

Practical relaxation of a special case of the Curriculum-Based Course Timetabling problem

Nelson Rangel-Valdez · Jorge Omar Jasso-Luna ·
Mario Humberto Rodriguez-Chavez ·
Gustavo Bujano-Guzman

Received: 10 December 2012 / Accepted: 10 January 2014 / Published online: 12 April 2014
© Springer-Verlag Berlin Heidelberg 2014

Abstract The Curriculum-Based Course Timetabling (CB-CTT) problem involves the task of scheduling lectures of courses to classrooms, considering teacher availability, a specified curricula, and a set of constraints. In the Polytechnic University of Victoria (PUV), Mexico, the CB-CTT problem occurs with the additional constraints that the courses must be assigned to teachers and that a teacher cannot teach more than one course in the same curriculum. This document proposes a methodology to solve this special case CB-CTT problem. The main contributions derived from the research are: (a) the solution of the special case of the CB-CTT problem; (b) the use of a Simulated Annealing based methodology, which relaxes the problem into three simpler subproblems (the assignment, distribution, and scheduling of courses), to reduce the search space and speed up the construction of a solution; and (c) the solution of a real world instance using the proposed methodology. The results show that the approach can construct solutions for the problem in the PUV in a few minutes, and of a similar quality to those manually constructed in a couple of weeks.

Keywords Curriculum-Based Course Timetabling · Simulated Annealing · Meta-heuristics

1 Introduction

The Curriculum-Based Course Timetabling (CB-CTT) is a problem that appears at the beginning of each term in many universities. To solve this problem, different constraints must be taken into account, and they can vary depending on the considered particular case. Mainly, the constraints are associated with the availability of classrooms, or teachers, or with the number of classes to be assigned, etc. The general case of this problem is NP-complete [5], which means that trying to find the optimal solution involves the consumption of great amount of computational resources. Despite of its complexity, it has been tackled with many approximated strategies over the years [8].

At the Polytechnic University of Victoria (PUV), the CB-CTT problem involves a complex combination of hard and soft constraints that occur at the beginning of each term. Currently, the problem is solved manually by the program directors, requiring several weeks to get an initial solution based on existing data. This solution may be adjusted using common sense and knowledge of the situation.

This work presents a methodology based on the use of a meta-heuristic for the solution of the special case of the CB-CTT problem at the PUV, which includes the constraint that no teacher can teach more than one course per curriculum. In addition, an analysis of different implementations of the meta-heuristics is presented based on the use of different evaluation functions.

The main contributions derived from the research are: (a) the formal definition of a special case of the CB-CTT problem that, to the best of our knowledge, do not completely match its

N. Rangel-Valdez (✉) · J. O. Jasso-Luna · M. H. Rodriguez-Chavez ·
G. Bujano-Guzman
Departamento de Tecnologías de la Información, Parque Científico y
Tecnológico de Tamaulipas, Universidad Politécnica de Victoria,
Carretera Victoria-Soto la Marina Km. 5.5, 87138 Cd. Victoria,
Tamaulipas, Mexico
e-mail: nrangelv@upv.edu.mx

J. O. Jasso-Luna
e-mail: ojassol@upv.edu.mx

M. H. Rodriguez-Chavez
e-mail: mrodriguez@upv.edu.mx

G. Bujano-Guzman
e-mail: 1030112@upv.edu.mx

more general definition [4]; and (b) a novel approach to solve the problem, which separates the problem in three subproblems (the assignment of courses to teachers, the distribution of the courses over the days of a week, and the daily schedule of the courses), such that this relaxation makes it easier to solve. It is important to point out that the approach reduces the time to construct the schedules from 2 weeks to a couple of minutes.

The remaining of this article is organized in the following way: Sect. 2 formally defines the problem under study in this article. Section 3 presents the work on scheduling problem solution related to the research presented. Section 4 shows the methodology proposed, which includes the model for the solution, and the definition of its elements based on a Simulated Annealing (SA) approach. Section 5 shows the results of the experiments performed over an instance taken from the PUV. Finally, Sect. 6 presents the conclusions derived from the research.

2 University Timetabling problem

Anthony Wren [18] defines the scheduling problem (SP) as *the arrangement of objects into a pattern in time or space in such a way that some goals are achieved, or nearly achieved, and that constraints on the way objects may be arranged are satisfied or nearly satisfied*. This problem is common, and it is related with the timetables that are daily performed in human life.

One important task that involves the SP is the construction of timetables to organize teachers, lectures of courses, and classrooms in a University. For example, in the particular case of the PUV, it is required that the lectures of each course must have assigned a teacher and a classroom, according to a specific set of constraints. The construction of a timetable for this activity takes over 15 days; and it must be done at least three times a year at the beginning of each term.

This section specifies the problem at the PUV, and presents a formal definition for it.

2.1 Problem description

The timetabling problem that is found at the PUV represents a special case of the CB-CTT [7]. Taking into account that the timetable for the problem at PUV is scheduled in a weekly basis, the following basic elements are defined:

Days and Timeslots Each week has 5 days. Each day is split in a fixed number of non overlapping timeslots, which are equal to all the days.

Courses and Teachers Each course consists of a fixed number of lectures (also denoted as classes) to be scheduled during the week. The course is taught by a teacher. Each

teacher specifies an availability chart, i.e., a daily set of the timeslots in which a lecture must be assigned to him/her.

Rooms or Classrooms There is a fixed number of classrooms in which the lectures must be scheduled.

Curricula or Groups A *curriculum* is a group of courses such that any pair of them cannot be schedule at the same time, because they share common students.

Idle Timeslots One unassigned timeslot between two assigned timeslots in a curriculum daily schedule, or a teacher availability chart, is considered an idle timeslot (or IT).

