wjert, 2018, Vol. 4, Issue 3, 319-329. *Original Article* ISSN 2454-695X

World Journal of Engineering Research and Technology
WJERT
www.wjert.org SJIF Impact Factor: 5.218

DEVELOPMENT OF A PARTICLE SWARM OPTIMIZATION BASED CULTURAL ALGORITHM FOR SOLVING UNIVERSITY TIMETABLING PROBLEM

Alade O. M.[*], Oyeleye C. A., Adedeji O. T., Omidiora E. O. and Oabiyisi S. O.

Department of Computer Science and Engineering, Ladoke Akintola University of Technology, Ogbomoso, Nigeria.

Article Received on 28/03/2018 Article Revised on 17/04/2018 Article Accepted on 07/05/2018

***Corresponding Author**
Alade O. M.
Department of Computer Science and Engineering, Ladoke Akintola University of Technology, Ogbomoso, Nigeria.

ABSTRACT

Timetabling problems are search problems in which courses must be arranged around a set of timeslots so that some constraints are satisfied. However, slow convergence speed and high computational complexity are one of drawbacks limiting the efficiency of the existing timetabling algorithms. In this paper, Particle Swarm Optimization based Cultural Algorithm (PSOCA) was developed for solving university lecture timetabling problems. Particle Swarm Optimization (PSO) algorithm was used in the population space of cultural algorithm in order to improve the convergence speed of the algorithm. Experimental results confirmed that PSOCA was able to improve the performance of PSO to solve timetabling problem with promising result.

KEYWORDS: Cultural Algorithm, Particle Swarm Optimization Algorithm, Timetabling, University Course Timetabling.

1. INTRODUCTION

Timetabling is the allocation of given resources to objects being placed in space time, subject to constraints, in such a way as to satisfy a set of necessary objectives as virtually as possible (Oyeleye, *et. al.*,2012; Alade, *et. al.*, 2016).Timetabling problems can be divided into course and examination timetabling. The course timetabling problem basically involves the allocation of courses, rooms, and students to a stable time period, normally a working week, while satisfying a given number of constraints which are hard and soft constraints (Alade, Omidiora and Olabiyisi, 2014). Hard constraints are constraints that must be fulfilled, while soft constraints are to be fulfilled as much as possible (Burke and Newall 2003; Brailsford, Potts and Smith 1999).

Particle swarm optimization (PSO) is one of the evolutionary computational techniques and population-based search algorithms (Yuhui, 2004). The characteristics of PSO method makes it very prevalent, it has memory which is vital to the algorithm. Also it is simple to implement, it has ability to swiftly converge to a good solution, as compared with other optimization methods; it is faster, cheaper and more effective. Also, there are a small number of factors to be adjusted in PSO. Compared with genetic algorithms (GAs), the information sharing mechanism in PSO is significantly different (Zheng, Jie and Cui, 2004; Qinghai, 2010).

Cultural Algorithm (CA) is a technique that incorporates domain knowledge obtained during the evolutionary process so as to make the search procedure more effective (Reynolds and Zhun, 2001). The goal is to increase the learning or convergence rates of the algorithm so as to provide a better response to a large number of problems (Benjamin and Marcel, 2000).

Different methods had been applied to tackle the problems associating with timetabling. These include sequential methods that treat timetable problems as graph problems, cluster methods in which the problem is divided into a number of event sets, constraint based methods and meta heuristics methods such as genetic algorithms, simulated annealing, ant colony algorithm, cultural algorithm and other heuristic approaches (Carlos, 2000; Carlos, David and Gary, 2002; Chiarandini, 2006; Abdullah and Turabieh, 2008; Wilke, Grobner and Oster, 2002; Shengxiang and Sadaf, 2011; Alade, Omidiora and Olabiyisi, 2014). Hybrid of more than one Meta heuristics methods had also been proposed in literature. Examples are hybrid of genetic algorithms and fuzzy logic (Chaudhuri and Kajal, 2010), Hybrid of Genetic algorithms with Guided and Local Search Strategies (Shengxiang and Sadaf, 2011), hybrid of genetic algorithms and simulated annealing (Oyeleye *et al*., 2012), hybrid particle swarm optimization- constraint-based reasoning (Irene, Deris and Mohad, 2009), Modified PSO based Cultural Algorithm (Alade et.al. 2016) among others.

