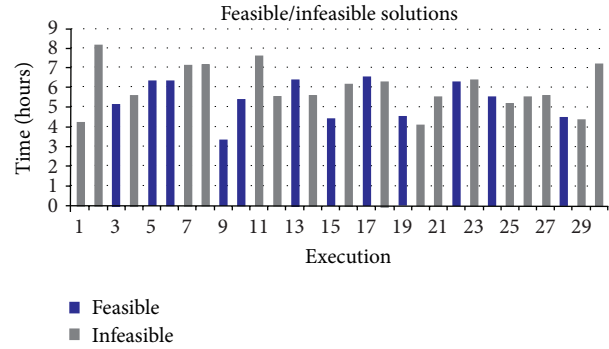


FIGURE 3: Solution generation time for 30 executions of the Constructive Approach Algorithm.

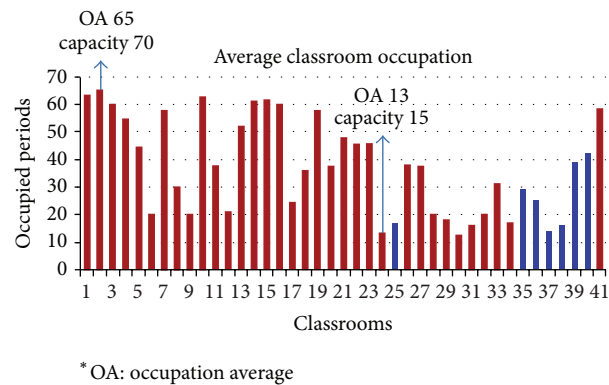


FIGURE 4: Average occupancy periods per classroom.

5.1. Frequency Allocation of Events in Classrooms. Figure 4 presents the average occupancy per classroom period of the 41 classrooms that are considered in the instance of the FCSE. The graph shows that no classroom is occupied in all 90 available periods. The highest average occupancy was classroom 2 with 65 of the 90 periods occupied, that is, 77.8%. This classroom has a capacity of 70 students. It was observed that classrooms with greater capacity have higher average occupancy.

The blue bars displayed in Figure 4 identify classrooms that are laboratories. They always have the same number of busy periods, ranging between 14 and 42. Only subjects that require laboratory equipment can be taught in these classrooms; they cannot be employed to deliver a lecture as they lack the infrastructure for such an activity.

The existence of seven laboratories and several small-capacity classrooms decreases the number of periods available for event scheduling. The seven laboratories have 630 (7 * 90) available periods but were only occupied in 203 periods, because they can only be occupied when the class requires a laboratory. There are 18 small-capacity meeting classrooms which hold between 15 and 30 students (see Figure 7). For classrooms with small capacity, occupancy averages between 13 and 38 periods, leaving between 77 and 52 unoccupied periods. For example, in Figure 4 it can be observed that classroom 24, with a capacity of 15, has only 13 out of 90 periods scheduled. The 18 small-capacity classrooms

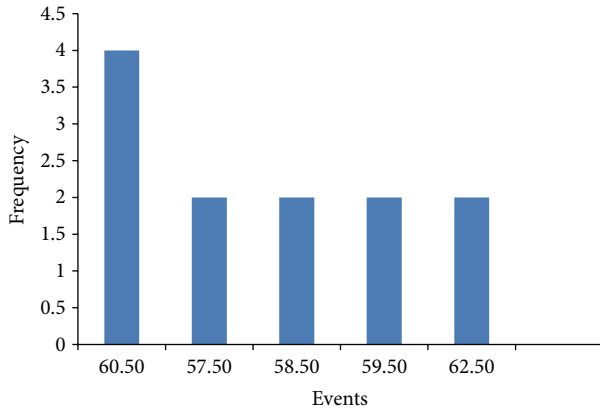


FIGURE 5: Average weekly events assigned to 70-student capacity classrooms in 12 feasible executions.

