

COGNITIVE RHYTHMS AND EVOLUTIONARY
ALGORITHMS IN UNIVERSITY TIMETABLES
SCHEDULING

RITMOS COGNITIVOS Y ALGORITMOS
EVOLUTIVOS EN LA PROGRAMACIÓN DE
HORARIOS UNIVERSITARIOS

OMAR CASTRILLÓN*

Received: 25/Feb/2014; Revised: 6/Aug/2014;
Accepted: 28/Aug/2014

*Universidad Nacional de Colombia—Sede Manizales— Facultad de Ingeniería y Arquitectura—Departamento de Ingeniería Industrial—GTA en Innovación y Desarrollo Tecnológico, Campus la Nubia—Manizales—Código Postal 170001, Colombia. E-mail: odcastrillong@unal.edu.co

Abstract

The main purpose of this research is to design a methodology based on evolutionary algorithms to university timetable scheduling. This methodology will consider the students' cognitive rhythms, which establish that teaching certain subjects in specific time intervals is much better than other techniques. This project takes place in three phases. First of all, there is a description of the different techniques used to solve this problem. Then, a new methodology based on cognitive rhythms and evolutionary algorithms is proposed, and finally, different methodologies are compared to determine the best. It is concluded that evolutionary algorithms are more efficient than other techniques in the university timetable scheduling. Future lines of research will determine the impact of these techniques within the students' learning process.

Keywords: scheduling, university timetables, genetic algorithms, fitness.

Resumen

El propósito de esta investigación es diseñar una metodología basada en algoritmos evolutivos para la programación de horarios universitarios. Esta metodología considerará los ritmos cognitivos de los estudiantes, los cuales establecen que enseñar algunas materias en intervalos de tiempo específicos es mejor que otras técnicas. Este proyecto es desarrollado en tres fases. Primero se realiza una descripción de las diferentes técnicas empleadas para solucionar este problema. Posteriormente una nueva metodología basada en ritmos cognitivos y algoritmos evolutivos es propuesta. Finalmente diferentes metodologías son comparadas para determinar la mejor. Se concluye que los algoritmos evolutivos son más eficientes que otras técnicas en la programación de horarios universitarios. Futuras líneas de investigación determinarán el impacto de estas técnicas en los procesos de aprendizaje de los estudiantes.

Palabras clave: programación; horarios universitarios; algoritmos genéticos; función objetivo.

Mathematics Subject Classification: 97P30.

1 Introduction

For the last 50 years, the university timetabling problems have been achieved with different models, however this is a difficult non-polynomic problem to solve (NP-Complete). Different techniques have been made to university timetable scheduling [20, 24]; however, these methodologies do not consider the students' cognitive rhythms in the establishment of the best time to attend certain classes [22, 38]. In general, the subjects with a high level of difficulty should be taught in the middle of the school day. Those with moderate difficulty should be taught at the beginning of the school day, and in the same way, the easier subjects, should be taught at the end of the school day. Cognitive rhythms are directly associated with circadian rhythms, they include factors such as complexity, individual differences (sex, chronotype, style, motivation) urban and rural factors [33]. These factors are very marked in the colombian universities therefore the application of these cognitive rhythms also takes on special importance in the university context.

Basically, there are two ways of timetable scheduling. The first one is based on analytic techniques where the main objective is to offer a solution. Any solution that takes into account all restrictions (teachers, classrooms and special rooms) is accepted.

The second way is based on artificial intelligence techniques, which generate new and best solutions from old solutions. Therefore, it is possible to go through the whole search space to obtain a good solution in a few seconds [35, 34, 32, 7].

This article is designed in four sections: State of the art, Methods, Experimentation, Results, Discussions and conclusions. The first section covers the state of the art in the last fifty years. In the section named "*Methods*", a methodology based on evolutionary algorithms is proposed. In last section "*Results and Discussions*" this methodology is compared with other existing methodologies. Finally the results show that this methodology is at least 15% more efficient than others.

2 State of the art

Twenty years ago, the evolution of this problem was related by different articles: In 1992 the logic programming (Prolog) was used to solve it [13]. In the same year [14] an evolutionary algorithm based on taboo search

was proposed. Finally, during the same year two kinds of constraints and a punishment function were proposed [19].