2. METHODOLOGY

In this paper, CA and MPSO were proposed for solving the timetabling problem. The cultural algorithm comprises of population space and belief space. In the population space of the cultural algorithm, Particle swarm optimization (PSO) was used. The algorithm was programmed in the MATLAB 8.1 (R2013a) environment with a system specification of 2.20GHz Processor, 500GB of HDD (hard disk drive), 6GB of RAM, and 64 bit operating system on window 7 platforms.

PSO Algorithm (PSO)

The main concept of the PSO algorithm consists of changing the velocity (accelerating) of each particle toward its gBest and lBest locations at each time step. Each particle is updated by following two best values in all the iteration. The best P_{id} previous position of the particle at the i^{th} iteration. The second one is tracked at its global best position from the first iteration to the i^{th} iteration. The velocity will adjust the particle movement which is based on the particle movement which is based on the particle's experience (cognitive) and experience of its neighbor (social component).

The PSO algorithm can be described as follows:

Step 1: Choose on how many particles that will be used to solve the problem. Every particle has its own position, velocity and best solution. Then

$$f\left(P_{id}^{i}\right) \leq f\left(P_{id}^{i-1}\right) \leq .... \leq f\left(P_{id}^{1}\right) \quad 2.1$$

Step 2: Estimate the fitness value of each particle

Step 3: If the fitness value of each particle's current position is better than its pbest, the pbest is set to the current position

Step 4: Fitness value of the particle is compared with that of the gbest. If it is better, the gbest is updated

Step 5: Update the velocity and position of each particles using

$$V_{id}^{i+1} = \chi\left(V_{id}^{i} + c_1 r_1\left(P_{id}^{i} - X_{id}^{i}\right) + c_2 r_2\left(P_{gd}^{i} - X_{id}^{i}\right)\right) \quad 2.2$$

$$X_{id}^{i+1} = X_{id}^{i} + V_{id}^{i+1} \quad 2.3$$

Step 6: The process is repeated from step 2 until the termination criteria is met.

V_{id} is the velocity component of the ith particle in the dth dimension

X_{id} is the position component of the ith particle in the dth dimension

χ is constriction factor

φ is sum of learning factors

q_1 and q_2 learning factors; (cognitive and social factor)

r_1 *and* r_2 are random numbers in [0, 1].

P_{id} is the individual historical best position of particle i in the dth dimension

P_{gd} is the historical best position component of the G_{best} in the dth dimension

$q_1 r_1 \left(P_{id}^{i} - X_{id}^{i}\right)$ is a cognitive component which measures the performance of the particles i relative to past performance

$q_2 r_2 \left(P_{gd}^{i} - X_{id}^{i}\right)$ is a social component which measures the performance of the particles relative to a group of particles or neighbors?

$\chi = \dfrac{2}{\left|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}\right|}$ and $\varphi = c_1 + c_2$, $\varphi > 4$

q_1, and q_2 together with r_1, and r_2 maintain the stochastic influence of cognitive and social components of the particles velocity respectively.

Cultural Algorithm Pseudo-Code

The cultural algorithm pseudo-code is given as follows:

1. Generate *s* random schedules (initial population)
2. Initialize the belief space (copying the best individual to the situational belief space)
3. Repeat
4. Evaluate Population P(t)
5. Update (B(t), Accept(P(t)))
6. Variation (P(t), Influence(B(t)))
7. t = t + 1
8. Select P(t) from P(t – 1)
9. Update the belief space (with the individuals accepted)
10. Until the end condition is satisfied.

The adopted cultural algorithm (CA) has the following steps.

Step1: Initialize the cultural algorithm parameters (belief space and population space).

Step 2: evaluate population space.

Step 3: update the belief space.

Step 4: Select the best particle from the pollution

Step 5: check if the stop conditions are satisfied. If no, go to step 2. Otherwise stop.

Formulation of Particle Swarm Optimization based Cultural Algorithm (PSOCA)

In formulating an PSOCA, particle swarm optimization algorithm (PSO) was substituted into the population space of the cultural algorithm framework. The formulated particle swarm optimization based cultural algorithm has the following steps:

Step 1: Initialize the algorithm parameters of PSOCA.

Step 2: Initialize the particle in both population space and belief space.

Step 3: Renew population space with PSO algorithm, compute the fitness of each particle, update and store the individual best particle of the population space.

Step 4: If accept condition is satisfied, carry on accept operation, send some better particles to belief space.

Step 5: If belief space satisfy reset condition, reset the best particle of belief space.

Step 6: Renew belief space with update formula, compute fitness of each particle, update and store the best particle of the belief space.

Step 7: If influence condition is satisfied, carry on influence operation, and substitute some better particles of belief space for some worst particle of population space.