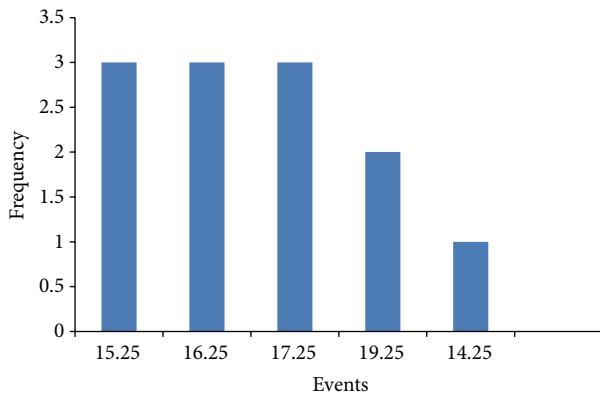


FIGURE 6: Average weekly events assigned to 15-student capacity classrooms in 12 feasible executions.

are almost unusable because most of the events require classrooms with capacities larger than what they can provide.

Figure 5 presents the Pareto chart for implementing the CAA for the 12 feasible solutions achieved in 30 executions. The Pareto chart shows the average frequency of event programming for the 9 larger classrooms which have a maximum capacity of 70 students. It is observed that, on average, for each large classroom, the CAA programs approximately 58 events per week in two executions, 59 events per week in two executions, 60 events per week in two executions, 63 events per week in two executions, and 61 events per week in four executions. Each scheduled event in a large classroom cannot exceed the maximum capacity of 70 students. Since the maximum number of events that can be scheduled in each classroom is 90 per week (see Table 1), for the nine 70-student capacity classrooms there is on average a 33% vacancy rate per week.

Figure 6 presents the Pareto diagram for implementation of the CAA for the 12 feasible solutions achieved in 30 executions. The Pareto chart shows the average frequency of event programming in the 4 small classrooms that have a maximum capacity of 15 students. It is observed that, on average, for each small classroom, the CAA programs less than 15 events per week in one execution, less than 16 events per week in three

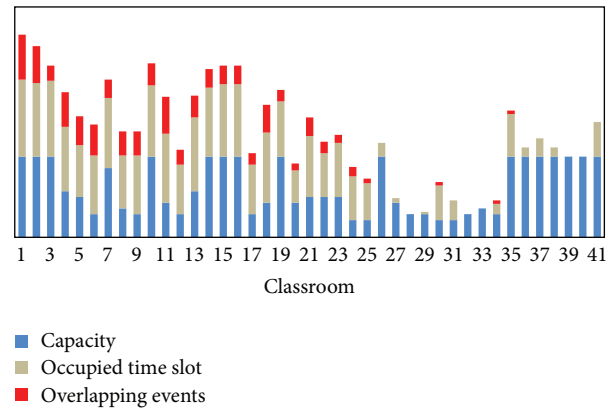


FIGURE 7: Solution obtained for the UCTP of the FCSE. FCSE administration. Semester, August–December 2015.

executions, less than 17 events per week in three executions, less than 18 events per week in three executions, and less than 20 events per week in two executions. Each scheduled event in small-capacity classrooms cannot exceed the maximum capacity of 15 students. Since the maximum number of events that can be programmed in a classroom is 90 per week, it can be concluded that, on average, the 4 small classrooms with seating for 15 people have close to a 78.6% vacancy rate.

5.2. Comparison with the Solution Obtained by the FCSE Administration for the Study Period. The solution generated by the administration of FCSE has the following characteristics. Of the 656 tenure events, 3 could not be assigned to the periods requested by tenured teachers. They were assigned to other periods; thus constraints set 5 was not satisfied. Among the 858 events of nontenured teachers, there were 28 events assigned during busy periods, so constraints set 3 was not satisfied. From this analysis, it can be observed that the solution obtained by the FCSE administration was not feasible. Analyzing further, it was observed that, of the 1514 events scheduled in this instance, there were 459 events which presented overlap (see Figure 7). Constraints set 7 was not satisfied. All other constraints sets were met. If events that overlap are part of 4-event classes, there are 114 classes with overlap. If the events that overlap are part of 6-event classes, there are 76 classes with overlap. The number of overlapping areas in the solution proposed by the administration of the FCSE is between 76 and 114 classes for the semester from August to December 2015. The solution obtained by FCSE allows overlap of classes, though it is not possible for students to attend all of their classes if there is overlap.

