SAT Model for the Curriculum-Based Course Timetabling Problem

Nelson Rangel-Valdez[1], Jose Torres-Jimenez[2], Jorge Omar Jasso-Luna[1], Mario Humberto Rodriguez-Chavez[1]

[1] Universidad Politécnica de Victoria, Departamento de Tecnologías de la Información, Cd. Victoria, Tamaulipas, Mexico
[2] CINVESTAV Tamaulipas, Cd. Victoria, Tamaulipas, Mexico
nrangelv@upv.edu.mx, jtj@cinvestav.mx, ojassol@upv.edu.mx, mrodriguezc@upv.edu.mx

Abstract. Two widely used problems are the Satisfiability problem (SAT) and the Curriculum-Based Course Timetabling (CB-CTT) problem. The SAT problem searches for an assignment that make true a certain boolean formula. On the other side, the CB-CTT involves the task of scheduling lectures of courses to rooms, considering teacher availability, a specified curricula, and a set of constraints. Given the advances achieved in the solution of the SAT Problem, this research proposes a SAT Model of the CB-CTT problem, to aid in the construction of timetables. To demonstrate that the model can aid in the solution of real instances of the CB-CTT problem, a case of study derived from a university in Mexico was considered. This special case of CB-CTT involves the constraint where each teacher cannot teach more than one course in the same curriculum, which is included in the set of 3 hard constraints and 2 soft constraints analyzed in this research. According to the results obtained, the considered complete SAT solver required a few minutes to find a solution for the instance.

Keywords: Curriculum-based course timetabling problem, SAT model.

1 Introduction

The Curriculum-Based Course Timetabling (CB-CTT) is a problem that occurs 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 of 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 [1], 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 [2].

At the Polytechnic University of Victoria (PUV), the CB-CTT problem involves a complex combination of hard and soft constraints, that occurs 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 work presents a SAT model for the special case of the CB-CTT problem found at the PUV, which includes the constraint that no teacher can teach more than one course per curriculum. An instance derived from the problem was solved using a complete SAT solver, reported in the literature. It is important to point out that the approach reduces the time to construct the schedules from two weeks to a couple of minutes; it also satisfies the 3 hard constraints, and the 2 soft constraints considered for this paper.

The remaining of this article is organized in the following way: Section 2 formally defines the special case of the CB-CTT problem that, to the best of our knowledge, does not completely match its more general definition [3]. Section 3 presents the work related to the solution of similar scheduling problems. Section 4 shows the SAT model proposed for the solution of the special case of the CB-CTT problem. Section 5 shows the results of the experiments performed over an instance taken from the PUV. Finally, Section 6 presents the conclusions derived from the research.

2 Curriculum-Based Course Timetabling

The *Curriculum-Based Course Timetabling* (CB-CTT) problem found at the PUV represents a special case of the CB-CTT problem. In order to define it, the following basic elements are presented:

Time-slots. A day is split in a fixed number of non overlapping time slots, 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 time slots 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, Curriculum, 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 Time-slots. One unassigned time slot between two assigned time slots in a curriculum daily schedule, or a teacher availability chart, is considered an Idle Time slot (or IT).

Figure 1 presents an simplified entity-relationship model describing the data in the PUV necessary to create an instance of the CB-CTT problem.

A feasible timetable for the PUV is one in which all lectures have been scheduled at a time slot and a classroom, so that the hard constraints $\{H_1, ..., H_3\}$ are satisfied. In addition, a feasible timetable has some desirable conditions, which are described by the two soft constraints $\{S_1, ..., S_2\}$ considered int he problem.