A feasible timetable for the PUV is one in which all lectures have been scheduled at a timeslot and a classroom, so that the hard constraints $\{H_1, \dots, H_4\}$ are satisfied. In addition, a feasible timetable satisfying the four hard constraints incurs a penalty cost for the violations of any of the four soft constraints $\{S_1, \dots, S_4\}$. Then, the objective of the CB-CTT problem is to minimize the number of soft constraint violations in a feasible solution. The four hard constraints (identified as H_i) and four soft constraints (identified as S_i) are:

- H_1 . *Lectures* All lectures of a course must be assigned to a particular timeslot in a classroom.
- H_2 . *Conflicts* Lectures of courses in the same curriculum, or taught by the same teacher, cannot be scheduled in the same timeslot. Moreover, courses in the same curriculum cannot be scheduled with the same teacher.
- H_3 . *Availability* If the teacher of a course is not available at a given timeslot, then no lectures of a course can be assigned to that timeslot.
- H_4 . *Covering* Each course must have a teacher assigned to it, according with its area of expertise.
- S_1 . *Minimum Curriculum IT* Given that the curriculum is associated with a group of students, this constraint means that the agenda of students should be as compact as possible.
- S_2 . *Minimum Teacher IT* It is preferred that the agenda of a teacher should be as compact as possible.
- S_3 . *Lectures per Day* A course cannot have more than two lectures per day; these lectures can be in non consecutive hours.
- S_4 . *Courses per Teacher* There is a previously specified maximum number of courses that a teacher can teach.

According with the information presented, the specialized case of the CB-CTT problem identified in the PUV can be seen as the solution of the following particular problems: (a) the assignment of courses to teachers; (b) the scheduling of the lectures during a week; (c) the scheduling of teacher timetables; and (d) the assignment of classrooms. The main difference with other CB-CTT formulations relies on the relaxation of the constraints, while no classroom capacity is considered in this problem, it considers the assignment

Table 1 Different sets required for the formal definition of the PUV-CBCTT

Symbol	Meaning
$CR = \{cr_1, cr_2, \dots, cr_s\}$	A set where each element $cr_i, 1 \leq i \leq s$, is a curriculum, i.e., a group of courses with students in common
$TC = \{tc_1, tc_2, \dots, tc_m\}$	A set where each element $tc_i, 1 \leq i \leq m$, contains the group of courses that a teacher might taught
$TA = \{ta_1, ta_2, \dots, ta_m\}$	A set where each element $ta_i, 1 \leq i \leq m$, contains the timeslots in which a teacher is available to teach a lecture
$CL = \{cl_1, cl_2, \dots, cl_n\}$	A set where each element $cl_i, 1 \leq i \leq n$, contains the lectures that must be taught by each course during the week

of courses to teachers. In addition, constraints involving the minimum spread time for the courses during the week, or constraints in the use of classrooms, are not necessary to be solved.

To avoid any confusion, the following section presents the mathematical formulation for this problem, which will be named from now on as PUV-CBCTT problem.

2.2 Problem formulation of PUV-CBCTT

The PUV-CBCTT problem consists of the sets C, T, P, R . The set $C = \{c_1, c_2, \dots, c_n\}$ corresponds to the courses. The set $P = \{p_{1,1}, p_{1,2}, \dots, p_{1,p}, p_{2,1}, p_{2,2}, \dots, p_{2,p}, \dots, p_{d,1}, p_{d,2}, \dots, p_{d,p}\}$ is the $d \cdot p$ timeslots, or slots of time, distributed in a week formed by d days. The set $T = \{t_1, t_2, \dots, t_m\}$ represents the teachers. And, the set $R = \{r_1, r_2, \dots, r_w\}$ is formed by the available classrooms. Additional information concerning these sets is presented in Table 1.

Having in mind the previous data, the PUV-CBCTT problem can be defined as the problem of finding a timetabling array \mathcal{M} of size $w \times d \times p$, where each cell $m_{i,j,k}$, for $1 \leq i \leq w$ and $1 \leq j \leq d$ and $1 \leq k \leq p$ contains a tuple $(cl_{u,l}, t_v) \in T \times C$, subject to:

- $\bigcup_{i,j,k} \mathcal{C}(m_{i,j,k}) = C$;
- $\mathcal{CR}(m_{i_1,j,k}) \cap \mathcal{CR}(m_{i_2,j,k}) = \emptyset$, for any $i_1 \neq i_2$ having fixed j, k ;
- $\mathcal{T}(m_{i_1,j,k}) \cap \mathcal{T}(m_{i_2,j,k}) = \emptyset$, for any $i_1 \neq i_2$ having fixed j, k ;
- $\mathcal{T}(m_{i_1,j,k}) \cap \mathcal{T}(m_{i_2,j,k}) = \emptyset$, for any $i_1, j_1, k_1, i_2, j_2, k_2$, where $\mathcal{CR}(m_{i_1,j_1,k_1}) = \mathcal{CR}(m_{i_2,j_2,k_2})$;
- $\mathcal{C}(m_{i,j,k}) \in tc_{\mathcal{T}(m_{i,j,k})}$, for any i, j, k ;
- $p_{j,k} \in ta_{\mathcal{T}(m_{i,j,k})}$; and
- $Idle(CR) + Idle(T) + \mathcal{R}(\mathcal{M})$ is minimized.

where $cl_{u,l}$ stands for the lecture l of the course u ; $\mathcal{C} : m_{i,j,k} \rightarrow c_v$ is a function that obtains the course v assigned to timeslot k of day j in classroom i ; $\mathcal{CR} : m_{i,j,k} \rightarrow cr_v$ is a function that obtains the curriculum v associated with the lecture l of the course u assigned to the classroom i , in the day j , and the timeslot k ; $\mathcal{T} : m_{i,j,k} \rightarrow t_v$ is a function that obtains the teacher v associated to that lecture; $Idle$ is a

function that determines the number of timeslots without any lecture assigned per curriculum or teacher; and $\mathcal{R}\mathcal{M} \rightarrow v$ is a function that returns the number v of unused classrooms.

For the purpose of this research, the strategy that is developed searches the hierarchically minimization of the three elements $Idle(R), Idle(T), \mathcal{R}(\mathcal{M})$. The main reason for such objective function is that it allows the study of the individual importance of each element in the solution of the PUV-CBCTT problem.

3 Related work

Several authors, among them, Schaerf [13] and Werra [16], believe that the Timetabling problem (TTP) cannot be completely automated. The reasons they give are two: a timetable is not easily shown in an automated system, and on the other hand, as the search space is enormous, human intervention may be useful to guide the search to directions that the system alone would not easily go. Due to these reasons, most systems allow human intervention to adjust the final solution, and are called interactive timetabling.

Werra [16] formally explains several TTP problems and presents their respective formulations. He also describes the most important research in which graph theory is applied. For the purpose of this research, the remaining of this section presents some of the most representative works of the CBCTT problem that has been developed in the recent years.