Step 8: Check whether the stop condition is satisfy. If the stopping condition is not satisfied then go to step 3. Otherwise stop and obtain the best solution from the global best position

3. Problem Descriptions and its Mathematical Representation

The timetabling problems deal with effective allocation of rooms which are limited. No two courses should occupy one particular room at the same time. In this paper five days was used to allocate university courses. Each day contains 10 timeslots from 8am in the morning till 6pm in the evening.

The following important parameters are defined as follows:

E = {1..e} of events, each of which contains certain students and needs certain features

R = {1..r} of rooms, each of which has a seat capacity and its own features.

S = {1..s} of students, each of whom enrolls in some events

F = {1..f} of features, such as overhead projectors or special whiteboards

P = {1..p} of timeslots where p = 40 (5 days with 8 periods on each day)

D = $\{D_1..D_5\}$ of days where each day has 8 periods

Ordered subsets P^d of P corresponding to a period in a day d where

$P^1 = \{p_1, p_2 \ldots\ldots p_8\}$, $P^2 = \{p_9, p_{10}, \ldots\ldots, p_{16}\}$ …

An ordered subset $L^d = \{p_8, p_{16}, p_{24}, p_{32}, p_{40}\}$ that contains the last periods of each day.

$L^d \varepsilon$ P, d ε D^d

e, r, s, f, p are the number of events, rooms, students, features and timeslots respectively

s_r^R = the size of room r, e ε E

s_e^E = the number of students enrolled in event e, e ε E

$$w_{f,e} = \begin{cases} 1 & if\ event\ e\ requires\ feature\ f \\ 0 & otherwise \quad e\varepsilon E\ and\ f\varepsilon F \end{cases}$$

$$y_{f,r} = \begin{cases} 1 & if\ room\ r\ contains\ feature\ f \\ 0 & otherwise \quad r\varepsilon R\ and\ f\varepsilon F \end{cases}$$

$$t_{s,e} = \begin{cases} 1 & if\ student\ s\ is\ enrolled\ in\ event\ e \\ 0 & otherwise \quad s\varepsilon S\ and\ e\varepsilon E \end{cases}$$

Decision variables

x are binary decision variables indexed by events, rooms, and timeslots.

$$x_{e,r,p} = \begin{cases} 1 & if\ event\ e\ occured\ in\ room\ r,\ and\ time\ period\ p \\ 0 & otherwise \quad e\varepsilon E,\ r\varepsilon R\ and\ p\varepsilon P \end{cases}$$

C_s^{ldp} (last period of day): Its value representing the number of violations of soft constraint S_1 by student s.

C_s^{3R} (More than three events in a row): Its value representing the number of violations of soft constraint S_2 by student s.

C_s^{ld} (single class in a day): Its value representing the number of violations of soft constraint S_3 by student s.

C_s^{sr} (student and room ratio): Its value representing the number of violations of soft constraint S_4 by student s.

$z_{s,d}$ are binary decision variables indexed by student and day; their value indicates that student s has a single class in a day d. s ε S and d ε D^d

The objective function is given as follows

$$\text{Minimize} \quad \sum_{s\varepsilon E} \left(C_s^{ldp} + C_s^{3R} + C_s^{id} + C_s^{sr} \right) \qquad 2.4$$

C_s^{ldp}, C_s^{3R}, C_s^{ld} and C_s^{sr} consecutively describe the violations of the soft constraints S1, S2, S3 and S4 made against the will of each student. When each violation occurs in the solution, it will be penalized by 1. Soft constraints are described by Equations (2.9/2.4) to (2.142.9)

Subject to

$$\forall e \varepsilon E \qquad \sum_{r\varepsilon R}\sum_{p\varepsilon P} x_{e,r,p} = 1 \tag{2.5}$$

$$\forall s \varepsilon S \qquad C_s^{ldp} = \sum_{e\varepsilon E}\sum_{r\varepsilon R}\sum_{q\varepsilon L} t_{s,e} x_{e,r,q} \tag{2.6}$$

$$\forall s \varepsilon S \qquad C_s^{3R} = \sum_{\substack{i,j,k,\varepsilon E \\ i\neq j\neq k}}\sum_{r\varepsilon R}\sum_{\substack{p,q,m\varepsilon P \\ q=p+1 \\ m=q+1}} t_{s,i} t_{s,j} t_{s,k} x_{i,r,p} x_{j,r,q} x_{k,r,m} \tag{2.7}$$

$$\forall s \varepsilon S \forall d \varepsilon D^d \qquad z_{s,d} = \begin{cases} 1 & \sum_{e\varepsilon E}\sum_{r\varepsilon R}\sum_{i\varepsilon P} t_{s,e} x_{e,r,i} = 1 \\ 0 & otherwise \end{cases} \tag{2.8}$$

$$\forall s \varepsilon S \qquad C_s^{1d} = \sum_{q\varepsilon D^d} z_{s,q} \tag{2.9}$$

$$\forall r \varepsilon R \forall p \varepsilon P \qquad C_s^{sr} = \sum_{e\varepsilon E} s_e^E x_{e,r,p} \leq s_r^R \tag{2.10}$$

