Volume 6, Issue 4, April 2016 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Automatic Generation of Timetable Using Firefly Algorithm

Deeptimanta Ojha
Department of CSE
ABIT, Cuttack, Odisha, India

Rajesh Kumar Sahoo
Department of CSE
ABIT, Cuttack, Odisha, India

Satyabrata Das
Department of CSE
VSSUT, Burla, Odisha, India

***Abstract*— One of the major challenges is how to generate a course timetable by metaheuristic algorithms. Already different searching techniques have been applied for course timetable but some demerits are the speed and effectiveness. This paper is to apply the firefly algorithm (FA) to optimize the course timetable which gives a better solution than other searching approaches with different parameter settings. Our main objective is to construct the course timetable through the Firefly Algorithm with their variations. Then the proposed algorithm is used to optimize that course timetable.**

***Keywords*— Optimization, Firefly Algorithm (FA), Course time table, Effectiveness.**

I. INTRODUCTION

The timetable is a time arrangement of different events that fulfils the task which is given by the event and makes sure that no time conflicts arises between the tasks and events. Timetable problem has been discovered in recent years. The level of research activity is increased in this area. There are different types of timetable problems are used such as teacher timetable, student timetable, nurse timetable and timetable of transportation. The preparation timetable manually in schools, colleges and universities which are a very time-consuming and tedious job and it requires lots of effort .Sometimes the proper use of resources is neither effective nor efficient by using the timetable approach. To overcome these problems and give a better result we propose to make an automated timetable system which will generate the timetable automatically by using the firefly algorithm. Firefly algorithm is a metaheuristic Method which is implemented in the course timetable problem with starting some initial solution. Meta-heuristic method includes some heuristic method which is inspired from nature and applies process-like nature by getting to get a population-based solution. The proposed system will take various inputs like a number of subjects, the number of teachers, subject limits of each teacher, preference value for each subject given by each teacher, etc. By taking the help of above all these inputs it will generate possible timetables making optimal use of all resources in a way that will best suit the constraints.

In 1996, Wren explains the timetable generation through the subject teachers' allocation, subject to different constraints, with various resources to objects, is being placed in space-time. It also satisfies a possible set of desirable objectives, as a result, a timetable specifies at which location and what time the teacher is allocated. In an academic institution, there are different courses are available, so there is no conflict of free timeslots available for every student within that time. Therefore, the teacher tries to find the timetable with the minimum conflicts [7]. An appropriate timetable is then chosen from the optimal solutions generated through firefly algorithm. Timetable design is the task to create a timetable while it satisfies some valid constraints. Firefly Algorithm (FA) is suitable for solving high dimensional and nonlinear problems. In this paper, we proposed firefly algorithm which generated the automated timetable and optimized that timetable.

This paper is organized as follows: Section-2 illustrated to literature survey and research in this specific area.Section-3 described firefly algorithm.Section-4 explained the proposed approach and working on proposed approach. Section 5 describes the pseudo code of Firefly Algorithm used in solving time table problem. Section-6 illustrated the result and discussion and Section-7 described the conclusion and future scope.

II. LITERATURE SURVEY

Adriano Denise [1] compared the PSO to Genetic Algorithm (GA) in generating lecturer timetable schedule. Based on the computational results, the amount of penalty obtained by the PSO is much smaller than the GA on 500th iteration. Betar and Khader [2] explained how harmony search is used for University Course timetabling problem which managed to find a solution which is nearer to the optimal solution. Bhaduri, A[3] focused how timetabling research is done by using a genetic algorithm where the local neighborhood search is used to explore the neighborhood solution or candidate solution through a genetic algorithm. Sophia et al. [5] described the timetable construction which satisfies all operational rules in an academic institution, at the same time timetable fulfills the wishes and requirements of the faculty members and the students. It is an important and difficult task for the staff those are involved. Generally, this task is left to the administrative staff to replicate the timetables of previous years with little changes to accommodate new situations. According to Emilio Fortunato et al. [6] the objective function derivative is needed for the initial position to be set by PSO. It also sets the feasibility of the initial position of the particles. Elizabeth et al.[9] described such as the appearance of the new lectures and exams during the semester which more difficult to handle. Lai et al.[8] described how the problems of examination timetabling and course timetabling with a small scale transaction. This paper also explained the

