

*Revista Electrónica Nova Scientia*

Case study: Simulated annealing for improving  
the educational timetable

Estudio de caso: Recocido simulado para  
mejorar el calendario educativo

**Diana Sánchez-Partida<sup>1</sup>, Enrique Gabriel Baquela<sup>2</sup>, Jaime  
Mora-Vargas<sup>1</sup> y Neale R. Smith<sup>1</sup>**

---

<sup>1</sup>School of Engineering and Sciences, ITESM, México

<sup>2</sup>Furgens Research, Argentina

---

**México - Argentina**

Diana Sánchez-Partida. E-mail: [dianasanchezpartida@yahoo.com.mx](mailto:dianasanchezpartida@yahoo.com.mx)

## Resumen

**Introducción:** En ocasiones, los problemas de Optimización Combinatoria (COP), tal como el Problema de Calendarización de horarios (CTTP), se puede resolver utilizando técnicas Investigación Operativa (IO); sin embargo, cuando el problema aumenta de tamaño, la búsqueda de una solución se vuelve más compleja. Este tipo de problema es NP-duro por lo que requiere de procedimientos como métodos metaheurísticos con el fin de resolver el problema. Este trabajo aborda un Problema real en una Institución de Educación Superior Mexicana acerca de la Calendarización de horarios basada en la Curricula (CB-CTT). Cada institución tiene sus propias reglas operacionales, por lo tanto la modelación del problema es único ya que conserva sus propias características. Primero como parte de la aportación a la solución del problema, se elaboró un Software de Mediación (MS) con el fin de organizar los datos en bruto y eliminar la restricción dura en relación con los planes de estudio. Posteriormente, el problema se dividió en cinco instancias, de acuerdo a los cursos que comparten el mismo espacio físico, los cuales fueron resueltos mediante el algoritmo tradicional de Recocido Simulado (SA). El problema fue resuelto de manera satisfactoria, obteniendo la asignación de 9620 sesiones en 174.5 horas aproximadamente, aportando una solución sin particionar el problema en dos subproblemas, impactando positivamente la reducción del tiempo de elaboración de los horarios, proporcionando un horario factible y sin errores para toda la universidad.

**Método:** El Algoritmo de Recocido Simulado, cristalización simulada o enfriamiento simulado, es un algoritmo de búsqueda metaheurística para problemas de optimización; el objetivo general de este tipo de algoritmos es encontrar una buena aproximación al valor óptimo de una función en un espacio de búsqueda grande. A este valor se lo denomina "óptimo local u óptimo global". El nombre e inspiración viene del proceso de recocido del acero y cerámicas, una técnica que consiste en calentar y luego enfriar lentamente el material para variar sus propiedades físicas. El calor causa que los átomos aumenten su energía y que puedan así desplazarse de sus posiciones iniciales (un mínimo local de energía); el enfriamiento lento les da mayores probabilidades de recristalizar en configuraciones con menor energía que la inicial (mínimo global). El método fue descrito independientemente por Scott Kirkpatrick, C. Daniel Gelatt y Mario P. Vecchi en 1983.

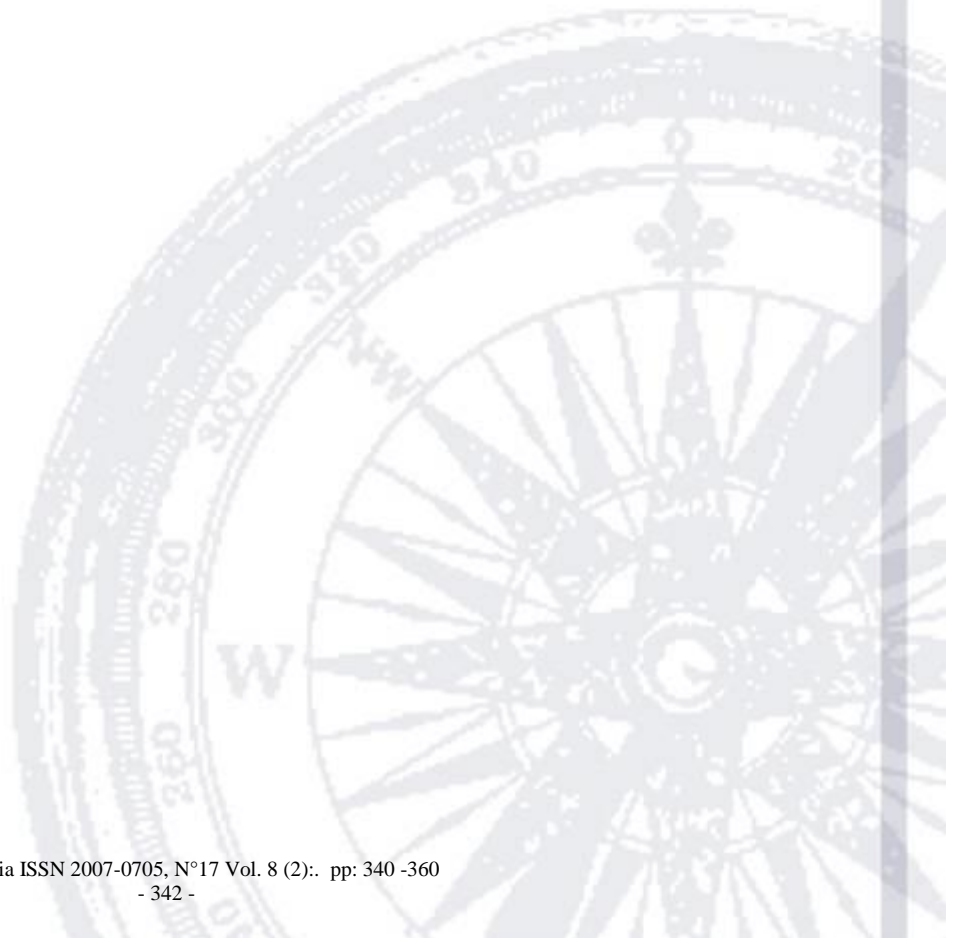
**Resultados:** Como conclusión este trabajo muestra la solución a un problema real de gran tamaño CB-CTT con 2507 cursos conformados por 1, 2, 3, 4, 5, 6, 7 ó 8 sesiones u horas por curso, totalizando 9620 sesiones; se logró asignar en 316 habitaciones con diferentes capacidades, y también satisfacer cada una de las solicitudes de 2178 profesores y 1668 grupos.