The solution obtained by the CAA algorithm presents the following results: 656 tenure events, all of which could be assigned to periods requested by the teachers, thus satisfying constraints set 5. Of the 858 nontenure events, all events were assigned to unoccupied periods, thus satisfying the constraints set 3. All other sets of restrictions were also enforced. The solution obtained by the proposed CAA for FCSE is feasible. Therefore, there is no overlap of events for students.

Figures 7 and 8 present the occupancy of classrooms and the behavior of the overlap, allowing a comparison between

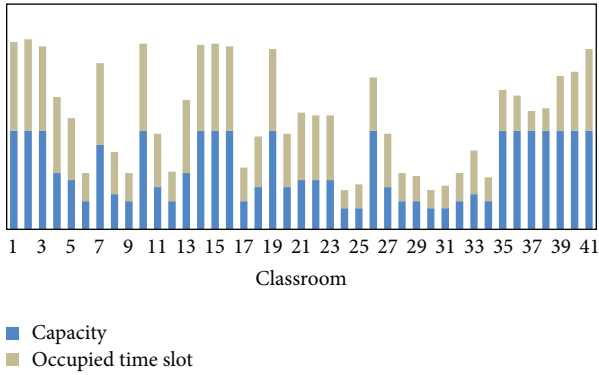


FIGURE 8: Solutions obtained for the UCTP of the FCSE. Constructive Approach Algorithm (CAA). Semester, August–December 2015.

the solutions generated by the FCSE administration versus the CAA. For the solution generated by the administration (Figure 7), a total of 459 events showed overlap for students. When a period (slot) in a classroom is occupied, it is understood that the period is occupied by an event. So in classroom 1, there are 67 (occupied time slot) events held during the week, of a total of 90 slots available in the classroom. Of those 67 events, 39 are overlapping. The capacity of classroom 1 is 70 students per event. There are only 13 classrooms where overlapping events do not appear in the administration’s solution (26, 27, 28, 29, 31, 32, 33, 36, 37, 38, 39, 40, and 41). It can also be seen that there are five classrooms in which events were not assigned (28, 32, 33, 39, and 40). The solution generated by the Constructive Approach Algorithm does not have overlapping events (Figure 8). In the solution proposed by the CAA, it can be seen that all classrooms are occupied. No classroom use reached its maximum limit (90 slots). The classrooms with smaller capacity have the fewest of events. For example, classrooms 24, 25, 30, and 31 have 13, 17, 13, and 16 events scheduled, respectively. Each of these classrooms has a capacity of 15 students.

6. Conclusions

The proposed mathematical model in this paper considers the performance of the hard constraints identified in the real case of university programming courses in the FCSE.

It was difficult to obtain solutions that complied with all the constraints of the proposed mathematical model. Only 40% of the 30 tests performed found feasible solutions. The solutions were found in an average of 7 hours each, which indicates the complexity of the hard constraints of the proposed model. This is compared to the infeasible solution obtained by the administration of FCSE after about two weeks of work.

The mathematical model proposed is the one that has been implemented manually in the FCSE for over 20 years. Due to the growth in student enrollment and additional faculty required, the number of schedule overlaps for subjects has increased. It has been more difficult for the administration of the FCSE to find good solutions. In addition, the number of classrooms that have been built in recent years is

not proportional to the increase in demand. The proposed mathematical model provides an opportunity to implement efficient scheduling of university courses without overlap in the FCSE. This model can be solved by any heuristic technique. In this case, because it is relatively simple and easy to analyze, a Constructive Approach Algorithm was used.