Lu and Hao [7] show an approach based on Tabu Search that solves the more general case of the CBCTT. In this problem, a set of lectures must be assigned into a weekly timetable. The hard constraints considered are related with those taken into account in this research, with the only exception of the covering constraint, i.e., the problem studied does not involve the assignment of courses to teachers.

Almilli [2] reports the use of a hybrid strategy for the solution of the Educational Timetabling problem. The approach combines SA and Genetic algorithms to solve a problem where the entities considered were courses, classrooms, students, and timeslots. This problem differs from the case of the CBCTT in which several constraints are not considered, e.g., the inner problem does not consider the assignment of teachers to courses and classrooms, or the availability of teachers.

Pothitos et al. [9] describe the course timetabling problem in a similar way than Lu and Hao do in [7]. The strategy followed to solve such problem consists in mapping it to the domain of a Constraint Satisfaction Problem (CSP) instance. There, a CSP solver is used to build a solution. The considered problem in that work also lacks the assignment of courses to teachers in its definition.

Abdullah et al. [1] present the solution of a similar CB-CTT problem like the one presented here. One of the slight differences is that it does not include the assignment of courses to teachers. The algorithm used was a combination of the Great-Deluge strategy and an electromagnetism-like mechanism.

Finally, the more recent approach presented by Rues-Maw and Hsiao [12] describes the use of a Particle Swarm Optimization approach for the solution of another variation of the course timetabling problem. This case of the CB-CTT includes tighter constraints, not considered for the special case at the PUV. In addition, the constraint involving teachers and the curricula is not taken into account in the work of Rues-Maw and Hsiao.

In summary, all the revised articles related with this research do not involve the constraint where a teacher cannot teach more than one course in the same curriculum. Because of that, the present document bases its study on the solution of that special case of the CB-CTT problem, which includes the commented constraint.

The next section shows the methodology followed to solve the special case of the CB-CTT problem at the PUV.

4 Methodology

The solution proposed to solve the PUV-CBCTT is outlined in Fig. 1. Basically, the problem is relaxed in the following three subproblems: (a) the assignment of courses to teachers; (b) the distribution of the lectures during the week; and (c) the scheduling of classrooms to courses, according with the availability of the teachers. These subproblems are detailed in the remaining of this section.

4.1 Solving the assignment of courses to teachers

The subproblem of assigning courses to teachers can be defined as follows. Given the set of courses \mathcal{C} , the set of teachers \mathcal{T} , and the set \mathcal{CT} of tuples (c, t) (which denotes that a course c can be taught by a teacher t); is it possible to match courses and teachers such that the constraints of the studied CB-CTT problem holds? The complexity of the search space for the most general case of this problem is $O(|\mathcal{T}|^{|\mathcal{C}|})$.

The approximated solution of this first subproblem is based on the solution of an instance of the maximum match-

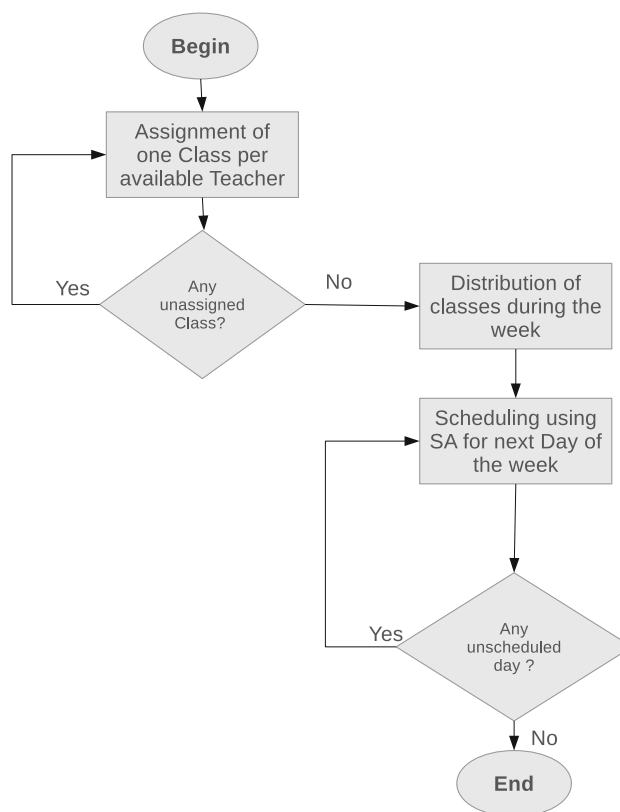


Fig. 1 Methodology of solution for the scheduling problem in the Mexican university

ing problem (MMP) [11], as is depicted in the Table 2a. Firstly, through lines 1–9, a bipartite graph B is built; it represents as nodes the courses and teachers, and as edges the possibility that a teacher teaches a specific course. After that, from lines 11–15, the MMP is solved over the graph instance B , and the matching S is returned. The edges in S are the matches produced by the solution of the MMP strategy; these edges are the assignments, and are removed from E . Then, the invalid edges, i.e., those involving teachers that cannot teach any more courses, are also removed and a new iteration is performed, to assign other missing courses. This process is iterated until every course has a teacher assigned to it, i.e., when $|E| = 0$. Table 2b shows a graphical example of the solution of this subproblem. The run-time complexity for the worst case scenario of this algorithm is $O(\max_courses \cdot (|\mathcal{C}| + |\mathcal{T}|)^3)$, where $\max_courses$ is the maximum number of course assigned to a teacher.

4.2 Distribution of the classes during the week

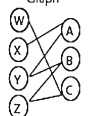
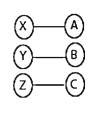
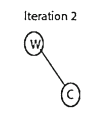
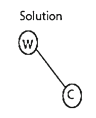
The second subproblem to be solved in the methodology proposed is the distribution of lectures during the week based on the constraints that define the studied CB-CTT problem. For this purpose, the MMP problem is also used. Firstly, lets consider the sets \mathcal{C} , and \mathcal{T} described in the previous

Table 2 Example of the solution of the assignment of courses to teachers

(a) Algorithm to assign courses to teachers

1. Build graph $B = (V_1, V_2, E)$
2. for each $c \in \mathcal{C}$
3. $V_1 \leftarrow V_1 \cup c$
4. for each $t \in \mathcal{T}$
5. $V_2 \leftarrow V_2 \cup t$
6. for each $c \in \mathcal{C}$
7. for each $t \in \mathcal{T}$
8. if $(c, t) \in \mathcal{CT}$ then
9. $E \leftarrow E \cup (c, t)$
10. repeat
11. $S \leftarrow MMP(B)$
12. for each $(c, t) \in S$
13. assign course c to teacher t
14. $E \leftarrow E - (c, t)$
15. $E \leftarrow E - \text{InvalidEdges}$
16. until $|E| = 0$

(b) Graphical example of assignment of courses to teachers.