**Discusión o Conclusión:** En esta investigación se han desarrollado diversas instancias, donde 3 de los 5 son considerados por la comunidad científica como instancias grandes, que se resuelve con el algoritmo SA, proporcionando todas las soluciones factibles. De esta manera se confirma que los métodos metaheurísticos son relativamente buenos dependiendo de las circunstancias y para esta institución puede ofrecer una buena solución.

**Palabras Clave:** Problema de horarios; optimización combinatoria; metaheurística recocido simulado; software de mediación

*Recepción: 07-07-2016*

*Aceptación: 25-10-2016*



## Abstract

**Introduction:** On occasions, Combinatorial Optimization Problems (COP), like the University Course Timetabling Problem (CTTP), can be solved using Operational Research (OR) techniques; however, when the problem increases in size, finding a solution becomes more complex. This type of problem is NP-hard, requiring procedures like metaheuristic methods in order to solve the problem. This paper confronts a real world situation in Mexico concerning the Curriculum-Based Timetabling Problem (CB-CTT). Each institution have their own operational rules due to the modeling of the problema is unique because preserve their own characteristics. First, as part of the contribution to the solution of the problem, it was implemented a Mediation Software (MS) in order to organize the raw data and eliminate a hard constraint related to curricula. Subsequently, the problem handle here was split into five instances in accordance to the courses that share the same physical space, which was solved using a typical Simulated Annealing (SA) metaheuristic. In addition, The problem was satisfactorily solved, assigning 9620 lectures in 174.5 hours approximatly, providing a solution without partitioning the problem into two subproblems, impacting positivly reducing the labor time, and providing a feasible and without errors educational timetable to the whole university.

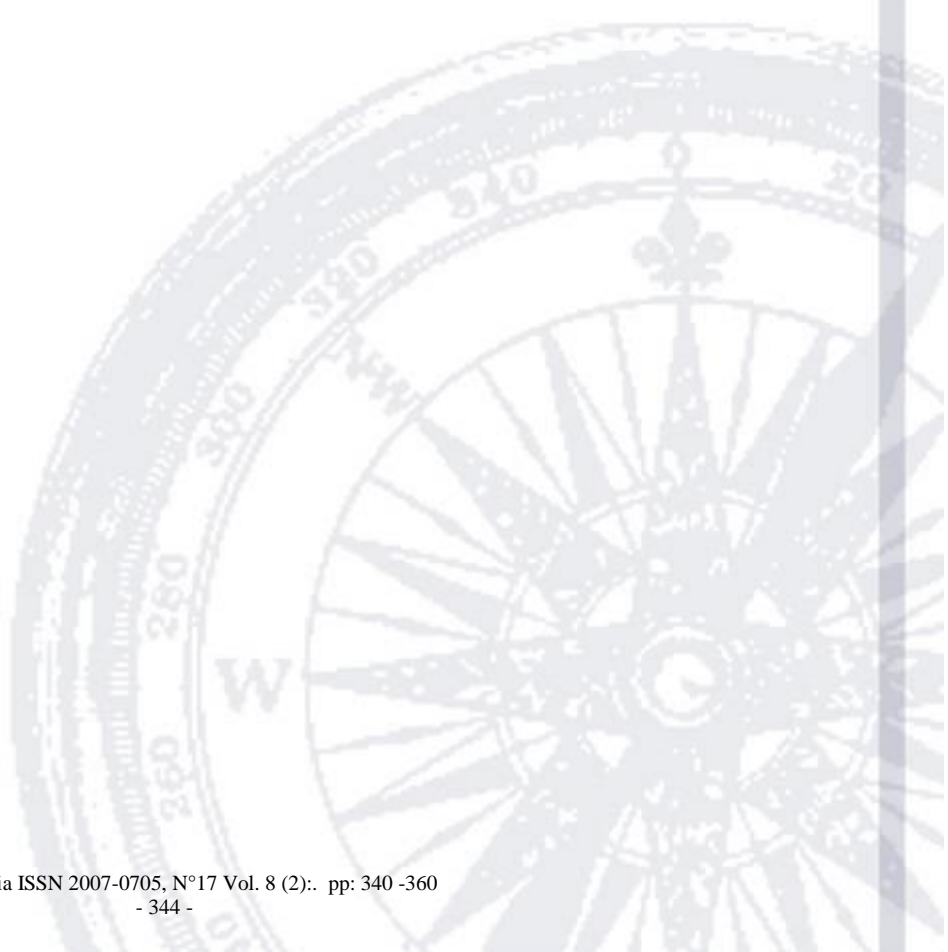
**Method:** The Simulated Annealing (SA) algorithm is a meta-heuristic search for global optimization problems; the overall objective of such algorithms is to find a good approximation to the optimal value of a function in a large search space. This value is called "global or local optimum".The name and inspiration comes from the process of annealing the steel and ceramics, a technique that involves heating and then slowly cooling the material to vary its physical properties. The heat causes the atoms increase their energy and can thus move from their initial (a local minimum energy) positions; slow cooling gives them more likely to recrystallize in configurations with lower energy than the initial (minimum overall) .The method was independently described by Scott Kirkpatrick, C. Daniel Gelatt and Mario P. Vecchi at 1983.