In the analysis of the results, it is observed that the classrooms with smaller capacity are underutilized. One proposal to improve the allocation of these classrooms is the formation of various small groups of students in classes, instead of the current large groups. The formation of small groups would generate a greater number of subjects (events) to program, which would in turn require either a greater enrollment of teachers or teachers who teach a greater number of subjects. It would facilitate the allocation of small classrooms to increase the use of these. The student enrollment could then increase in the FCSE, because it would still be possible to obtain feasible solutions without investing in the construction of additional classrooms for the new smaller groups. Another proposal is to use other heuristic techniques to solve the proposed model with greater efficacy. This could facilitate finding appropriate solutions according to the capacity of the classrooms. A change would need to be made to the proposed mathematical model so that new constraints needed to be included. The change would cause the heuristics to occupy the classrooms with less capacity first and then use the classrooms with higher capacity.

Appendix

Constructive Approach Algorithm Generates Feasible Solution (See Algorithm 1)

The description is as follows:

- (i) *Line (1)*. The algorithm starts with the instruction to read the input instance. The instance information is obtained from the 6 text files. The coding of these 6 text files is explained in Section 4.
- (ii) *Line (2)*. Before performing the schedule of events, items are classified by type (taking into account tenured teachers) and features (considering the needs of events).
- (iii) *Line (3)*. The cycle for $i = 0$ to $NumSoluciones$ do verifies the condition for each execution cycle to process the number of solutions ($NumSoluciones$) that are generated.
- (iv) *Line (4)*. With the instruction *initialize to zero arrays and empty structures*, the structures that are used to manipulate data during the creation of a solution are initialized. Three-dimensional arrangements permit the solutions to be saved. These arrangements are necessary due to the organization of data that is to be saved (see Figure 1). Such information can then be used to obtain schedules for classrooms.
- (v) *Lines (5)–(10)*. The schedule of events begins with the performance of hard constraint (RD) number (5) of the model (see Section 3). The schedule of events

```

(1) Read input instance
(2) Event classification (type, characteristics)
(3) for  $i = 0$  to NumSolutions do
(4)   Initialize to 0 and empty arrays
(5)   while TenuredEventsNotScheduled do
(6)     Assign tenuredEvents in empty slots satisfying hard constraints (HC)
(7)     Assign unscheduled events (uE) in temporary arrays
(8)     if (uE)
(9)       Assign uE in empty slots randomly satisfying HC*
(10)   endwhile
(11)   while  $j < \text{maxStages}$  or allEventsScheduled do
(12)     Sort nontenureEvents (capacity)
(13)     Assign nontenureEvents in empty slots randomly satisfying RD
(14)     Assign unscheduled nontenureEvents (uNTE) in temporary arrays
(15)     while  $j < \text{maxAttempts}$  do
(16)       for  $k = 0$  to uNTE do
(17)         Assign uNTE in empty slots without validating RD
(18)       endfor
(19)       for  $m = 0$  to events scheduled in empty slots without validating RD do
(20)         while  $s < \text{maxAttemptsB}$  do
(21)           Select events scheduled in empty slots without validating RD
(22)           Select Random temporalSolution occupied or unoccupied slots
(23)           Swap or Insert events in empty slots if events satisfy RD
(24)         endwhile
(25)       endfor
(26)       Reschedule events
(27)     endwhile
(28)   endwhile
(29)   if temporalSolution feasible
(30)     Print Feasible Solution and time in seconds
(31)     Save solution
(32)   else Print Infeasible Solution and time in seconds
(33) endfor

```

ALGORITHM 1: Constructive Approach Algorithm for generating feasible solutions for the FCSE pattern.

that will be taught by tenured teachers (line (6)) is performed. An attempt is made to schedule events on the day and in the period that the tenured teachers requested to teach the events of their corresponding subjects. A random assignment of a classroom is made and events are scheduled on the day and at the time requested only if the hard constraints are satisfied. Otherwise (line (9)), a maximum number of attempts are made to schedule the events into time slots randomly chosen to satisfy hard constraints. Once all the events of tenured teachers have been scheduled, the result is a feasible partial schedule.