Subsequently in 1996, a combination of data and data base knowledge were proposed. Four years later [15] some special considerations (group sections and special constrains) were proposed to solve this problem, and, in 2001, methodologies based on integer programming were developed in Access to solve this problem.

In 2002, multi-criteria objectives were considered to solve this problem [11]. One year later [2], a constraints methodology based on search techniques was proposed to solve this problem. In [39] a computer network system was proposed to solve this problem. Afterwards, in [12] an integer program with cost function and search reduced space was proposed.

Similarly, in [9] a solution based on integer programming was proposed. In [26] some particularities that permit the student design its own curriculum were solved based on integer linear programming. In [40] specific constraints were considered: all student must have a least two classes, there is a maximum student number, the students have a maximum number of difficult subjects for semester.

In 2008, an evolutionary meta-heuristic solution was proposed by [17] for this objective. In [25] the gap between theory and practice was evidenced. Later that year, in [23] an integer programming fast model was proposed. One year later (2009) in [18] some extra constraints were included in new meta-heuristics. Similarly that year, in [10] a hierarchical low level combination was proposed. Finally that year, in [28] a hybrid algorithm based on harmonic algorithms and climber optimizers was proposed.

In 2010 a methodology based on meta-heuristic Monte Carlo and hybrid meta-heuristics with the aim to minimize the linear combinations were proposed by the authors [4]. In [3] a hybrid algorithm between taboo search and greedy search was proposed. In [21] the author first resolves the hard constraints and then the soft constraints.

In 2011, [29] a methodology based on graph coloring and heuristic multiples was proposed. Similarly, in [31] a mimic algorithm based on heuristics and knowledge databases was proposed. Finally, in [27, 36] an algorithm with an adaptive behavior based on Particles swarm optimization (PSO) was proposed.

In 2013, an adaptive hybrid algorithm based on bees' behavior was proposed [37]. Later in [33, 35] this problem was solved based on NSGA methodology for a school. In [1] a methodology based on integer linear

programming was proposed. The problem was split in two objectives. In [5] an adaptive linear combination between graph coloring and multiple heuristic was proposed. Finally other applications can be found in [6].

3 Methods

Methodology based on genetic algorithms is described. It has the following steps:

Step 1: Problem variables

For a solution to the problem, four multidimensional arrays are defined: data, scheduling, professors and classrooms. Table 1, and Figures 1 and 2 represent these variables.

Table 1: Data. This variable contains the initial definition of the problem. Adapted and translated [8].

Sem	Subj code	Subj name	Hour/ Gr week	Prof code	Prof name	Max hours /day	Room	Ideal time	Prof.2 code	Prof.2 name
1										
.										
N										

The variable named Classroom has a similar structure. Figures 1 and 2 show it.

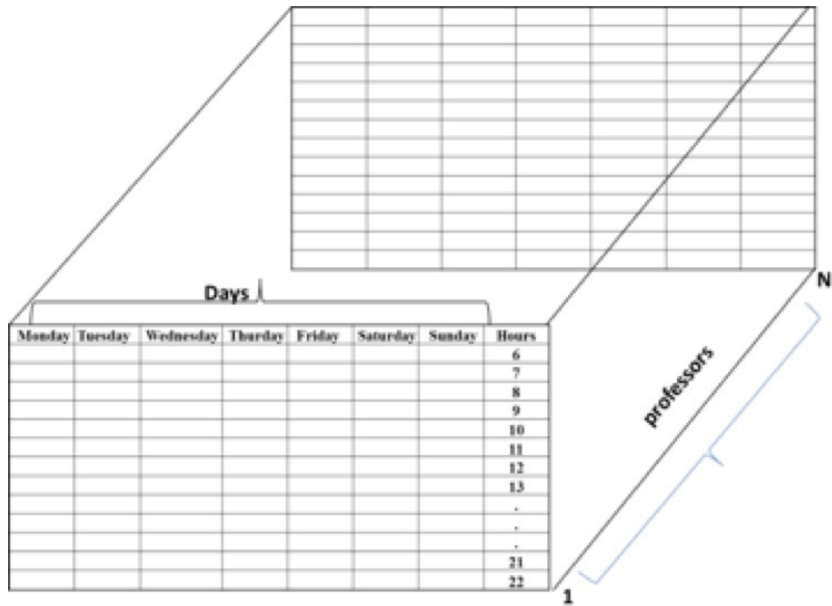


Figure 1: Multidimensional array, timetables professors. Adapted and translated [8].