**Results:** Concluding this work, is shown that is confronted a large real CB-CTT problem with 2507 courses conform by 1, 2, 3, 4, 5, 6, 7, or 8 lectures or hours, totalizing 9620 lectures; being able to assign into 316 rooms with different capacities, and also satisfying all the requested of 2178 professors and 1668 groups.

**Discussion or Conclusion:** In this research were developed various instances, where 3 of 5 are considered by the research community like large instances, and solved with SA Algorithm,

regarding all feasible solutions. Thereby the metaheuristic methods are relatively good depending on the instance, and for this institution can offer a good solution.

**Keywords:** Timetabling problem, combinatorial optimization simulated annealing metaheuristic, mediation software





## Introducción

The University Course Timetabling Problem (CTTP) is NP-complete in almost all of its variants. Thus, an exact solution is achievable only for small cases (e.g., less than 100 courses according to the dataset of Socha et al. (2002) in (Babaei et al. 2015), whereas real world situations usually involve a few hundred courses. Therefore, the following case study uses metaheuristics because it is well-known that these approaches have been extensively applied in order to calculate a reasonable solution within a reasonable time period (Lewis et al. 2008), even though frequently an optimal solution is not found.

The CTTP consists in scheduling a sequence of lectures or each course within a given number of rooms and time periods satisfying a set of hard constraints, with the quality of feasible solutions evaluated based on violations of soft constraints (Schaerf, 1999). This type of problem is a large resource allocation problem in which both time periods and rooms are determined for each course (Phillips et al. 2015).

Essentially, the problem addressed here is based on the standard formulation called the Curriculum-Based Timetabling Problem (CB-CTT). The courses in conflict are those that belong to the same predefined group of courses or curricula. A course consists of multiple lectures. However, these are only some characteristics. They may include many other distinctive features and cost components; for more details see (Schaerf, 2015; Bellio et al. 2016).

Traditionally, this problem has been considered in the Operations Research field and solved based on techniques including Graph Coloring (GC) (Burke et al. 2010; Lara-Velázquez et al. 2011), Integer/Linear Programming (IP/LP) Method (Schimmelpfeng, 2007; Rudová et al. 2011; Sánchez-Partida et al. 2014; Kassa, 2015), Constraint Satisfaction(s) Programming (CSPs) (Marte, 2007) and Benders' Partitioning (Sarin et al. 2010).

On the other hand, metaheuristic approaches have also been extensively used. Some metaheuristics are the Case-Based Reasoning Method (CBR), Genetic Algorithms (GAs), Ant Colony Optimization (ACO), Memetic Algorithm (MA), Harmonic Search Algorithm (HAS), Partial Swarm Optimization (PSO), Artificial Bee Colony Optimization (ABC) and single solution algorithms including the Tabu Search Algorithm (TS), Variable Neighborhood Search (VNS), Randomized Iterative Improvement with Composite Neighboring Algorithm (RIICN), Simulated Annealing (SA) and Great Deluge Algorithm (GD). Recently, there have been multi-criteria and multi-objective techniques: novel intelligent algorithms such as hybrid approaches,

artificial intelligence, fuzzy theory, clustering algorithm and distributed multi-agent systems approach (Babaei et al. 2015).

Until now, there are many models and methods explored and applied for the same problem because each institution has its own operational rules, and hence the characteristics and the structure are significantly different. In this paper, the Curriculum-Based Timetabling Problem (CB-CTT) is studied, which was solved using a Simulated Annealing (SA) metaheuristic.

### **Literature review**

Simulated Annealing (SA) is one of the most popular metaheuristics for solving Combinatorial Optimization Problems (COP). SA is a general probabilistic local search algorithm proposed approximately 30 years ago by Cerny and Kirkpatrick in order to solve large instances of practical difficulty (Ben-Ameur, 2004).

This representation comes from a physical process called annealing, the process for growing crystals, which can be simulated by the Metropolis Monte Carlo method (Locatelli, 2000). SA is a black box stochastic algorithm that generates a sequence of random solutions covering a global optimum. It employs a slow annealing process that accepts worse solutions more easily in the beginning stages of the search as compared to later stages. Using this feature, SA escapes from local optimum and overcomes the difficulties encountered by derivative-based numerical methods (Padamallu, et al. 2008).