- (vi) *Lines (11)–(27)*. The rest of the events that have been classified as pertaining to nontenured teachers are then scheduled. Do this for a maximum number of attempts or until a feasible solution is found, which is the stop criteria of the algorithm. To find a solution, the activities on lines (12) to (25) are performed.
- (vii) *Line (12)*. The events are ordered taking into account the capacity (number of students attending the event). These events are arranged in decreasing order according to their capacity. This is done so that the events

that are scheduled first are those that require classrooms with greater capacity. The events scheduled lastly are those which require less capacity.

- (viii) *Line (13)*. The events to be scheduled are assigned randomly in empty time slots that satisfy the hard constraints. Through a random selection of classrooms and time slots (day, period, and classroom), as seen Figure 1, an attempt is made to program as many nontenured teacher events as possible, while satisfying the hard constraints.
- (ix) *Line (14)*. The events that cannot be assigned in a maximum number of attempts are stored in the temporary arrangement uNTE.
- (x) *Lines (16) and (17)*. The events stored in uNTE are programmed in empty slots without verifying whether they satisfy the hard constraints. This solution becomes infeasible.
- (xi) *Lines (19)–(25)*. To attempt to recover the feasibility of the solution, a step of relocation of events takes place. This event relocation step makes repeated attempts to reschedule events that do not meet hard constraints. Lines (22) and (23) select random time slots and then

study the feasibility of the exchange. If the time slots are occupied with events already scheduled which meet hard constraints, the situation is analyzed to determine whether hard constraints will be met with the exchange. If the slots selected for relocation are empty, the situation is analyzed to determine whether the events can be moved to the empty slots and satisfy the hard restrictions. The relocation of events takes place only with nontenured teacher events.

- (xii) *Line (26)*. Each time the cycle to program unTE ends, it includes a step for reprogramming events, which consists of exchanges of events scheduled in the solution provided they continue to satisfy the hard constraints. It ensures that the events of tenured teachers are not rescheduled.
- (xiii) *Lines (29) through (32)*. Once the program cycle for the nontenured teacher event schedule is finished, the algorithm is printed. It is printed if the solution is feasible or infeasible, along with the time it took the algorithm to obtain the solution.

To validate the satisfaction of the hard constraints RD of the model (see Section 3) in the CAA, it is necessary to observe each student (RD = 7) and teacher (RD = 8) to ensure that the restrictions are satisfied. This means that there should not be overlap. To ensure that there is no overlap or other invalidation of the hard constraints, an iterative search per time slot in all the classrooms is required. This is the most time consuming part of the algorithm execution. To search, a combination of two algorithms, binary search and search BTREE are applied, making it a more efficient search for UCTP [24].

Competing Interests

The authors declare that there are no competing interests regarding the publication of this paper.