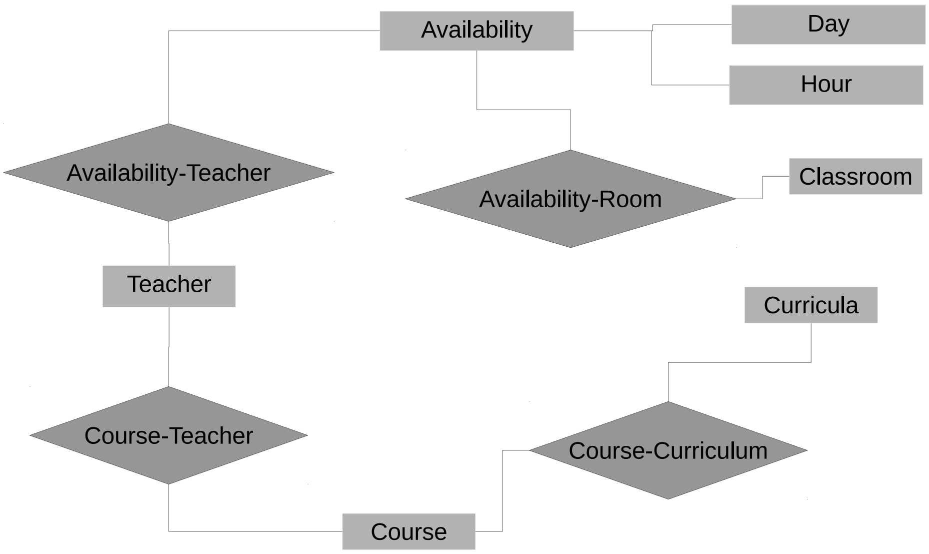

Fig. 1. Information required to derive a PUV-CBCTT instance

The three hard constraints (identified as H_i) and two soft constraints (identified as S_i) are:

– H_1. **Lectures:** All lectures of a course must be assigned to a particular time slot, 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 time slot. Moreover, courses in the same curriculum cannot be scheduled with the same teacher. Additionally, two lectures cannot be assigned to the same classroom in a particular time slot.
– H_3. **Availability:** If the teacher of a course is not available at a given time slot, then no lectures of a course can be assigned to that time slot.
– 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.

According with the information previously presented, the specialized case of the CB-CTT problem mainly differs from others CB-CTT formulations, as the one presented in [4], in which: a) it is scheduled in a daily basis; b) it does not consider the room occupancy; and, c) it includes the additional constraint that a teacher cannot teach more than one course per curriculum. A formal definition for this problem, which will be named from now on as PUV-CBCTT problem, is presented in the next subsection.

2.1 Problem Formulation of PUV-CBCTT

Let's define $C = \{c_1, c_2, ..., c_n\}$ as the set of courses to be scheduled; $P = \{p_1, p_2, ..., p_u\}$ as the set of time slots in which a day is split; $T = \{t_1, t_2, ..., t_m\}$ as the set of teachers; $R = \{r_1, r_2, ..., r_w\}$ as the set of available classrooms; $CR = \{cr_1, cr_2, ..., cr_s\}$ as the set where each element cr_h, $1 \leq h \leq s$, is a curriculum; $TC = \{tc_1, tc_2, ..., tc_m\}$ as the set where each element tc_q, $1 \leq q \leq m$, contains the group of courses that teacher q might taught; and, $TA = \{ta_1, ta_2, ..., ta_m\}$ as the set where each element ta_q, $1 \leq q \leq m$, contains the time slots in which teacher q is available to teach a lecture. Then, the PUV-CBCTT problem can be defined as the problem of finding a timetabling array $\mathcal{M}$ of size $w \times u$, where each cell $m_{i,j}$, for $1 \leq i \leq w$ and $1 \leq j \leq u$, contains a tuple $(cl_k, t_q) \in C \times T$, subject to:

1. $\bigcup_{\forall i,j} \mathcal{C}(m_{i,j}) = C$;
2. $\mathcal{CR}(m_{i_1,j}) \bigcap \mathcal{CR}(m_{i_2,j}) = \emptyset$, for any $i_1 \neq i_2$ having fixed the time slot j;
3. $\mathcal{T}(m_{i_1,j}) \bigcap \mathcal{T}(m_{i_2,j}) = \emptyset$, for any $i_1 \neq i_2$ having fixed the time slot j;
4. $\mathcal{T}(m_{i_1,j_1}) \bigcap \mathcal{T}(m_{i_2,j_2}) = \emptyset$, for any different pairs $(i_1, j_1), (i_2, j_2)$, where $\mathcal{CR}(m_{i_1,j_1}) = \mathcal{CR}(m_{i_2,j_2})$;
5. $\mathcal{C}(m_{i,j}) \in tc_{\mathcal{T}(m_{i,j})}$, for any i, j;
6. $p_j \in ta_{\mathcal{T}(m_{i,j})}$, for any i, j.