This metaheuristic has been widely applied in the different educational timetabling problems. For example, Thompson et al. (1998) implemented an examination scheduling system based on this method. Moreover, Burke et al. (2012) proposed a SA with reheating as a hyper-heuristic move acceptance method for examination timetabling. Ceschia et al. (2012) made an experimental analysis of a SA approach for the Post-Enrollment Course Timetabling Problem (PE-CTT). Bellio et al. (2016) focus their attention on the formulation known as the CB-CTT, contributing a feature-based tuning of SA.

On the other hand, there have been contributions such as those from Bouffard et al. (2007) that improved the solution of a fictitious instance, assigning 100 courses to 100 periods. Alternatively, Frausto-Solís (2008) give a solution with SA to the instances used for the PATAT competition. Also, in (Tarawneh et al. 2013a, Tarawneh et al. 2013b) the technique is used and proposed adaptive neighborhood structure selection mechanisms that memorized the best improvement.

Other ways to use metaheuristics are by mixing them. For instance, in Yunfeng et al. (2013), the Genetic Algorithm is combined with the Chaos and Simulated Annealing Algorithm in order to provide the solution in 8 seconds. For further information on metaheuristics used to solve such problems, refer to (Lewis et al. 2008; Mühlenthaler et al. 2015).

The last decade has seen a number of successful simulated annealing based solutions to a variety of real world problems. However, the quality of the solution depends on the choice of parameters and the way in which the problem is modelled (Thompson et al. 1998).

With regard to CB-CTT authors that explore this type of problema with similar structure even though did not use the SA as method of solution are Sarin et al. (2010) that solved the problema en Virginia Tech considering 36 coordinations and 1680 courses. Other authors that solved a big problema were Rudová et al. (2011), they give a solution to all the Purdue University using a weighted constraint satisfaction model, solving two central and six departmental problems, individually, and as a combined problem with almost 2,500 courses. Also is posible see more publications like (Mirrazavi et al. 2003; Al-Yakoob et al. 2010; and Phillips et al. 2015) ranging from 3 to 10 coordinations.

### **Structure of the paper**

This paper explores the use of Simulated Annealing in the Curriculum-Based Timetabling Problem (CB-CTT). In particular, this proposal provides a feasible solution for a Mexican university. The remainder of the paper is organized as follows. Section 3 presents the problem description. Section 4 explains Pseudocodes and Optimization of Parameters. Section 5 will examine a numerical experience and the obtained results for the instances at question. Finally, Sections 6 and 7 develop the practical implications and conclusions.

### **Problem description**

The institution located in Mexico offers approximately 2650 courses to almost 12,000 students. The institution has six departments, which manage 71 faculties, 14 modalities, 57 academic coordinators and 2178 professors. Most of the faculties share the same physical space and in some cases, they share the same professors. Currently, the assignment process is carried out manually with the help of a spreadsheet program. The scheduling is performed in two separate



stages classifying the subproblems as a partial Curriculum-Based Timetabling Problem (CB-CTT) and a Classroom Assignment Problem.

The CB-CTT is a type of problema where a serie of courses is assigned to a period of time and at the same time to a room. In this institution, the first stage posed is a partial CB-CTT because only the coordinators chose the period of time needed for the courses of thier faculties as well as the order and sequence through the day and the week, but not chose the room. The second stage, is clasified as Classroom Assignment Problem, where the administrative staff conformed for 6 people spent almost a quartly assigning the courses to the correct room.

The issues with this process are the time spent by administrative staff to verify the feasibility of the schedule made by the coordinators and the work done by the coordinators to reschedule courses. Frequently, this occurs when two or more coordinators request their respective courses in the same time period or in the same physical space. To avoid this loss of time, workload and overlapping a comprehensive solution is proposed.

First, hard and soft constraints are considered for this research. The basic components in this study are consecutive and single hours, solve conflicts among professors, students, and courses, such as course precedence in accordance to the transcript, adequate room to teach the course, schedule available by the professor, avoid overlapping courses teaching by the same professor. Briefly, we will describe the constraints considered for this study.

### **Hard Constraints**

- H1. A professor cannot attend two courses during the same time period.
- H2. A course must be taught only in one room for each time period.
- H3. A group of students cannot attend two courses during the same time period.
- H4. The students cannot take advanced courses without having first taken the preceding courses.

### **Soft Constraints**

- S1. The capacity of the rooms should be less than or equal to the number of students pre-enrolled in the course.
- S2. The courses should be reserved in the same room if possible.
- S3. Some professors can choose certain time periods for their courses.
- S4. Some professors may request a special room for a given course due to medical conditions.

Classes may begin at 7 a.m. and may end as late as 10:00 p.m. (timeslots) Monday through Friday (days), considering 75 time periods. Usually, the maximum teaching hours for professors in a classroom are between 6 and 7 hours and the maximum lecture hours for students are 7 hours. In the institution, there are 316 rooms, where 199 are rooms with only chairs and tables and the rest are laboratories of different types.

### Eliminating a hard constraint

In order to eliminate one hard constraint related to the curricula, a Mediation Software (MS) program was made considering the raw data stored in the information system. This program not only allows us to know the characteristics and size of the problem, but also allows us to organize the information.