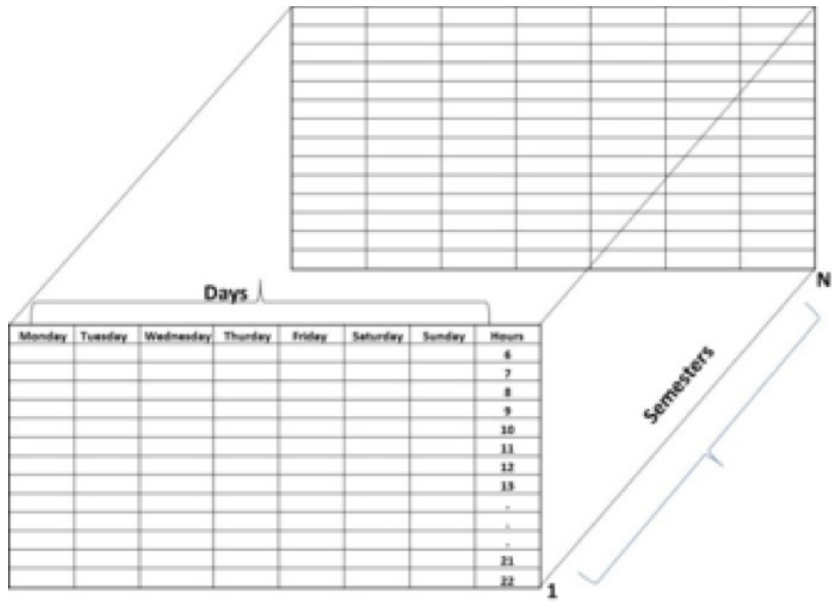


Figure 2: Multidimensional array, timetables scheduling. Adapted and translated [8].

This problem considered some hard constraints: professors, rooms, the quantity of hours by day, the subjects of the same semester can not be crossed, there are some unavailable time periods for teachers and rooms. Similarly, there are other soft constraints: The subjects should be programmed at an ideal time; the timetable should be the most compact possible. When all subjects (table 1) are programmed in the three tridimensional structures, the hard constraints are resolved. In other case, the problem does not have a solution. The scheduling of these subjects is described in the next step (2).

Step 2: Initial population

All subjects represented in Table 1, should be organized (according to table 2) into a timetable scheduling to obtain a solution. When the subject is scheduled in the tridimensional structures at the ideal time (field ideal time table 1), there is no penalty. However, it is not always possible and the subject could be assigned at a wrong time; in this case, the total solution has a penalty proportional to the scheduled time. Because of the fact that the penalty depends on the scheduling sequence it is necessary to define an initial random array called "Father" (Table 2).

Table 2: Subject scheduling sequence (Father).

1...	(Rows number of table 1 = N)

The chromosome represented in Table 2 contains the first N numbers without repetitions. (N represents the number of subjects that should be scheduled on the three dimensional structures) According to the Father (Table 2), all subjects are scheduled with three variables: scheduling, professors and classrooms that are exemplified by the algorithm showed in following lines:

```

Times = p
Evolution = 1
Get (Father, Size_father) /*Initial Father*/
While evolution  $\leq$  times
    Success = 0
    Penalty = 0
    Tot_pen = 0

```

```

Cont = 1
While (cont ≤ size_father) and (Success == 0)
  Programming (Subject, professors, classroom, timetabling,
  success, penalty, father)
  /*The subjects are programming on the three
  dimensional structures */
  /*When it is not ideal time, the subject is penalized*/
  Cont = Cont + 1
  Pentot = Pentot + Penalty
Wend
If (success = 0) /*The solution is possible*/
  If Fitness < bestfitness /*The best solution is saved*/
    Bestfitness = Fitness
    Lasevolution = Evolution
    Save(solution)
  End If
  If fitness > Worstfitness /*The worst solution is saved*/
    Worstfitness = Fitness
    Save(solution)
  End if
Else /*When the solution is not possible a new father is
  generated.*/
  New_father
End If
If evolution mod 33 = 0 /*3*/
  Mutation
Else
  Combination
End If
Evolution = Evolution + 1
Wend
If success = 1 /*If this variable has 1 the algorithm does not find a
  possible solution*/
  Mgs("Solution does not exist")
End if

```

When the solution generated by table 2 is not possible this solution

is penalized with a fitness function equal to infinite. The solution is not feasible when is not possible to solve all hard restrictions.