Input			Iteration 1	
Teachers	Classes	Max.Classes	Bipartite Graph	Solution
A	XY	2		
B	YZ	2		
C	WZ	2		
Output			Iteration 2	Solution
Teachers	Classes			
A	X			
B	Y			
C	ZW			

subproblem, the set \mathcal{CL} where each element cl_i represents the lectures of the course i , and the set $W = \{d_1, d_2, \dots, d_n\}$ containing the days of the week. Also lets consider that there are $|R|$ available classrooms, each having p timeslots to be assigned. This subproblem is related to the three-dimensional matching problem [5], and the complexity of its search space is $O(2^{H \cdot |R| \cdot |W|})$, where $H = \sum_{vi \in \mathcal{CL}} |cl_i|$.

The algorithm used to distribute the lectures follows a greedy approach. Firstly, it tries to fill the first day of the week with as much lectures as it is possible. Then, it proceeds in this way in the next days. The algorithm shown in Table 3 summarizes this process. The process `InvalidEdges` assures that each course has at most two lectures assigned per day. The run-time complexity of this algorithm is $O(|W| \cdot (H + |R|)^3)$, because for each considered day, a bipartite matching problem must be solved.

pg

Note that in this step is ignored the availability chart, i.e., the set of timeslots in which each professor is available; therefore, this task is performed in the following phase, the schedule construction step, using the SA approach.

4.3 Solving the scheduling problem per day of class

The last and final step of the methodology of solution involves the final construction of the schedule, which programs classes to classrooms and available timeslots. The constraints to be

Table 3 Algorithm to distribute lectures during the week

1. Build graph $B = (V_1, V_2, E)$
2. for each $c \in \mathcal{C}$
3. for $h = 1$ to $h = |cl_c|$
4. $V_1 = V_1 \cup cl_{ch}$
5. for each $r \in \mathcal{R}$
6. for $t = 1$ to p
7. $V_2 = V_2 \cup rt$
8. for each $v \in V_1$
9. for each $w \in V_2$
10. $E \leftarrow E \cup (v, w)$
11. for each $d \in \mathcal{W}$
12. $S \leftarrow MMP(B)$
13. for each $(v, w) \in S$
14. assign lecture v to day d in room w
15. $E \leftarrow E - (v, w)$
16. $E \leftarrow E - \text{InvalidEdges}$

considered to this point are: (a) a teacher can not teach two classes at the same time; (b) a classroom can not be assigned to two classes at the same time; (c) a group can not have two classes at the same time; and (d) the teachers must be scheduled in the timeslots preferred by them.

To solve this last subproblem, the SA meta heuristic was used [6]. This algorithm requires the definition of the follow-

Table 4 Example of the matrix \mathcal{M} representing a schedule of the CB-CTT problem studied in this paper

(a) Tuples	(b) An schedule in \mathcal{M}	
(A,X,G ₁)		<i>timeslot</i> ₁ <i>timeslot</i> ₂
(B,Y,G ₂)	<i>classroom</i> ₁	(A,X,G ₁) (C,Z,G ₁)
(C,Z,G ₁)	<i>classroom</i> ₂	(C,W,G ₂) (B,Y,G ₂)
(C,W,G ₂)		

ing elements: (a) a model of the solution; (b) an initial solution; (c) a neighborhood function; (d) an evaluation function; and (e) the algorithm parameters initial temperature T_i , final temperature T_f , cooling factor α , and length of the Markov chain L . A detailed description of these elements is presented in the remaining of this section.

4.3.1 Solution representation

To implement the SA, the solution for the CB-CTT problem was represented as a two-dimensional matrix \mathcal{M} , where its rows indicate the classrooms, its columns the different timeslots to be considered, and the values of each cell will contain a tuple $\mathcal{T} = (T, G, C)$, representing the teacher, group, and class to be taught there. An example of this representation can be seen in Table 4.

In the remaining of the document, the values to be scheduled will be called *tuples*, and denoted by $\mathcal{T} = (T, G, C)$, and represented by the three elements: teacher, group, class.

4.3.2 Initial solution

The construction of an initial solution follows a simple greedy heuristic, it searches for the first available timeslot in a classroom to assign a tuple T . The assignment of classroom and timeslots was done in increasing order in both cases, taking into account first the classroom and navigating through the different timeslots in it. In this phase, the constraint involves a

Table 6 Values tested for each of the considered parameters of the SA

Parameter	Values		
	0	1	2
T_i	15	10	1
T_f	0.001	0.000001	0.000000001
α	0.99	0.95	0.90
L	10	100	1000

maximum number of classroom is relaxed, allowing a greater number of classrooms, such that it is always possible to construct an initial solution. Then, it is expected to minimize this number during the construction phase.

4.3.3 Neighborhood function

The neighborhood function also followed a simple strategy; it was based on the roulette rule, which randomly chooses a teacher or a group having idle time. Once that the teacher/group is selected, their minimum m and maximum M timeslots are located in the cells of the matrix \mathcal{M} (that contains the actual solution). After that, a greedy search is performed to identify the couple (r, p) , where $m < p < M$ and r is a classroom that minimizes the idle time when $\mathcal{M}_{r,p}$ is exchanged with $\mathcal{M}_{m,p}$ or $\mathcal{M}_{M,p}$.