Reviewing the curricula records is possible to join the courses in groups avoiding that the students take advanced courses without having taken the preceding courses, making possible to eliminate the H4 constraint. Therefore, knowing information such as open courses in the quarterly, the amount of lectures for each course, courses required in consecutive or isolated lecture, lists of curricula courses, lists of courses by professor, special requests for some professors or some courses, availability and types of rooms, it is possible to create the instances.

---

#### Curricula Assignment Pseudocode

---

1. Get the file with the original information **do**
2. Read Courses suffixes, days, and consecutive hours.
3. Add to filtering lists.
4. Get the file with the classrooms.
5. Get the file with curricula maps.
6. Get the output file name.
7. **Then**
8. When user selects filters to apply and launches the process **do**
9. Read the entire file with the original information.
10. **For** each record:
11. **If** it complies with the filter **do**
12. Add to output file.
13. Add course
14. Add scholar
15. Get consecutive hours and add

```

to list 16. then
17. When all records have been read.
With each course in the output list do
18. Create groups
19. Read Curricula Map and find the
course.
20. If there is no conflict between the
current course and the other ones in the
group, the course is added.
21. If there is conflict, the course is
added to a new group.
22. Add range name to the output file.
23. Save the output file.
24. end while

```

---

The outputs are instances and are formed by all the courses offered by the different faculties that share the same physical space and professors. Then, these instances are introduced as an input in the SA program.

### Method: SA pseudocode and optimization of parameters

In order to determine, or at least approximate, the value  $f^*$  in Simulated Annealing is defined as follows:

$$f^* = \min_{x \in X} f(x), \quad (1)$$

where  $f$  is the objective function and  $X \subset R$  is the feasible region.

Therefore, the cost function  $\text{Cost}(f)$  in this case is a weighted sum of the number of conflicts or violations of the hard  $W_{hard}$  and soft  $W_{soft}$  constraints conflicts.

$$\text{Cost}(f) = \text{Conflicts}(x) * W_{hard} + \text{Conflicts}(x) * W_{soft}, \quad (2)$$

SA differs from standard iterative improvement methods by allowing “uphill” moves that spoil, rather than improve, the temporary solution. After picking some initial configuration, most iterative methods continue by choosing a new configuration at each step, evaluating it and possibly replacing the previous one with it. This action is repeated until some termination condition is satisfied (e.g., no move reduces the objective function). The procedure ends in a minimum configuration, but generally, it is a local minimum, rather than the desired global

minimum. The SA method tries to escape from these local minima by using rules that are derived from an analogy to the process in which liquids are cooled to a crystalline form, a process called annealing.

It is well known that when a liquid is cooled slowly, it reaches a totally ordered form, called crystal, which represents the minimum energy state of the system. In contrast, rapid cooling results in amorphous structures, that have higher energy, representing local minima. The difference lies in the fact that when a liquid is cooled slowly, the atoms have time to reach a thermal equilibrium in each temperature. In this state, the system obeys the Boltzmann distribution:

$$P(E) \simeq e^{-E/kT}, \quad (3)$$

Here,  $p(E)$  specifies the probability distribution of the energy values of the states  $E$ , as a function of the temperature  $T$  and the Boltzmann constant  $k$ . On the one hand, for every temperature, each energy  $E$  has nonzero probability, and thus the system can change its state to one with higher energy. On the other hand, at low temperatures, the system tends to be in states with very low energy, with the global minimum achieved at temperature zero (Davidson et al. 1996).

For defining a particular SA, it is necessary to specify the stopping criterion as maximum explored and the three functions  $A$ ,  $D$ , and  $U$  given a distribution  $D(Y_k, X_{k+1})$ , where  $A$  is the function with value  $(0, 1)$ , and  $C_{k+1} = U(Z_{k+1})$  is the set  $Z_k$  that contains the information collected by the algorithm up to iteration  $k$ , and  $U$  is a non-negative function, which respectively define the distribution of the next candidate point. The probability of accepting the candidate point as the next iteration and the cooling schedule, i.e.  $C_k$  temperature in iteration  $k$ , which is a parameter through which the acceptance of the candidate points is controlled. A common choice for the acceptance function or acceptance probability is the Metropolis function:

$$A(Y_k, X_{k+1}, C_k) = \min \left\{ 1, \exp \left\{ \frac{f(Y_k) - f(X_{k+1})}{C_k} \right\} \right\}, \quad (3)$$

Where  $f(Y_k) - f(X_{k+1})$  is the difference between the current objective function in iteration  $k$  and the previous objective function in iteration  $k$ , which always accepts descent steps, but also

accepts with a positive probability (unless  $C_k = 0$ ) ascent steps in order to avoid being trapped in a local minimum which is not a global minimum. Note how the parameter  $C_k$  controls the acceptance of ascent steps by decreasing  $C_k$  the acceptance, probability of ascent steps is decreased, requiring a random number of  $p$  so-called  $p$ -random between (0, 1) in order to compare the acceptance.

In addition, it is important to determine various parameters like the penalty method that improves the performance searching. The optimal cooling schedule ensures linearity in the expected number of sample points needed to become arbitrarily close to a global optimum. For more information, see (Locatelli, 2000).

It is important to know the starting temperature, cooling factor, Markov chain length, process termination condition, and the number of iterations needed in order to allow the SA to escape from local optimum. There are many researches about the tuning parameters, such as (Wang et al. 1999; Abramson et al. 1999; Ben-Ameur, W. 2004; Shen et al. 2007; Lee et al. 2014; Bellio et al. 2016). Below the Simulated Annealing Algorithm is shown.