where $cl_{k,l}$ stands for the lecture l of course c_k; $\mathcal{C} : m_{i,j} \rightarrow c_k$ is a function that obtains the course c_k assigned to time slot j in classroom i; $\mathcal{CR} : m_{i,j} \rightarrow cr_h$ is a function that obtains the curriculum cr_h associated with the course c_k derived from $\mathcal{C}(m_{i,j})$; and, $\mathcal{T} : m_{i,j} \rightarrow t_v$ is a function that finds the teacher t_q associated to the course c_k derived from $\mathcal{C}(m_{i,j})$. Finally, it is important to point out that the idle time must be minimized during the search for a solution. A summary of the previously described sets is presented in Table 1.

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

Symbol	Meaning
$C = \{c_1, c_2, ..., c_n\}$	A set of courses to be scheduled.
$P = \{p_1, p_2, ..., p_u\}$	A set of time slots in which a day is split.
$T = \{t_1, t_2, ..., t_m\}$	A set of teachers.
$R = \{r_1, r_2, ..., r_w\}$	A set of available classrooms.
$CR = \{cr_1, cr_2, ..., cr_s\}$	A set where each element cr_h, $1 \leq h \leq s$, is a curriculum.
$TC = \{tc_1, tc_2, ..., tc_m\}$	A set where each element tc_q, $1 \leq q \leq m$, contains the group of courses that teacher q might taught.
$TA = \{ta_1, ta_2, ..., ta_m\}$	A set where each element ta_q, $1 \leq q \leq m$, contains the time slots in which teacher q is available to teach a lecture.

3 Related Work

Several authors, among them, Schaerf [5] and Werra [6], consider that the automatization of the Timetabling Problem (TTP) cannot be done completely. 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 [6] 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 CB-CTT problem, that has been developed in the recent years.

Lu and Hao [4] shows an approach based in Tabu Search that solves the more general case of the CB-CTT. 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 [7] reports the use of a hybrid strategy for the solution of the Educational Timetabling Problem. The approach combines Simulated Annealing and Genetic algorithms to solve a problem where the entities considered were courses, classrooms, students, and time slots. This problem differs from the case of the CB-CTT in which several constraints are not considered, e.g. the inner problem does not consider the assignment of teachers to courses and classrooms, nor the availability of teachers.

Pothitos et al. [8] describes the course timetabling problem in a similar way than Lu and Hao do in [4]. 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. [9] presents 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 [10] 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 SAT Model for Solving the PUV-CBCTT Problem

The SAT model proposed for the solution of the PUV-CBCTT problem is formed by seven diferent sets of restrictions with the following purposes: 1) to guarantee that each class must be assigned to just one classroom; 2) to guarantee that each lecture must be assigned to just one hour; 3) to avoid overlaps in classrooms; 4) to avoid scheduling classes in hours were the teacher is not available; and, 5) to avoid that two different classes of the same teacher or group be scheduled in the same time slot.

In order to define the restrictions of the model the boolean variables depicted in Table 2 are defined.

Table 2. Boolean variables used in the SAT model of the CB-CTT problem

$$X_{i,j} \leftarrow \begin{cases} 1 \text{ if course } c_i \text{ is assigned to classroom } r_j \\ \\ 0 \text{ otherwise} \end{cases}$$

$$Y_{i,j} \leftarrow \begin{cases} 1 \text{ if course } c_i \text{ is assigned to time slot } p_j \\ \\ 0 \text{ otherwise} \end{cases}$$

4.1 Restrictions to Guarantee a Classroom per Class

The first set of restrictions consists in the assignment of classrooms to courses. For this purpose the sets defined in Equation 1 are proposed. There, while the first set guarantees that at least each course must have a classroom assigned, the second set assures that only one must be assigned.

$$\begin{gathered} \bigwedge_{i=1}^{n} \left(\bigvee_{j=1}^{w} X_{i,j} \right) \\ \bigwedge_{i=1}^{n} \bigwedge_{j=1}^{w} \bigwedge_{k=j+1}^{w} \left(\overline{X_{i,j}} \vee \overline{X_{i,k}} \right) \end{gathered} \tag{1}$$