Step 3: Algorithm

With mutation and combination operators, the values of tables 2 are changed. The combination operator has 99% of probability (the crossover process is used by this algorithm), the mutation has 1% (This process selects two random positions into the father vector to be changed). With these operators new fathers are generated. The algorithm evolves until the fitness function does not improve during K interactions. The fitness function is defined as the sum of differences between the ideal hour and programed hour for each subject.

The fitness function can be defined as:

$$Fitness_{Function} = \sum_{J=1}^{N(Subjects)} Abs(Ideal_{Hour} - TimetablesScheduling \times (Day, Programmed_{Hour}, Semester))$$

according to prove the methodology this process is repeated during ten times. The ten best results of the fitness function are selected. These results are compared with other artificial intelligence techniques (random algorithms, data mining and taboo search) with an analysis of variance under the model $y_i = \mu + T_i + \varepsilon_i$, where y_i represents the variable of the answer, T_i the effect caused by the i^{th} treatment, ε_i the i^{th} experimental error. The information collected must get over independence and normality requirements. Finally, the solution generated by traditional techniques is compared with the best solution obtained with this methodology.

Experimentation: A hypothetical problem is used to prove this methodology (Table 3): Three room types are considered (Normal Room, Multimedia Room, Computer Room). This problem is solved for two semesters (because the whole problem employs a lot of space). However this is NP-Hard problem with $13! = 6.227.020.800$ possibilities.

4 Results

This methodology produces the following data: steps 1-3. Tables 4 and 5 show the best results of these steps. In this timetable scheduling, the best fitness function found is 41 and the worst is 53.

Table 3: Problem Definition. According to the methodology, the possibilities are $13! = 6.227.020.800$.

Sem	Subj code	Subject name	Hour/week	Gr	Prof code	Prof name	Max hours/day	Room type	Ideal time hour	Prof.2 code	Prof.2 name
7	29	Simulation	3	1	5	Professor 5	3	3	9	0	
7	29	Simulation	3	2	5	Professor 5	3	3	9	0	
7	30	Quality	4	1	6	Professor 6	2	1	4	0	
7	31	Production	8	1	7	Professor 7	3	1	7	0	
7	31	Production	8	2	3	Professor 3	3	1	7	0	
7	32	Systems	6	1	8	Professor 8	3	3	9	0	
7	32	Systems	6	2	5	Professor 5	3	3	9	0	
7	33	Business	6	1	2	Professor 2	3	1	16	0	
8	34	Quality CE	4	1	3	Professor 3	2	1	16	0	
8	35	Plants	10	1	1	Professor 1	3	2	14	2	Professor 2
8	35	Plants	10	2	9	Professor 9	3	2	14	2	Professor 2
8	36	Projects	4	1	10	Professor 10	2	1	16	0	
8	37	Financial	6	1	4	Professor 4	3	1	16	0	

Table 4: Timetabling 7th semester.