---

#### Simulated Annealing Procedure Pseudocode

---

```

1. find random first solution  $S$ ;
2.  $bestSoFar \leftarrow cost(S)$ ;
3.  $counter \leftarrow 0$ ;
4. while  $phase \leq maxP$  do
5.    $phase \leftarrow 0$ ;
6.    $counter \leftarrow 0$ ;
7.   while  $counter \leq maxC$  do
8.     select a random move  $m$  from
      $neighborhood(S)$ ;
9.     let  $S'$  be the solution obtained from
      $S$  with  $m$ ;
10.    if  $cost(S') < cost(S)$  then
11.       $accept \leftarrow true$ ;
12.    else
13.       $accept \leftarrow true$  with probability  $\exp$ 
       $(-\Delta/T)$ ,
14.      false otherwise;
15.    end if
16.    if  $accept$  then
17.       $S \leftarrow S'$ 
18.      if  $cost(S') < bestSoFar$  then
19.         $counter \leftarrow 0$ ;  $phase \leftarrow 0$ ;

```



```

20.     bestSoFar ← cost (S');
21.     else
22.         counter++;
23.     end if
24. end if
25. end while
26. phase++;
27. T ← decrease the temperature T;
28. end while

```

---

### Using SA to solve the timetabling problem:

In order to use SA in our problem, we use a bilevel approach. SA defines the order of courses to schedule, and a little assignment algorithm assigns each course in the first free slot available. Therefore, the SA works with an integer multidimensional-array, in which each component represent the assignment of a Lecture to a specific day, hour and room. The possible values for each component are integer number mapped to the elements in the lecture sets. Initial solution is made without any number repeated, and the neighbor generation consist in a permutation procedure, in which the current assignment is shaken.

The optimization procedure consist in make permutations to this array, call the assignment algorithm, and evaluate the size of the assignment. This approach causes that in many cases the assignment algorithm break some constraint, so in this case the evaluation of the solution is penalized multiplying it by  $(1+m)$ . In our experiments, "m" was equal to 0.25.

### Computational experience and results

For the instances are considered nine buildings, that contain 316 rooms with their own characteristics. Quarterly, is offered about 2507 courses to undergraduate student. Each course can be taught in consecutive or isolated lecture; all the lectures considered here are 9620. The total groups formed of all the faculties were 1668, and the professors hired in this period were 2178. The instances made was five, each one divide by the shared physical spaces; the features of each one is shown in Table 1.

Table 1. Features of the five instances

<b>Instances/Features</b>	<b>Courses</b>	<b>Lectures</b>	<b>Rooms</b>	<b>Professors</b>	<b>Groups</b>
<b>Ins1</b>	1846	6960	199	1626	1228
<b>Ins2</b>	223	840	44	207	107
<b>Ins3</b>	92	330	15	80	43
<b>Ins4</b>	293	1293	23	222	252
<b>Ins5</b>	53	197	35	43	38
<b>Total</b>	2507	9620	316	2178	1668

A set of four random starts runs was developed for obtaining a solution of each instance. In the Table 2, can be seen the average of constraints violated and the elapsed time necessary for obtaining a good result.

Table 2. Average over 4 random starts

	<b>Violated Constraints (%)</b>	<b>Run Time (sec)</b>
<b>Ins1</b>	11	259200.03
<b>Ins2</b>	7	86408.84
<b>Ins3</b>	3	86587.4
<b>Ins4</b>	9	109612.42
<b>Ins5</b>	5	86400.01

Therefore, we reached a good solution for all the entire university in a short time. In the Table 3, is shown a timetable of the room 3 with capacity of 30 students. Is possible observe that the program tries to occupy the most of every room, leaving empty spaces. Here they are assigned 23 courses and in Table 4, can be seen the specifications of each course.

Table 3. An example of a timetable

Room 3 Capacity 30					
Timeslots/Days	[1,]	[2,]	[3,]	[4,]	[5,]
[,1]	39	49	16	57	83
[,2]	39	49	16	57	83
[,3]	39	49	16	57	83
[,4]	39	50	16	57	10
[,5]	28	50	54	9	10
[,6]	28	50	54	9	10
[,7]	28	24	54	9	10
[,8]	46	24	64	3	4
[,9]	46	24	64	3	4
[,10]	46	24	64	3	4
[,11]	55	24	64	3	4
[,12]	55	32	32	60	60
[,13]	55	32	32	60	90
[,14]	86	86	51	51	90
[,15]	86	86	51	51	90

One of the specifications that helped to assign such courses to salon 3 is the pre-enrolled students against the capacity of the room. In addition, may notice that the amount of lectures and the number of consecutive lectures were satisfactorily assigned; for example the course 3 must be taught in 4 lectures weekly in 4 consecutive lectures.