4.3.4 Evaluation function

The purpose of the study in this research is to analyze the performance of the SA when using different evaluation functions. To do that, a set of different evaluation functions were proposed, each of them measuring the objective values in the CB-CTT problem, minimizing the number of classrooms, minimizing the idle time of teachers, and minimizing the idle time of groups, where idle stands for the number of free hours. The evaluation functions proposed favor a previously established priority between the three objective exposed; these

Table 5 Set of evaluation functions used in the SA

Nomenclature	Meaning	Equation
CGT	Priority order is: classrooms, groups, and teachers	$ I_C \cdot F + I_G + \frac{ I_T }{\max I_T}$
GCT	Priority order is: groups, classrooms, and teachers	$ I_G \cdot F + I_C + \frac{ I_T }{\max I_T}$
TCG	Priority order is: teachers, classrooms, and groups	$ I_T \cdot F + I_C + \frac{ I_G }{\max I_G}$
CTG	Priority order is: classrooms, teachers, and groups	$ I_C \cdot F + I_T + \frac{ I_G }{\max I_G}$
GTC	Priority order is: groups, teachers, and classrooms	$ I_G \cdot F + I_T + \frac{ I_C }{\max I_C}$
TGC	Priority order is: teachers, groups, and classrooms	$ I_T \cdot F + I_G + \frac{ I_C }{\max I_C}$
ALL	There is no priority order	$ I_C + I_T + I_G $

Table 7 Instance of the CB-CTT problem taken from a real case derived from a Mexican University

Information	Amount	Description							
(a) General information									
No. of days	1	For the purpose of showing the performance, only one day is scheduled							
No. of timeslots	12	Each day has 12 non overlapping timeslots							
No. of courses	70	The courses to be scheduled during the day							
No. of teachers	27	Each teacher is available in any timeslot							
No. of classes per teacher	3	Each teacher can taught at most three classes							
No. of classrooms	17	Maximum number of classrooms available							
No. of groups	14	The curricula that describes the instance							
	Curriculum	Course	Lectures	Teacher	No.	Curriculum	Course	Lectures	Teacher
(b) No. Curricula description									
1	1	787	1	46	36	8	173	1	67
2	1	788	1	46	37	8	174	1	59,65
3	1	789	1	46	38	8	175	1	58
4	1	790	1	2, 28	39	8	214	1	36,64
5	1	791	1	48, 49, 56	40	8	215	1	2
6	2	790	1	2, 28	41	9	173	1	67
7	2	791	1	48, 49, 56	42	9	174	1	59, 65
8	3	790	1	2, 28	43	9	175	1	58
9	3	791	1	48, 49, 56	44	9	214	1	36, 64
10	4	161	1	45, 48	45	10	181	1	50
11	4	162	1	43	46	10	183	1	61
12	4	163	1	44, 47	47	10	184	1	3
13	4	164	1	45	48	10	185	1	53
14	4	205	1	36	49	10	186	1	50
15	4	206	1	75	50	10	222	1	24
16	4	207	1	3, 73	51	10	223	1	29
17	5	161	1	45, 48	52	11	187	1	63
18	5	162	1	43	53	11	188	1	60
19	5	163	1	44, 47	54	11	189	1	47, 60
20	5	164	1	45	55	11	190	1	50, 67
21	5	205	1	36	56	11	191	1	61
22	5	206	1	75	57	11	224	1	24
23	5	207	1	3, 73	58	12	187	1	63
24	6	170	1	48, 73	59	12	188	1	60
25	6	171	1	47, 58	60	12	189	1	47, 60
26	6	172	1	44, 56	61	12	190	1	50, 67
27	6	210	1	64	62	12	191	1	61
28	6	211	1	28	63	12	224	1	24
29	6	212	1	49	64	13	199	1	53, 72
30	7	170	1	48, 73	65	13	200	1	43, 44
31	7	171	1	47, 58	66	13	201	1	72
32	7	172	1	44, 56	67	13	203	1	63
33	7	210	1	64	68	14	199	1	53, 72
34	7	211	1	28	69	14	200	1	43, 44
35	7	212	1	49	70	14	201	1	72

Table 8 Values tested for each of the considered parameters of the SA

(a) Abstract $CA(9; 2, 4, 3)$				(b) $CA(9; 2, 4, 3)$																																								
<table border="1" style="margin-left: auto; margin-right: auto;"> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>2</td><td>2</td><td>2</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>2</td><td>1</td></tr> <tr><td>1</td><td>2</td><td>0</td><td>1</td></tr> <tr><td>2</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>2</td><td>1</td><td>2</td></tr> <tr><td>1</td><td>0</td><td>2</td><td>2</td></tr> <tr><td>2</td><td>1</td><td>0</td><td>2</td></tr> </tbody> </table>				0	0	0	0	1	1	1	0	2	2	2	0	0	1	2	1	1	2	0	1	2	0	1	1	0	2	1	2	1	0	2	2	2	1	0	2	Parameters				
				0	0	0	0																																					
1	1	1	0																																									
2	2	2	0																																									
0	1	2	1																																									
1	2	0	1																																									
2	0	1	1																																									
0	2	1	2																																									
1	0	2	2																																									
2	1	0	2																																									
Conf. No.	T_i	T_f	α	L																																								
0	15	0.001	0.99	10																																								
1	10	0.000001	0.95	10																																								
2	1	0.000000001	0.90	10																																								
3	15	0.000001	0.90	100																																								
4	10	0.000000001	0.99	100																																								
5	1	0.001	0.95	100																																								
6	15	0.000000001	0.95	1000																																								
7	10	0.001	0.90	1000																																								
8	1	0.000001	0.99	1000																																								

functions are summarized in Table 5. The selection of a hierarchically combined set of functions makes it possible to use a given obtained solution as initial solution of another execution of the approach; this methodology has advantage over weighting functions in which specific priority results can be achieved from them.

The meaning of the nomenclature is: $|I_C|$ is the number of classrooms required to do the assignment, and \max_{I_C} is the maximum number of existing classrooms. $|I_T|$ is the idle time of the teachers; where \max_{I_T} represents a value that exceeds the worst case scenario in the assignment of timetables to teacher classes. $|I_G|$ is the idle time of the groups; hence, \max_{I_G} represents a value that exceeds the worst case scenario of idle time in groups. Finally, the factor F is a large enough value which in combination with the maximum among \max_{I_C} , \max_{I_T} , \max_{I_G} avoid that one of the objectives of interest affects the others.

4.3.5 Simulated annealing parameters

The parameters of the SA algorithm considered in this research were the initial and final temperatures (denoted by T_i , and T_f , respectively), the cooling factor α , and the length of the Markov chain L . The values considered were taken from the literature [15, 17] and are shown in Table 6.

5 Experimental results

Fine-tuning of parameters is the process of adjusting the parameters of an algorithm to solve a problem. For the purpose

of adjusting and measuring the performance of the SA, a methodology based on Covering Arrays, as described in [10] is followed.