4.2 Restrictions to Guarantee a Lecture of a Classroom is assigned to just one hour

Once that a classroom is assigned, the following step to consider in the definition of the SAT Model is the assignment of time slots to each lecture of a course. The sets of constraints shown in Equation 2 are defined for this purpose. The first set assures that a lecture has at least one time slot assigned, while the second complements the restriction to ensure that it has just one.

$$\begin{gathered}\bigwedge_{i=1}^{n}(\bigvee_{j=1}^{u} Y_{i,j})\\ \bigwedge_{i=1}^{n}\bigwedge_{j=1}^{u}\bigwedge_{k=j+1}^{u}(\overline{Y_{i,j}} \vee \overline{Y_{i,k}})\end{gathered} \tag{2}$$

4.3 Restrictions to Avoid Overlaps in Classrooms

In order to avoid that two different classes are taught in the same classroom the set of restrictions shown in Equation 3 is considered. This set forms a single clause for each different combination of two classes sharing the same classroom and time slot, such that if both of them are true, then the clause will be false, satisfying the required assignment.

$$\bigwedge_{i=1}^{w}\bigwedge_{j=1}^{u}\bigwedge_{m_1=1}^{n}\bigwedge_{m_2=m_1+1}^{n}(\overline{X_{m_1,i}} \vee \overline{Y_{m_1,j}} \vee \overline{X_{m_2,i}} \vee \overline{Y_{m_2,j}}) \tag{3}$$

4.4 Restrictions to Assure Teacher Availability

The present restriction must assure that the courses are taught in time slots where the teacher is available. For this purpose a set of unit clauses is formed per teacher, through the set of boolean variables $Y_{i,j}$. This set will contain a clause with a single literal $\overline{Y_{i,j}}$ for each course c_i and time slot p_j in which the professor that teaches that class is not available.

4.5 Restrictions to Avoid Overlaps in Teacher and Curricula Time slots

This subsection presents the set of restrictions shown in Equation 4. The purpose of them is the assignment of classes to classrooms, considering the constraint where a teacher cannot be assigned twice to the same curricula. The function $f : c_i \rightarrow t_j$ obtains the teacher t_j that teach the course c_i, and the function $g : c_i \rightarrow cr_j$ obtains the curricula cr_j to which the course c_i belongs to. The two-literal clauses take into account each possible combination of undesired situations for these restrictions, such that it will turn false the formula if one of them occurs.

$$\begin{gathered}\bigwedge_{\forall(m_1,m_2)|f(m_1)=f(m_2)}\bigwedge_{i=1}^{u}(\overline{Y_{m_1,i}} \vee \overline{Y_{m_2,i}})\\ \bigwedge_{\forall(m_1,m_2)|g(m_1)=g(m_2)}\bigwedge_{i=1}^{u}(\overline{Y_{m_1,i}} \vee \overline{Y_{m_2,i}})\end{gathered} \tag{4}$$

4.6 Analysis of the Model

This subsection summarizes the complexity of the SAT model presented in this section in Table 3. There, it is shown the approximated number of clauses and literals per clause required to transform an instance of the PUV-CBCTT to a SAT formula. The values n, w, u, correspond to the number of courses, rooms, and time slots in the PUV-CBCTT instance.

Table 3. Characterization of the clauses in the SAT formula resulting from the transformation of an instance of PUV-CBCTT, following the proposed SAT model

Restriction	No. of Clauses	Literals per Clause
1	n	w
	$n \cdot w \cdot w$	2
2	n	u
	$n \cdot u \cdot u$	2
3	$w \cdot u \cdot n \cdot n$	4
4	$n \cdot u$	1
5	$n \cdot n \cdot u$	2

Note that the overall formula require two sets of boolean variables, each having $n \cdot w$, and $n \cdot u$ clauses, respectively. The number of literals per clause are mainly less than or equal to 4, and just a few of sizes n and u. In general, it is possible to comment that the formula produced by the SAT model proposed is simple, and only requires a number of clauses proportional to $O(w \cdot u \cdot n \cdot n)$, and a number of boolean variables to $O(n \cdot w)$.