Equation (2.5) describes the implicit constraint which means that timetable solution must be complete and each event must be presented once. Equation (2.6) for (S1), equation (2.7) for (S2), equation (2.9) for (S3), equation (2.8) is necessary for describing (S3) which penalizes students who have only attended a single event in a day by 1, while equation (2.9) calculates all violations of any students for all days. Equation (2.10) for (S4). Also in equation (2.10) True represent 1 and false represent 0.

4. Hard and Soft Constraints

Hard constraints are the constraints that must be fulfilled, while soft constraints are the one to be fulfilled as much as possible. A feasible timetable is one in which all hard constraint are satisfied and nearly all soft constraint are satisfied too, while a non-feasible timetable is the one in which part of the hard constraint is not fulfilled even though all soft constraints are satisfied. In this research, the hard constraints under consideration are as follows:

H1: Lectures having students in common cannot take place at the same time

H2: Each classroom can only be used for one course in the same timeslot

H3: Lecturer cannot teach more than one course at a time

H4: No courses are to be conducted in the 13-14 hours and 15-17 hours each Friday and Wednesday as that slot are allotted for Muslim prayers and Sport respectively in LAUTECH

Concurrently, the following soft constraints were used:
S1: A student shall not have a class in the last slot of the day.
S2: A student shall not have more than three classes in a row.
S3: A student shall not have a single class on one day.
S4: The number of students that attend the course for each lecture, must be less than or equal to the number of seats of all the rooms that host its lectures.

Fitness Function

In this research work, the overall satisfaction is affected by the satisfactions of the class (room) and lecturer. Also the principal objective of this study is to find the optimal satisfaction of lecturers and classes with the results of course timetabling. Since soft constraints can increase such satisfaction, the fitness function will be defined as the result of subtracting the soft constraint penalty function from the total value of teacher and class satisfactions of the course timetable.

5. EXPERIMENTAL RESULT

The algorithm was tested with 858 courses and 135 venues; particle size is 20, with c_1 as 2.8, c_2 as 1.3. The results for standard PSO algorithm is shown in table 1 while table 2 show the result for PSOCA. The two algorithms were tested under 4 separated runs.

Table 1: Results after 4 runs for standard PSO.

Run	Time (sec)	Fitness value	Hard constraint Violation	Soft constraint violation	Number of subjects unallocated
1	35.59	0.83	0	2	58
2	32.33	0.85	0	2	60
3	33.31	0.84	0	2	60
4	40.33	0.89	0	2	62
Average	35.39	0.85	0	2	60

Table 2: Results after 4 runs for PSOCA.

Run	Time (sec)	Fitness value	Hard constraint Violation	Soft constraint violation	Number of subjects unallocated
1	35.57	0.90	0	0	0
2	37.02	0.87	0	0	0
3	39.23	0.92	0	0	0
4	38.90	0.89	0	0	0
Average	37.68	0.89	0	0	0

From table 1, the standard PSO have a total average number of 60 subjects unallocated compare to PSOCA that successfully allocated all the subjects. The main reason is that the standard PSO doesn't have the capability to handle constraint like PSOCA. From table 2, PSOCA successfully allocate all subjects, there is no violation of both hard and soft constraints. PSOCA generate higher best fit values and has a higher execution time as compared to the standard particle swarm optimization algorithm. From this result, it can be deduced that the particle swarm optimization based cultural algorithm improved the performance of the standard particle swarm optimization in term of generating more optimized timetable even though it takes longer time to execute.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we have been able to develop a Particle Swarm Optimization based Cultural algorithm for solving lecture timetabling problems in universities or other related higher institutions. The algorithm developed consisting of influence factors which are highly probable in minimizing high convergence speed associated with the standard Particle Swarm Optimization algorithm. It is recommended that future research may be geared towards modifying the population space (PSO), hybridizing the algorithm with another techniques to better improve the performance of the algorithm especially the computational time.