Table 4. Specifications of courses assigned to the room 3

Courses	Student pre-enrolled	Amount of lectures	Isolated or consecutive lectures	Group	Professor	Forbidden periods
3	2	4	4	1	2	4,5,6
4	2	4	4	1	2	4,5,6
9	4	3	3	4	7	
10	5	4	4	4	8	
16	6	4	4	8	13	
24	9	3	3	13	20	
28	11	3	3	15	24	
32	11	4	2	17	27	
39	12	4	4	19	32	
46	15	3	3	24	39	
49	16	3	3	24	41	
50	16	3	3	24	42	

51	16	4	2	26	43	
54	16	3	3	28	46	
55	16	3	3	29	47	
57	16	4	4	30	49	
60	17	3	1	31	52	
64	17	4	4	32	56	
83	19	3	3	38	75	
86	20	4	2	41	77	
90	29	3	3	43	79	

Moreover, checking the remaining constraints of the same course 3, we can demonstrate that the professor 2, which also teaches the course 4, and the prohibited periods to teach the courses are 4, 5, and 6. Analyzing the constraint of the group, we can see that both courses belong to the group 1. If we again see Table 3, the course 3, and 4 are assigned to periods 8,9,10, and 11 in the days 4 and 5 corresponding to Thursday and Friday in accordance with the constraints mentioned.

### Conclusions and future works

Concluding this work, the timetabling is one of the most important activities in Planning in Higher Education Institutions, since it is not only is an organization of resources but also is the assignment of the infrastructure and equipment necessary in order to guaranteed quality education to students (Learning and Teaching Scotland, 2006).

In this institution is confront a large real CB-CTT problem with 2507 courses conform by 9620 lectures; being able to assign into 316 rooms with different capacities, and also satisfying satisfactorily the requested of 2178 professors and 1668 groups. The contribution of this paper is twofold. First, is propose an effective MS that helps to organize the raw data, being capable of eliminating the curricula constraint. Second, a real problem was solved in one procedure eliminating the two stage previously mention; the solution was developed applying a Generalized SA method obtained a good average time the problem.

In this research were developed various instances, where 3 of 5 are considered by the research community like large instances, and solved with SA Algorithm, regarding all feasible solutions. Thereby the metaheuristic methods are relatively good depending on the instance situation, for this institution were satisfactory implemented. In this paper, the transitions are accepted in accordance with the Metropolis criterion. On the other hand definitely is necessary to adjust the parameters for each instance in order to get better approaches. The performance of SA has been

enhanced by introducing smaller iterations and frequenting drops of temperature, but is advisable determine the parameters and the stopping criterion with Design of Experiment (DOE) in order to guaranty a better solution.

Lastly, there are studies into the utilization of teaching space at real universities suggest that rooms are occupied up to 60 - 80 % of the time on average, which is typical for the CTPP problems. In this study, we achieve to have utilization up to 70 - 100%; this is normal due to the type of operational rules that regulate the institution.

### Acknowledgements

This paper was a product of postdoctoral study financially supported by Consejo Nacional de Ciencia y Tecnología – CONACyT (CV: 371231 register: 250076). The author would like to express gratitude to the group of internal and external reviewers.

### References

- Abramson, D., Mohan, K. A. & Dang, H. (1999). Simulated annealing cooling schedules for the school timetabling problema. *Asia - Pacific Journal of Operational Research*, 16(1), 1-22.
- Al-Yakoob, Salem M.; Sherali, Hanif, D.; Al-Jazzaf, M. (2010). A mixed-integer mathematical modeling approach to exam timetabling. *Computational Management Science*, 7, 19–46.
- Babaei, H., Karimpour, J., & Hadidi, A. (2015). A survey of approaches for university course timetabling problem. *Computers & Industrial Engineering*, 86, 43. doi:10.1016/j.cie.2014.11.010
- Ben-Ameur, W. (2004). Computing the initial temperature of simulated annealing. *Computational Optimization and Applications*, 29(3), 369-385. doi:10.1023/B:COAP.0000044187.23143.bd
- Bellio, R., Ceschia, S., Gaspero, L. D., Schaerf, A., & Urli, T. (2016). Feature-based tuning of simulated annealing applied to the curriculum-based course timetabling problem. *Computers & Operations Research*, 65, 83-92. doi:10.1016/j.cor.2015.07.002
- Bouffard, V., & Ferland, J. A. (2007). Improving simulated annealing with variable neighborhood search to solve the resource-constrained scheduling problem. *Journal of Scheduling*, 10(6), 375-386. doi:10.1007/s10951-007-0043-7



Burke, E. K., Mareček, J., Parkes, A. J., & Rudová, H. (2010). A supernodal formulation of vertex colouring with applications in course timetabling. *Annals of Operations Research*, 179(1), 105-130. doi:10.1007/s10479-010-0716-z

Burke, E. K., Kendall, G., Mısıır, M., & Özcan, E. (2012; 2010;). Monte carlo hyper-heuristics for examination timetabling. *Annals of Operations Research*, 196(1), 73-90. doi:10.1007/s10479-010-0782-2

Ceschia, S., Gaspero, L. D., & Schaerf, A. (2012). Design, engineering, and experimental analysis of a simulated annealing approach to the post-enrolment course timetabling problem. *Computers & Operations Research*, 39(7), 1615-1624. doi:10.1016/j.cor.2011.09.014

Davidson, R.;HAREL, D. 1996). Drawing Graphs Nicely Using Simulated Annealing. *ACM Transactions on Graphics*, 15(4), 301–331.