A covering array (CA) [3, 14] is an array M of size $N \times k$ consisting of N vectors of length k with entries from $\{0, 1, \dots, v-1\}$ (v is the size of the alphabet) such that every one of the v^t possible vectors of size t occurs at least once in every possible selection of t elements from the vectors. The parameter t is referred to as the covering strength.

The experimental design followed is based on a instance β of the CB-CTT problem, a CA ($N; t, k, v$) to study the effect of the interaction between parameters of the SA approach, and a set of different evaluation functions, described previously. In the experiment, the performance in terms of the quality of the solution of the SA approach, and the time spent to find it, were measured for each different configuration derived from the CA and the evaluation functions. Then, they were compared to identify the best evaluation function for the approach. For this purpose, each configuration was used to solve the instance 31 times, and the minimum, maximum, and average values of the performance were obtained.

The most relevant characteristics of the instance of the CB-CTT problem are described in Table 7.

The CA shown in Table 8 was used. The number of columns k is 4, each of them representing one of the parameters of the SA. The number of rows is 9, a set of combinations that cover all the different pair of values derived from the parameters, which, as defined in [3], represents a level of interaction $t = 2$. This characteristic allowed to reduce the number of test cases to minimize the time involved in the experimentation.

Table 9 Summary of the performance of SA using each of the different configurations considered

No.	Evaluation function	Configuration	<i>F</i>	Min. Cost	Max. Cost	Avg. Cost	Min. Time	Max. Time	Avg. Time
1	CGT	0	10,000	60032.105	60043.105	60035.976	186	1,032	276.51
2	CGT	1	10,000	60032.105	60042.050	60034.359	60	131	72.16
3	CGT	2	10,000	60028.121	60043.035	60035.121	38	127	51.12
4	CGT	3	10,000	60028.109	60043.035	60035.457	281	887	362.09
5	CGT	4	10,000	60030.105	60042.050	60034.109	4222	5391	4702.25
6	CGT	5	10,000	60031.109	60043.035	60035.191	246	1104	324.54
7	CGT	6	10,000	60030.105	60043.035	60034.875	8012	10668	9062.25
8	CGT	7	10,000	60030.105	60043.035	60035.421	1611	2652	2008.93
9	CGT	8	10,000	60030.105	60042.050	60033.691	24273	30060	26681.35
10	GCT	0	1,000	8.163	1009.148	73.794	134	1137	267.80
11	GCT	1	1,000	8.135	1010.120	170.564	47	155	70.83
12	GCT	2	1,000	8.145	1009.126	41.216	28	138	50.06
13	GCT	3	1,000	8.166	1010.111	170.691	202	897	280.54
14	GCT	4	1,000	8.154	1010.111	138.180	3225	6772	4413.48
15	GCT	5	1,000	8.151	1010.129	138.116	161	2342	337.90
16	GCT	6	1,000	8.141	1010.111	138.276	6023	9143	7447.61
17	GCT	7	1,000	8.141	1010.101	105.859	1195	1835	1433.03
18	GCT	8	1,000	8.151	1010.111	138.243	18421	25338	21491.87
19	TCG	0	1,000	39.470	40046.468	21976.814	191	1101	295.09
20	TCG	1	1,000	38.470	40046.468	11331.440	60	147	72.83
21	TCG	2	1,000	38.529	40046.468	15203.224	37	655	72.67
22	TCG	3	1,000	36.529	40046.468	12621.868	298	944	379.22
23	TCG	4	1,000	38.529	40046.468	14235.960	4344	6297	4868.35
24	TCG	5	1,000	39.470	40046.468	17783.710	242	1154	333.19
25	TCG	6	1,000	38.470	40046.468	18106.832	8540	17212	10819.77
26	TCG	7	1,000	36.529	40040.531	11653.583	1497	3153	1893.38
27	TCG	8	1,000	36.529	40046.468	16170.650	26232	30896	28083.48
28	CTG	0	10,000	60005.304	60013.269	60006.632	195	1283	317.41
29	CTG	1	10,000	60005.304	60013.269	60006.632	64	135	78.41
30	CTG	2	10,000	60005.304	60017.242	60006.570	41	145	55.41
31	CTG	3	10,000	60005.304	60013.269	60007.664	300	1552	446.38
32	CTG	4	10,000	60005.304	60013.269	60006.250	4379	5401	4920.54
33	CTG	5	10,000	60005.304	60013.269	60006.503	254	585	378.22
34	CTG	6	10,000	60005.304	60013.269	60006.753	8477	10135	9441.19
35	CTG	7	10,000	60005.304	60026.261	60007.179	1627	2264	1873.64
36	CTG	8	10,000	60005.304	60013.269	60006.828	26626	28933	28019.58
37	GTC	0	10,000	40.529	10048.529	684.390	156	1265	235.19
38	GTC	1	10,000	37.529	10046.529	1328.379	45	149	50.32
39	GTC	2	10,000	37.529	10043.587	1006.565	27	97	30.22
40	GTC	3	10,000	46.529	10046.529	1008.045	200	643	226.29
41	GTC	4	10,000	43.588	10046.529	1651.757	2894	5469	2904.00
42	GTC	5	10,000	42.588	10046.587	1007.206	181	273	175.41
43	GTC	6	10,000	40.588	10045.529	1652.079	5755	8600	5612.64
44	GTC	7	10,000	40.588	10033.587	362.337	1203	1689	1138.12
45	GTC	8	10,000	40.529	10047.529	1328.814	17136	22953	16571.03
46	TGC	0	10,000	39.470	40046.468	21976.814	188	848	306.48
47	TGC	1	10,000	38.470	40046.468	11331.440	58	647	92.93