References

- [1] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Dover, New York, NY, USA, 1998.
- [2] S. A. Cook, "The complexity of theorem-proving procedures," in *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing (STOC '71)*, pp. 151–158, Association for Computing Machinery, 1971.
- [3] M. G. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman & Co., New York, NY, USA, 1990.
- [4] S. Even, A. Itai, and A. Shamir, "On the complexity of timetable and multicommodity flow problems," *SIAM Journal on Computing*, vol. 5, no. 4, pp. 691–703, 1976.
- [5] O. Ben-Ayed and S. Hamzaoui, "Multiobjective multiproduct parcel distribution timetabling: a real-world application," *International Transactions in Operational Research*, vol. 19, no. 4, pp. 613–629, 2012.
- [6] C. C. Ribeiro, "Sports scheduling: problems and applications," *International Transactions in Operational Research*, vol. 19, no. 1-2, pp. 201–226, 2012.
- [7] Y. Shen, J. Xu, and Z. Zeng, "Public transit planning and scheduling based on AVL data in China," *International Transactions in Operational Research*, 2015.
- [8] S. Kiriş, "AHP and multichoice goal programming integration for course planning," *International Transactions in Operational Research*, vol. 21, no. 5, pp. 819–833, 2014.
- [9] A. Saldaña-Crovo, C. Oliva-SanMartín, and L. Pradenas-Rojas, "Integer programming models for scheduling problem on universities," *Chilen Engineering Magazine*, vol. 15, no. 3, pp. 245–259, 2007 (Spanish).
- [10] K. Banczyk, T. Boinski, and H. Krawczyk, "Parallelization of genetic algorithms for solving university timetabling problems," in *Proceedings of the International Symposium on IEEE Xplore Conference: Parallel Computing in Electrical Engineering*, pp. 325–330, 2006.
- [11] O. Rossi-Doria, M. Sampels, M. Birattari, M. Chiarandini, M. Dorigo, and L. M. Gambardella, "A comparison of the performance of different metaheuristics on the timetabling problem," in *Practice and Theory of Automated Timetabling IV*, vol. 2740, pp. 329–351, Springer, Berlin, Germany, 2003.
- [12] T. Thepphakorn, P. Pongcharoen, and C. Hicks, "Modifying regeneration mutation and hybridising clonal selection for evolutionary algorithms based timetabling tool," *Mathematical Problems in Engineering*, vol. 2015, Article ID 841748, 16 pages, 2015.
- [13] C. Di Stefano and A. G. B. Tettamanzi, "An evolutionary algorithm for solving the school time-tabling problem," in *Applications of Evolutionary Computing*, vol. 2037 of *Lecture Notes in Computer Science*, pp. 452–462, Springer, New York, NY, USA, 2001.
- [14] E. Yu and K.-S. Sung, "A genetic algorithm for a university weekly courses timetabling problem," *International Transactions in Operational Research*, vol. 9, no. 6, pp. 703–717, 2002.
- [15] M. N. M. Kahar and G. Kendall, "The examination timetabling problem at Universiti Malaysia Pahang: comparison of a constructive heuristic with an existing software solution," *European Journal of Operational Research*, vol. 207, no. 2, pp. 557–565, 2010.
- [16] N. Basir, W. Ismail, and N. Norwawi, "A simulated annealing for Tahmidi course timetabling," *Procedia Technology*, vol. 11, pp. 437–445, 2013.
- [17] A. Gunawan, K. Ming Ng, and K. Leng Poh, "A hybridized Lagrangian relaxation and simulated annealing method for the course timetabling problem," *Computers & Operations Research*, vol. 39, no. 12, pp. 3074–3088, 2012.
- [18] S. Chacha and R. M. Allen, "Optimal solution strategy for university course timetabling problem," *International Journal of Advanced Research in Computer Science*, vol. 4, no. 1, p. 35, 2013.
- [19] D. Zhang, S. Guo, W. Zhang, and S. Yan, "A novel greedy heuristic algorithm for university course timetabling problem," in *Proceedings of the 11th World Congress on Intelligent Control and Automation (WCICA '14)*, pp. 5303–5308, IEEE, Shenyang, China, July 2014.
- [20] S. Shimazaki, K. Sakakibara, and T. Matsumoto, "Iterative optimization techniques using man-machine interaction for university timetabling problems," *SpringerPlus*, vol. 4, no. 1, article 251, 2015.
- [21] V. Pereira and H. Gomes Costa, "Linear integer model for the course timetabling problem of a faculty in Rio de Janeiro," *Advances in Operations Research*, vol. 2016, Article ID 7597062, 9 pages, 2016.

- [22] M. H. Cruz-Rosales, *The scheduling-courses problem at a university and a proposed solution [Thesis]*, CIICAp-UAEM, 2010 (Spanish).
- [23] B. Martínez-Bahena and Y. Y. Calderón-Segura, *Development of a management system for FCQeI hours, using a three-layer model client, web service and database [Engineering Thesis]*, Upemor, Jiutepec, Mexico, 2008 (Spanish).
- [24] M. A. Cruz-Chávez and A. Martínez-Oropeza, "B-Tree algorithm complexity analysis to evaluate the feasibility of its application in the university course timetabling problem," *Journal of Artificial Intelligence and Soft Computing Research*, vol. 3, no. 4, pp. 251–263, 2014.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