Hour	Monday	Tuesday	Wednesday	Thursday	Friday
7		Quality-1-AUL:1		Quality-1-AUL:1	
8	Production-1-AUL:1	Quality-1-AUL:1	Production-2-AUL:1	Quality-1-AUL:1	Production-2-AUL:1
9	Production-1-AUL:1		Production-2-AUL:1	Production-2-AUL:1	Production-2-AUL:1
10	Simulation-1-AUL:3	Systems-1-AUL:3	Systems-1-AUL:3	Production-2-AUL:1	Production-2-AUL:1
11	Simulation-1-AUL:3	Systems-1-AUL:3	Systems-1-AUL:3	Production-2-AUL:1	Systems-2-AUL:3
12	Simulation-1-AUL:3	Systems-1-AUL:3	Systems-1-AUL:3	Production-1-AUL:1	Systems-2-AUL:3
13	Systems-2-AUL:3	Production-1-AUL:1	Simulation-2-AUL:3	Production-1-AUL:1	Systems-2-AUL:3
14	Systems-2-AUL:3	Production-1-AUL:1	Simulation-2-AUL:3	Production-1-AUL:1	
15	Systems-2-AUL:3	Production-1-AUL:1	Simulation-2-AUL:3	Production-1-AUL:1	
20		Business-1-AUL:1	Business -1-AUL:1		
21		Business-1-AUL:1	Business -1-AUL:1		
22		Business-1-AUL:1	Business -1-AUL:1		

Table 5: Timetabling 8th semester.

Hour	Monday	Tuesday	Wednesday	Thursday	Friday
12	Plants-1-AUL:2	Plants-2-AUL:2			Plants-2-AUL:2
13	Plants-1-AUL:2	Plants-2-AUL:2			Plants-2-AUL:2
14	Plants-1-AUL:2	Plants-2-AUL:2	Plants-1-AUL:2		Plants-2-AUL:2
15	Plants-2-AUL:2	Plants-1-AUL:2	Plants-1-AUL:2	Plants-2-AUL:2	Plants-1-AUL:2
16		Plants-1-AUL:2	Plants-1-AUL:2	Plants-2-AUL:2	0
17	Projects-1-AUL:1	Plants-1-AUL:2	Financial -1-AUL:1	Plants-2-AUL:2	Financial-1-AUL:1
18	Projects-1-AUL:1	Projects-1-AUL:1	Financial -1-AUL:1	Quality CE-1-AUL:1	Financial-1-AUL:1
19	Quality CE-1-AUL:1	Projects-1-AUL:1	Financial -1-AUL:1	Quality CE-1-AUL:1	Financial-1-AUL:1
20	Quality CE-1-AUL:1				

Table 6: Repetitions by groups of 10.

Fitness	Repetitions										
	1	2	3	4	5	6	7	8	9	10	
Genetics	41	42	41	43	41	44	44	44	41	41	422
Random	42	45	41	41	42	43	41	42	56	43	436
Data Mining	42	41	43	44	42	41	43	42	45	45	428
Taboo search	41	42	41	43	41	42	45	43	41	41	420

Table 7: Variance analysis.

Source Varia	GL	SC	CM	Fcal	F Tab
Sum C Total		253.10			
Treatment	3.00	15.50	5.17	0.78	2.87
Exper error	36.00	237.60	6.60		
Total	39,00	253,10	11,77		
The model is not significative (99,95%)					

In order to analyze the consistency of the results, the process is repeated 10 times and compared with other techniques of artificial intelligence: genetic algorithms, random algorithm, data mining and taboo search. (Tables 6 and 7). According to the variance analysis the four methodologies compared are equal. The best fitness found with this methodology (41) is not compared with the solution generated by traditional techniques because it was not possible to solve the problem with these techniques. Traditional techniques solve the problem with the initial sequence, they do not provide a solution.

To build a feasible solution under traditional techniques it is necessary to define at least two normal rooms. In this case the best fitness function is 40. However, in the same case with the artificial intelligence techniques the best solution is 34. The fitness function is better at least by 15% than the traditional techniques.

5 Discussions and conclusions

Today, many techniques have been developed to solve this problem [30, 16]. However, in the literature there is not one solution for university timetable scheduling that considers students' cognitive rhythms. In this article, an evolutionary algorithm is formulated, to find the best solution. The so-

lution proposed is more efficient (15%) than traditional solutions. Additionally, the best values compared as is illustrated in the variance analysis show the system stability. Finally, the artificial intelligence techniques proposed in this study can be applied to solve general type of NP-Hard multi-objective problems.

Acknowledgment

This work is done as a research byproduct carried out during the author's sabbatical year and was funded by the "Universidad Nacional de Colombia", industrial engineering department.