Table 9 continued

No.	Evaluation function	Configuration	F	Min. Cost	Max. Cost	Avg. Cost	Min. Time	Max. Time	Avg. Time
48	TGC	2	10,000	38.529	40046.468	15203.224	38	179	55.87
49	TGC	3	10,000	36.529	40046.468	12621.868	288	1023	374.96
50	TGC	4	10,000	38.529	40046.468	14235.960	4178	7065	5214.06
51	TGC	5	10,000	39.470	40046.468	17783.710	239	489	327.90
52	TGC	6	10,000	38.470	40046.468	18106.832	8548	11951	9709.54
53	TGC	7	10,000	36.529	40040.531	11653.583	1,583	3551	1880.48
54	TGC	8	10,000	36.529	40046.468	16170.650	25,289	33,459	28706.96
55	ALL	0	—	24	43	31.903	164	1,056	311.93
56	ALL	1	—	26	43	32.580	56	144	77.77
57	ALL	2	—	28	43	33.161	35	121	49.61
58	ALL	3	—	26	42	30.709	259	949	343.19
59	ALL	4	—	29	40	31.612	4,110	8,824	4861.41
60	ALL	5	—	25	42	31.451	220	367	296.83
61	ALL	6	—	28	43	31.967	8,265	13,707	9776.83
62	ALL	7	—	27	42	32.774	1,580	2,468	1909.38
63	ALL	8	—	28	42	32.096	19,753	31,246	26401.29

The symbol—means that the factor F was not necessary in the evaluation function. The time given is measured in milliseconds. The bold configurations are the best for each evaluation function

A summary of the results derived from the experimental design are shown in Table 9. The number of evaluation functions required is of L at each iteration of the algorithm. There the time is presented in milliseconds. Also, it is presented the average quality of the solution obtained by each of the configurations tested. To understand the value of the evaluation function, the factor F is given; each evaluation function (with the only exception of ALL) has a priority in the minimization of the objectives: number of classrooms, idle time of teachers, and idle time of groups. Given a value C resulting from the evaluation of one of those functions, information corresponding to each element can be inferred from it, e.g., in the case of CGT, dividing C by F , and obtaining the integer part of the result gives the number of classrooms used (because this is the leading priority, the number of classrooms); by obtaining the module of the integer part of C by F , the idle time of groups (the second term of priority) can be obtained; and finally, the fractional part corresponds to the percentage of the total of hours of the teachers that are in idle time. The remaining functions can be understood in a similar fashion, considering the value of F , and the priority specified for each of them.

According with the results presented in Table 9, the expected outcome from each evaluation function was achieved, i.e., the hierarchical minimization of the entities considered. The summary of the objectives achieved is the following: (a) the maximum average time required was around 28.706 s, in the configuration (TGC, 8), meaning that the construction of a solution was performed in less than a minute; (b) the minimum number of classrooms in which the

schedule was programmed was of 6, which represents the use of a bit more of the 30 % of the available classrooms, and was achieved by evaluation functions CGT, CTG; (c) the idle time of teachers was reduced to 0 when using the functions TGC or TCG; and, (d) the idle time of groups was also reduced to 0 with the functions GCT, GTC.

Figure 2 summarizes the behavior of the best configurations for each evaluation function. These configurations were chosen because they minimize the different objectives in a balanced fashion, with a clear reduction in the priority value. Some of them quickly improve the initial solution, i.e., GTC, TGC, but maintain the same solution along the process. Others have a smoother curve, but achieving also good results at the end, i.e., CTG, GCT. Note that the convergence to a final solution by the SA varies from a flat curve, to a smoother one. The first type of curve presented in Fig. 2c, e, and f is mainly due to the number of iterations being small (they were 100), to improve the quality of the initial solution. On the other hand, the smoothest curve appears when no priority is specified, i.e., when all the entities have the same weight. In the latter case, the algorithm searches for a solution without minimizing a particular entity. It can be seen as a search without a specific purpose, which in turn will lead to a wide diversification of the achieved solution.

6 Conclusions

The research presented in this document involves the solution of a special case of the CB-CTT, which can be found in the