Frausto-Solís, J., Alonso-Pecina, F., & Mora-Vargas, J. (2008). An Efficient Simulated Annealing Algorithm for Feasible Solutions of Course Timetabling. In Gelbukh A., & Morales, E.F (Eds.): *MICAI 2008, LNAI 5317*, pp. 675–685. Springer-Verlag, Berlin Heidelberg. doi: 10.1007/978-3-540-88636-5\_64

Kassa, B.A. (2015). Implementing a Class-Scheduling System at the College of Business and Economics of Bahir Dar University, Ethiopia. *Interfaces* 45(3), 203-215. <http://dx.doi.org/10.1287/inte.2014.0789>

Lara-Velázquez, P., López-Bracho, R., Ramírez-Rodríguez, J. & Yáñez, J. (2011). A model for timetabling problems with period spread constraints. *Journal of the Operational Research Society*, 62, 217 -222.

Lee, C., & Lee, D. (2014). Determination of initial temperature in fast simulated annealing. *Computational Optimization and Applications*, 58(2), 503-522. doi:10.1007/s10589-013-9631-y

Lewis, R. (2008; 2007). A survey of metaheuristic-based techniques for university timetabling problems. *OR Spectrum*, 30(1), 167-190. doi:10.1007/s00291-007-0097-0

Learning and Teaching Scotland (2006). *Focusing on Inclusion and the Education (Additional Support for Learning)*. Scottish executive. 1-40. ISBN-13: 978-184399-133-5; ISBN-10: 1-84399-133-0

Locatelli, M. (2000). Convergence of a simulated annealing algorithm for continuous global optimization. *Journal of Global Optimization*, 18(3), 219-233. doi:10.1023/A:1008339019740

Marte, M. (2007). Towards constraint-based school timetabling. *Annals of Operations Research*, 155(1), 207-225. doi:10.1007/s10479-007-0218-9

Mirrazavi, SK.; Mardle, SJ.; Tamiz, M.; (2003). A two-phase multiple objective approach to university timetabling utilising optimisation and evolutionary solution methodologies. *Journal of the Operational Research Society*, 54, 1155–1166.

Mühlenthaler, M., & Wanka, R. (2015). Fairness in Academic Course Timetabling. *Lecture Notes in Economics and Mathematical Systems* 678, Switzerland: Springer. doi: 10.1007/978-3-319-12799-6\_4

Pedamallu, C.S. & Özdamar, L. (2008). Comparison of Simulated Annealing, Interval Partitioning and Hybrid Algorithms in Constrained Global Optimization, In Siarry, P., & Michalewicz, Z. *Advances in metaheuristics for hard optimization (Natural computing series)*, 1-22. New York: Springer.

Phillips, A. E., Waterer, H., Ehrgott, M., & Ryan, D. M. (2015). Integer programming methods for large-scale practical classroom assignment problems. *Computers & Operations Research*, 53, 42-53. doi:10.1016/j.cor.2014.07.012

Rudová, H. Müller, T. & Murray, K. (2011). Complex university course timetabling. *Journal of Scheduling*, 14, 187–207. doi 10.1007/s10951-010-0171-3

Sánchez-Partida, D., Martínez-Flores, J.L. & Olivares-Benítez, E. (2014). An integer linear programming model for a university timetabling problem considering time windows and consecutive periods. *Journal of Applied Operational Research*, 6(3), 158–173.

Sarin, S. C., YuqiangWang & Varadarajan, A. (2010). A university-timetabling problem and its solution using Benders' partitioning a case study, *Journal of Schedule*, 13, 131–141. doi: 10.1007/s10951-009-0157-1

Schaerf, A. (1999). A survey of automated timetabling. *Artificial Intelligence Review*, 13(2), 87-127. doi:10.1023/A:1006576209967

Schaerf, A. (2015). Comments on: An overview of curriculum-based course timetabling. *TOP*, 23(2), 362-365. doi: 10.1007/s11750-015-0365-0

Schimmelpfeng, K., & Helber, S. (2007). Application of a real-world university-course timetabling model solved by integer programming. *OR Spectrum*, 29(4), 783-803. doi:10.1007/s00291-006-0074-z

Shen, Y., Kiatsupaibul, S., Zabinsky, Z. B., & Smith, R. L. (2007). An analytically derived cooling schedule for simulated annealing. *Journal of Global Optimization*, 38(3), 333-365. doi:10.1007/s10898-006-9068-2

Tarawneh, H. Y., Ayob, M., & Ahmad, Z. (2013a). A hybrid simulated annealing with solutions memory for curriculum-based course timetabling problem. *Journal of Applied Sciences*, 13(2), 262-269.

Tarawneh, H. Y., & Ayob, M. (2013b). Adaptive neighbourhoods structure selection mechanism in simulated annealing for solving university course timetabling problems. *Journal of Applied Sciences*, 13(7), 1087-1093.

Thompson, J. M., & Dowsland, K. A. (1998). A robust simulated annealing based examination timetabling system. *Computers and Operations Research*, 25(7-8), 637-648. doi:10.1016/S0305-0548(97)00101-9

Wang, T., & Wu, K. (1999). A parameter set design procedure for the simulated annealing algorithm under the computational time constraint. *Computers and Operations Research*, 26(7), 665-678. doi:10.1016/S0305-0548(98)00079-3

Yunfeng, D. (2013). Research on time table problem based on improved genetic algorithm combined chaos and simulated annealing algorithm. *Journal of Applied Sciences*, 13(15), 2947-2952. doi:10.3923/jas.2013.2947.2952