References

- [1] Abdullah, S.; Alzaqebah, M. (2013) "A hybrid self-adaptive bees algorithm for examination timetabling problems", *Applied Soft Computing* **13**(8): 3608–3620.
- [2] Burke, E.K.; Petrovic, S. (2002) "Recent research directions in automated timetabling", *European Journal of Operational Research* **140**(2): 266–280.
- [3] Burke, E.K.; Mareček, J.; Parkes, A.J.; Rudová H. (2010) "Decomposition, reformulation, and diving in university course timetabling", *Computers & Operations Research* **37**(3): 582–597.
- [4] Burke, E.K.; Kendall, G.; Misir, M.; Özcan, E. (2012) "MonteCarlo hyper-heuristics for examination timetabling", *Annals of Operations Research* **196**(1): 73–90.
- [5] Cacchiani, V.; Caprara, A.; Roberti, R.; Toth, P. (2013) "A new lower bound for curriculum-based course timetabling", *Computers & Operations Research* **40**(10): 2466–2477.
- [6] Castiñeiras, I.; Sáenz-Pérez, F. (2013) "Applying CP(F D), CLP(F D) and CFLP(F D) to a Real-life Employee Timetabling Problem", *Procedia Computer Science* **18**: 531–540.
- [7] Castrillón, O.D. (2013) "Planning University Schedules Using Evolutionary Algorithms and Cognitive Rhythms", *EIIC-2013 2nd Electronic International Interdisciplinary Conference 2013, Budapest virtual conference*: 448–450.

- [8] Castrillón, O.D. (2014) “Combinación entre un Algoritmo Genético y Aleatorio para Programación de Horarios Universitarios Basado en Ritmos Cognitivos”, *Información Tecnológica* **25**(4): 51–62.
- [9] Daskalaki, S.; Birbas, T.; Housos, E. (2004) “An integer programming formulation for a case study in university timetabling”, *European Journal of Operational Research* **153**(1): 117–135.
- [10] De Causmaecker, P.; Demeester, P.; Vanden Berghe, G. (2009) “A decomposed metaheuristic approach for a real-world university timetabling problem”, *European Journal of Operational Research* **195**(1): 307–318.
- [11] Dimopoulou, M.; Miliotis, P. (2001) “Implementation of a university course and examination timetabling system”, *European Journal of Operational Research* **130**(1): 202–213.
- [12] Dimopoulou, M.; Miliotis, P. (2004) “An automated university course timetabling system developed in a distributed environment: A case study”, *European Journal of Operational Research* **153**(1): 136–147.
- [13] Fahrion, R.; Dollansky, G. (1992) “Construction of university faculty timetables using logic programming techniques”, *Discrete Applied Mathematics* **35**(3): 221–236.
- [14] Ferland, J.A.; Lavoie, A. (1992) “Exchanges procedures for timetabling problems”, *Discrete Applied Mathematics* **35**(3): 237–253.
- [15] Foulds, L.R.; Johnson, D.G. (2000) “SlotManager: a microcomputer-based decision support system for university timetabling”, *Decision Support Systems* **27**(4): 367–381.
- [16] Goerigk, M.; Obel, A. (2013) “Improving the modulo simplex algorithm for large-scale periodic timetabling”, *Computers & Operations Research* **40**(5): 1363–1370.
- [17] Head, C.; Shaban, S. (2007) “A heuristic approach to simultaneous course/student timetabling”, *Computers & Operations Research* **34**(4): 919–933.
- [18] Hernández, R.; Miranda, J.; Rey, P.A. (2008) “Programación de Horarios de Clases y Asignación de Salas para la Facultad de Ingeniería