5 Experimental Results

In order to demonstrate the viability of the approach to solve the PUV-CBCTT problem, the instance of it described in Table 4 was considered. This instance was modeled using the SAT model presented in this paper, and solver using The complete SAT solver used for this problem was the boolean satisfaction and optimization library in Java, SAT4J[1].

A summary of the results derived from the experimental design are shown in Table 5; in this table, the symbol $-$ means that the factor F was not necessary in the evaluation function. The time given is measured in milliseconds. The configurations shown in bold are the best for each evaluation function. Note that the SAT solver could find a solution for this instance in a few seconds, which indicates that the model can be used to solve more complex cases of this problem.

[1] http://www.sat4j.org/

Table 4. Instance of the PUV-CBCTT problem, taken from a real case derived from a Mexican University

(a) General Information

Information	Amount	Description
No. of time slots	12	A day has 12 non overlapping time slots
No. of courses	70	The courses to be scheduled during the day
No. of teachers	27	Each teacher is available in any time slot.
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

(b) Curricula Description

No.	Curriculum	Course	Lectures	Teacher	No.	Curriculum	Course	Lectures	Teacher
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 5. Summary of the performance of SA using each of the different configurations considered

Number of Variables	3,594
Number of Clauses	841,550
Time to find a solution	3.726 seconds
Satisfiable	YES

6 Conclusions

The research presented in this document involves the solution of a special case of the Curriculum-Based Course Timetabling (CB-CTT), which can be found in the Polytechnic University of Victoria (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.

A SAT model for the CB-CTT problem is presented, and a real world instance is used to test its validity. The instance was solved using a complete SAT solver, which could find a solution in a few seconds satisfying all the considered restrictions, the 3 hard constraints and the 2 soft constraints.

Acknowledgements. Authors want to thank the support received from projects: 58554 Cálculo de Covering Arrays, 51623 CONACyT - Gobierno del estado de Tamaulipas; and, the PROMEP projects UPV-PTC-023, titled *Análisis de la Dificultad del Problema de Minimización del Ancho de Banda en Grafos por Medio de Parejas de Invariantes*, and UPV-CA-4, entitled *Estudio del Impacto de la Experiencia de Usuario en los riesgos de integración durante el desarrollo de Sistemas Generadores de Horarios.*

References

1. Garey, M., Johnson, D.: Computers and Intractability: A Guide to the Theory of NP Completeness. Freeman: San Francisco. (1979) ISBN: 0-7167-1045-5.
2. Pillay, N.: A survey of school timetabling research. Annals of Operations Research (2013) 1–33
3. Bonutti, A., Cesco, F., Gaspero, L., Schaerf, A.: Benchmarking curriculum-based course timetabling: formulations, data formats, instances, validation, visualization, and results. Annals of Operations Research **194**(1) (2012) 59–70
4. Lü, Z., Hao, J.K.: Adaptive tabu search for course timetabling. European Journal of Operational Research **200**(1) (2010) 235 – 244
5. Schaerf, A.: A survey of automated timetabling. Technical report, Centrum voor Wiskunde en Informatica (CWI), Amsterdam, The Netherlands (1995)
6. Werra, D.D.: An introduction to timetabling. . European Journal of Operational Research **19** (1985) 151–162

7. Al-Milli, N.: Hybrid genetic algorithms with simulating annealing for university course timetabling problems. Journal of Theoretical & Applied Information Technology **29**(2) (2011) 100 – 106
8. Pothitos, N., Stamatopoulos, P., Zervoudakis, K.: Course scheduling in an adjustable constraint propagation schema. In: Tools with Artificial Intelligence (ICTAI), 2012 IEEE 24th International Conference on. Volume 1. (2012) 335–343
9. Abdullah, S., Turabieh, H., McCollum, B., McMullan, P.: A hybrid metaheuristic approach to the university course timetabling problem. Journal of Heuristics **18**(1) (2012) 1–23
10. Ruey-Maw, C., S., H.F.: Solving university course timetabling problems using constriction particle swarm optimization with local search. Algorithms **6** (2013) 227–244