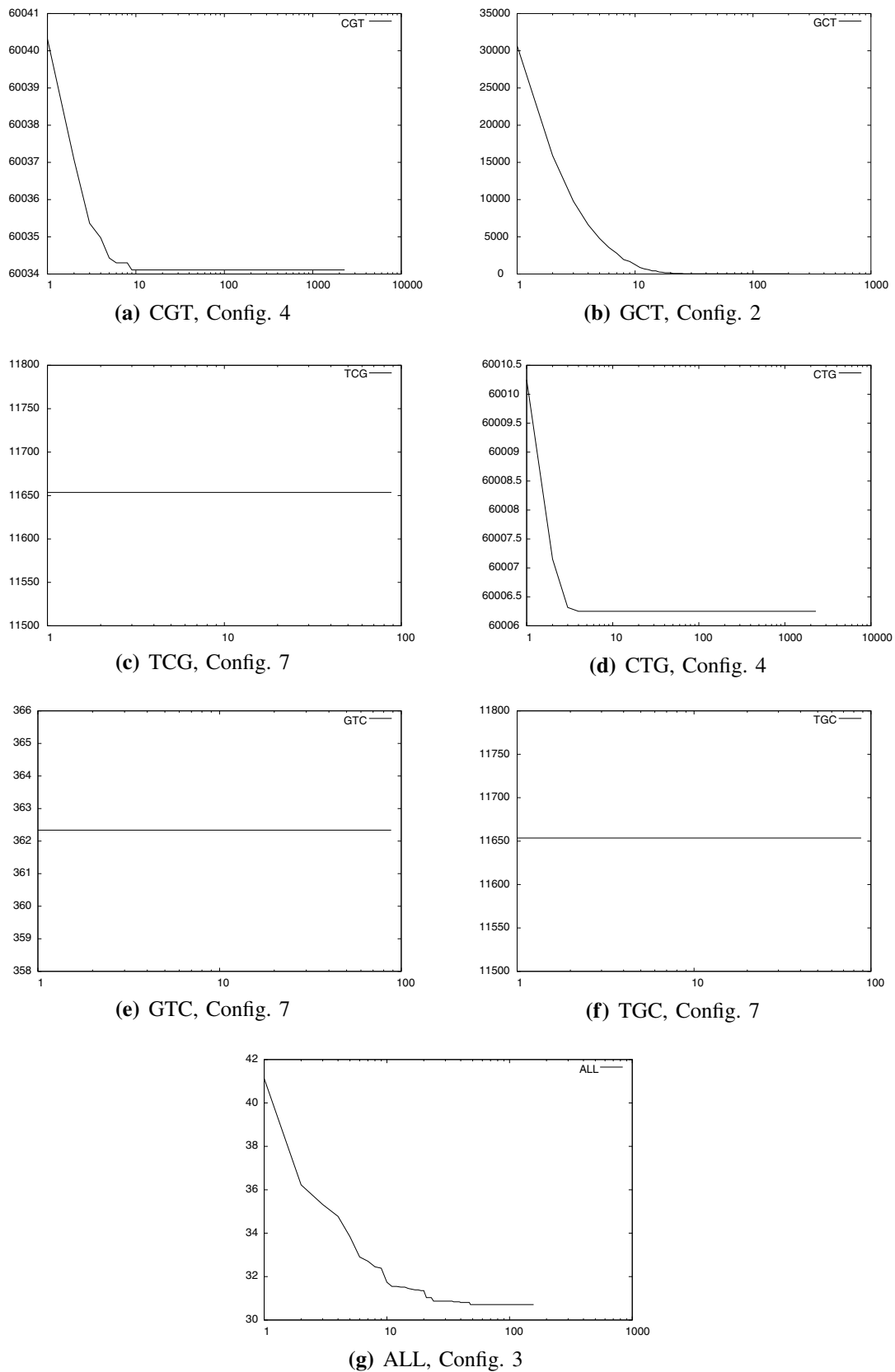


Fig. 2 Graph of the behavior of SA for each of the best configurations per evaluation function. The X axis denotes the number of evaluations during the search process. The Y axis describes the average quality of the solution

Polytechnic PUV, in Mexico, and has not been completely addressed in the literature.

The special case of the CB-CTT problem involved the inclusion of the constraint of not assigning a teacher to more than one course per curriculum, and the solution of the task that assigns courses to teachers. In this document, it is shown a formal definition for the problem.

The solution for the CB-CTT problem follows a novel strategy based on a SA approach, and in the relaxation of the main problem into three simpler subproblems. This relaxation reduces the search space, making it easier to find a solution. The reported results show the effectiveness of the approach to find a solution for the real case instance considered.

In addition, the performance of the SA was analyzed in different scenarios, each scenario involved the use of a different evaluation function which hierarchically organized a set of goals involving the minimization of: (a) classrooms used, (b) idle time of teachers, and (c) idle time of students (related to curricula). As expected, the performance of the SA varied according with the objective pursued.

Finally, the results shown an improvement in the time required for the construction, reducing it from over 2 weeks (necessary for a manual construction), to less than 1 min.

Future work in this research contemplates a post processing of the solution, e.g., using as initial solution of the SA a solution already constructed by it. Taking into account that a single run hierarchically minimizes one of three objectives, feeding the SA with one of these solutions can make further improvements, because minimizing a different goal could produce a reduced noise over the initial solution.

Acknowledgments Authors want to thank the support received from: the project “Estudio de los Sistemas de Información para la Integración en Base a la Calidad de Informacin y a la Experiencia de Usuario proporcionada” de FOMIX CONACYT; and the project *Análisis de la Dificultad del Problema de Minimización del Ancho de Banda en Grafos por Medio de Parejas de Invariantes* supported by PROMEP/103.5/12/3620 of the Programa de Mejoramiento al Profesorado (PROMEP).

References

1. Abdullah, S., Turabieh, H., McCollum, B., McMullan, P.: A hybrid metaheuristic approach to the university course timetabling problem. *J. Heuristics* **18**(1), 1–23 (2012)
2. Al-Milli, N.: Hybrid genetic algorithms with simulating annealing for university course timetabling problems. *J. Theor. Appl. Inf. Technol.* **29**(2), 100–106 (2011)
3. Avila-George, H., Torres-Jimenez, J., Hernandez, V.: New bounds for ternary covering arrays using a parallel simulated annealing. *Mathematical Problems in Engineering*. Accepted on July 07 2012
4. Bonutti, A., Cesco, F., Gaspero, L., Schaerf, A.: Benchmarking curriculum-based course timetabling: formulations, data formats, instances, validation, visualization, and results. *Ann. Oper. Res.* **194**(1), 59–70 (2012)
5. Garey, M., Johnson, D.: *Computers and Intractability: A Guide to the Theory of NP Completeness*. Freeman, San Francisco (1979). ISBN 0-7167-1045-5
6. Kirkpatrick, S., Gelatt, C., Vecchi, M.: Optimization by simulated annealing. *JSTOR* **220**(4598), 671–680 (1983)
7. Lü, Z., Hao, J.K.: Adaptive tabu search for course timetabling. *Eur. J. Oper. Res.* **200**(1), 235–244 (2010)
8. Pillay, N.: A survey of school timetabling research. *Ann. Oper. Res.* 1–33 (2013)
9. Pothitos, N., Stamatopoulos, P., Zervoudakis, K.: Course scheduling in an adjustable constraint propagation schema. In: *IEEE 24th International Conference on Tools with Artificial Intelligence (ICTAI)*, vol. 1, pp. 335–343 (2012)
10. Rangel-Valdez, N., Torres-Jimenez, J., Bracho-Ríos, J., Quiz-Ramos, P.: Problem and algorithm fine-tuning—a case of study using bridge club and simulated annealing. In: *Dourado, A., Rosa, A.C., Madani, K. (eds.) IJCCI*, pp 302–305. INSTICC Press (2009)
11. Reeves, C.R. (ed.): *Modern Heuristic Techniques for Combinatorial Problems*. McGraw-Hill, New York (1995)
12. Ruey-Maw, C., Shih, H.F.: Solving university course timetabling problems using constriction particle swarm optimization with local search. *Algorithms* **6**, 227–244 (2013)
13. Schaerf, A.: A survey of automated timetabling. Technical report, Centrum voor Wiskunde en Informatica (CWI), Amsterdam (1995)
14. Torres-Jimenez, J., Rodriguez-Tello, E.: New bounds for binary covering arrays using simulated annealing. *Inf. Sci.* **185**(1), 137–152 (2012)
15. Weicker, K.: *Evolutionäre Algorithmen*. Teubner Verlag (2002)
16. Werra, D.D.: An introduction to timetabling. *Eur. J. Oper. Res.* **19**, 151–162 (1985)
17. Weyland, D.: Simulated annealing, its parameter settings and the longest common subsequence problem. In: *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation, GECCO '08*, pp. 803–810. ACM, New York (2008)
18. Wren, S.: Timetabling and rostering a special relationship? The Practice and Theory of Automated Timetabling: Selected Papers from the 1st International Conference. *Lecture Notes in Computer Science*, vol. 1153, pp. 46–75 (1996)