- de la Universidad Diego Portales Mediante un Enfoque de programación Entera”, *Revista Ingeniería de Sistemas* **22**: 121–141.
- [19] Hertz, A. (1992). “Finding a feasible course schedule using Tabu search”, *Discrete Applied Mathematics* **35**(3): 255–270.
- [20] Kahar, M.; Kendall, G. (2010) “The examination timetabling problem at University Malaysia Pahang: Comparison of a constructive heuristic with an existing software solution”, *European Journal of Operational Research* **207**(2): 557–565.
- [21] Lü, Z.; Hao, J.-K. (2010) “Adaptive Tabu Search for course timetabling”, *European Journal of Operational Research* **200**(1) 235–244.
- [22] Martínez, H.; Camargo, C.; Reyes, M. (2004) *Ritmos Cognitivos en la Escuela*. Universidad Pedagógica Nacional, Bogotá.
- [23] McCollum, B.; McMullan, P.; Burke, E.K.; Parkes, A.J.; Qu, R. (2008) “A New Model for Automated Examination Timetabling”, *School of Computer Science, University of Nottingham*: 1–24.
- [24] Mejía, J.; Paternina, C. (2010) “Asignación de horarios de clases Universitarias Mediante Algoritmos Evolutivos”, *Educación e Ingeniería* **5**(9): 140–149.
- [25] Mejía, J.C. (2008) *Asignación de Horarios de Clases Universitarias Mediante Algoritmos Evolutivos*. Universidad del Norte, Barranquilla.
- [26] MirHassani, S.A. (2006) “A computational approach to enhancing course timetabling with integer programming”. *Applied Mathematics and Computation* **175**(1): 814–822.
- [27] Özcan, E.; Parkes, A.J.; Alkan, A. (2012) “The Interleaved Constructive Memetic Algorithm and its application to timetabling”, *Computers & Operations Research* **39**(10): 2310–2322.
- [28] Pillay, N.; Banzhaf, W. (2009) “A study of heuristic combinations for hyper-heuristic systems for the uncapacitated examination timetabling problem”, *European Journal of Operational Research* **197**(2): 482–491.

- [29] Pillay, N.; Banzhaf, W. (2010) “An informed genetic algorithm for the examination timetabling problem”, *Applied Soft Computing* **10**(2): 457–467.
- [30] Rahman, S.; Bargiela, A.; Burke, E.; Özcan, E.; McCollum, B.; McMullan, P. (2014) “Adaptive linear combination of heuristic orderings in constructing examination timetables”, *European Journal of Operational Research* **232**(2): 287–297.
- [31] Soza, C.; Landa, R.; Riff, M.C.; Coello, C.A. (2011) “Solving timetabling problems using a cultural algorithm”, *Applied Soft Computing* **11**(1): 337–344.
- [32] Suárez, V.; Castrillón, O. (2011) “Diseño de una metodología basada en técnicas inteligentes para la distribución de procesos académicos en ambientes de trabajo Job Shop”, *XV Congreso de Ingeniería de la Organización*, Universidad Politécnica de Cartagena, España: 285–289.
- [33] Suarez, V. (2012) *Asignación de horarios de clase en instituciones de educación básica y media de acuerdo a ritmos de eficiencia cognitiva*. Tesis de Maestría, Universidad Nacional de Colombia.
- [34] Suarez, V.; Castrillón, O.; Guerrero, A. (2013) “Asignación de horarios de clase basado en los ritmos de aprendizaje de los estudiantes usando un algoritmo genético”, *Ingeniería y Ciencia* **9**(17): 77–95.
- [35] Suarez, V.; Guerrero, A.; Castrillón, O. (2013) “Programación de Horarios Escolares basados en Ritmos Cognitivos usando un Algoritmo Genético de Clasificación No-dominada, NSGA-II”, *Información Tecnológica* **24**(1): 103–113.
- [36] Tassopoulos, I.X.; Beligiannis, G.N. (2012a) “A hybrid particle swarm optimization based algorithm for high school timetabling problems”, *Applied Soft Computing* **12**(11), 3472–3489.
- [37] Tassopoulos, I.X.; Beligiannis, G.N. (2012b). “Solving effectively the school timetabling problem using particle swarm optimization”, *Expert Systems with Applications* **39**(5): 6029–6040.
- [38] Valentinuzzi, V.; Fontenele, J. (2008) “Los ritmos biológicos y el aprendizaje”, *Ingenierías* **XI**(40): 41–46.

- [39] Valouxis, C.; Housos, E. (2003) “Constraint programming approach for school timetabling”, *Computers & Operations Research* **30**(10): 1555–1572.
- [40] Van den Broek, J.; Hurkens, C.; Woeginger, G. (2006) “Timetabling problems at the TU Eindhoven”, *Electronic Notes in Discrete Mathematics* **25**: 27–28.