REFERENCES

1. Abdullah, S. and Turabieh, H (2008): Generating university course timetable using Genetic Algorithms and Local Search. The Third International Conference on Convergence and Hybrid Information Technology ICCIT, 1: 254-260.
2. Alade O.M, Omidiora E.O and Olabiyisi S.O (2014): Development of a University Lecture Timetable using Modified Genetic Algorithms Approach", International Journal of Advanced Research in Computer Science and Software Engineering, 4(9): 163-168
3. Alade O. M., Oyeleye C. A., Adedeji O.T., Omidiora E.O., Olabiyisi S.O. (2016): Development of a Modified Particle Swarm Optimization based Cultural Algorithm for solving University Timetabling Problem. *LAUTECH Journal of Engineering and Technology*, 2016; 10(1): 1-5.
4. Benjamin Franklin and Marcel Bergerman.(2000): Cultural algorithms: Concepts and experiments. In *Proceedings of the 2000 Congress on Evolutionary Computation*, Piscataway, NJ, 2000. IEEE Service Center., 1245–1251.

5. Brailsford, S.C., Potts C.N. and Smith, B.M. (1999): “Constraint Satisfaction Problems: Algorithms and Applications,” *European Journal of Operational Research*, 119: 557-581.
6. Burke, E. K. and Newall, J. P. (2002): "A New Adaptive Heuristic Framework for Examination Timetabling Problems," Working Group on Automated Timetabling, University of Nottingham, 8: 23-43.
7. Burke, E.K. and Newall J.P. (2003): "Enhancing Timetable Solutions with Local Search Methods", The Practice and Theory of Automated Timetabling, Lecture Notes in Computer Science. Springer, 2740: 195-206.
8. Carlos A.C, David A. V, and Gary B. L. (2002): *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, New York.
9. Chaudhuri A, and De Kajal (2010): “Fuzzy Genetic Heuristic for University Course Timetable Problem”. IJASCA, 2(1): 1-13.
10. Chiarandini, M., Birattari, M., Socha, K and Rossi-Doria, O. (2006): An effective Hybrid Algorithm for University Course Timetabling. *Journal of Scheduling*, 9(5): 403-432.
11. Irene, S., Deris, S., and Mohad Hashim, S. (2009): A combination of PSO and Local Search in University Course Timetabling Problem. International Conference on Computer engineering and Technology, 492-495.
12. Meysam S. K. and Mohammad S. A. (2012): Hybrid Genetic Algorithms for University Course Timetabling. International Journal of Computer Science Issues, 9(2): 446-455.
13. Oyeleye, C. Akinwale, Olabiyisi, S. Olatunde, Omidiora, E. Olusayo and Oladosu, J. Babalola: Performance Evaluation of Simulated Annealing and Genetic Algorithm in solving examination timetabling problem, Scientific Research and Essays, 7(17): 1727-1733.
14. Qinghai Bai. (2010): Analysis of Particle Swarm Optimization Algorithm. Computer and Information Science, 3(1): 180-184. (2001): Knowledge-based function optimization using fuzzy cultural algorithms with evolutionary programming. IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, 31(1): 1-18.
15. Reynolds R. G.(1994): An Introduction to Cultural Algorithms. In A. V. Sebald and L. J. Fogel, editors, *Proceedings of the Third Annual Conference on Evolutionary Programming* World Scientific, River Edge, New Jersey, 131–139.
16. Shengxiang Y. and Sadaf N. (2011): Genetic Algorithms with guided and local search strategies for university course timetabling. IEEE Transactions on systems, MAN and Cybernetics- Part C: Applications and reviews, 41(1): 93-106.

17. Thanh, N. D. (2006):" Solving timetabling problem using genetic and heuristics algorithms", Journal of Scheduling, 9(5): 403-432.
18. Towards Memetic Algorithms. Caltech Concurrent Computation Program, Report 790,
19. Wilke, P., Grobner, M., Oster, N. (2002): A Hybrid Genetic Algorithm for School Timetabling. In: McKay, B., Slaney, J. (eds.) AI 2002: Advances in Artificial Intelligence, Springer. Verlag, New York, 455–464.
20. Wren, A.(1995): "Scheduling, timetabling and rostering – A special relationship?" Proceeding of the 1st International Conference on the Practice and Theory of Automated Timetabling, 474-495.
21. Yuhui Shi (2004): Particle Swarm Optimization Electronic Data Systems, Inc. IEEE Neural Networks Society, 8-13.
22. Zheng, J., Jie Jing and Cui Zhihua (2004): Particle swarm optimization algorithm [M]. Science publishing Company of Beijing.