methods of artificial Intelligence and computational intelligence to develop the timetable problem. Lastly, this paper solves many problems faced by the teaching staff, such as handling preferences and subject allotment as it may vary in every semester.

III. FIREFLY ALGORITHM

Xin-She Yang[10] developed firefly algorithm in 2008. The Firefly Algorithm is a bio-inspired heuristic algorithm which is a population-based stochastic method which is derived and motivated by the flashing or mating behaviour of fireflies. The position of all fireflies represents a possible set of solutions and their light intensities represent corresponding fitness values or quality of all solutions.

There are three idealized rules of firefly algorithm:

1. All fireflies will be attracted to each other without considering their sex. All fireflies are considered to be unisex
2. Attractiveness is proportional to their brightness and decreases as distance among them increases.
3. The brightness of Firefly is associated with an objective function.

IV. WORKING OF THE PROPOSED APPROACH

In order to study the computational effort involved in solving the timetable generation problem through FA, the following mathematical programming model is proposed. We define the following sets to be used in the proposed model:

tno – total number of teachers available.

sno – total number of subjects to be taught.

pno- maximum number of subject preferences that a teacher can provide as his/her options.

tlim-an array which gives information about maximum number of subjects that a teacher can teach

For example, a faculty member might take more classes than a senior professor. So tlim value for faculty member might be higher than that of a senior professor. The value of tlim cannot be zero or negative.

smat- a 2-D array used to store preference matrix. Row represents the total number of subjects and Column represents the total number of teachers available.

The fitness function value of each solution is given by

```
For j=1 to sno
    Fx(i)=Fx(i)+Pref(i,j)*Prob(i,j)
End For
```

Where Fx(i) – denote fitness function value of candidate solution number ‘i’

Pref(i,j) – denote preference value of teacher for that particular subject

Prob(i,j)-denote probability of selecting a particular teacher and can be calculated as

$$prob(i,j)=1/tlim(i,j) \quad ...(1)$$

For our proposed problem, we have to maximize the fitness function value to get the optimal result. Initially, a set of candidate solutions is generated which satisfies all the required constraints. Then their corresponding fitness function values are calculated and the initial best solution is memorized. At the start of iteration, the fitness function value of a firefly is checked with another randomly chosen firefly. If the fitness function value of the current firefly is less than that of the randomly chosen firefly then the current candidate solution undergoes two stages. In first stage, any two slots of the candidate solution are chosen randomly and replaced with new random values. Then any one slot chosen randomly from the current candidate solution is replaced with the value of that slot position of the randomly chosen solution. If the fitness function value of the new solution is better than the current solution then it is replaced. At the end of the iterations, the best solution is calculated. The flow chart of time table generation by Firefly Algorithm is depicted in figure 1.

V. PSEUDO CODE OF TIMETABLE GENERATION BY USING FIREFLY ALGORITHM

```
Initialize the number of generation.
Initialize population size=10.
Evaluate its fitness function value 'fx'
Find the initial best solution
While generation<MAX do
  For i=1 to n number fireflies
    Select a firefly j at random
    If fitness (i) < fitness (j)
      Randomly change any two slots of the i^th solution.
      Copy two slot chosen at random from the j^th solution to the corresponding slot in the i^th solution.
      Evaluate the fitness function value of the new solution
      If (fitness(new)>fitness(old))
        then replace the older solution
      End If
      Find the current best solution
  End IF
     generation=generation+1
End While
Select the solution with best fitness function value which is the optimal result.
```


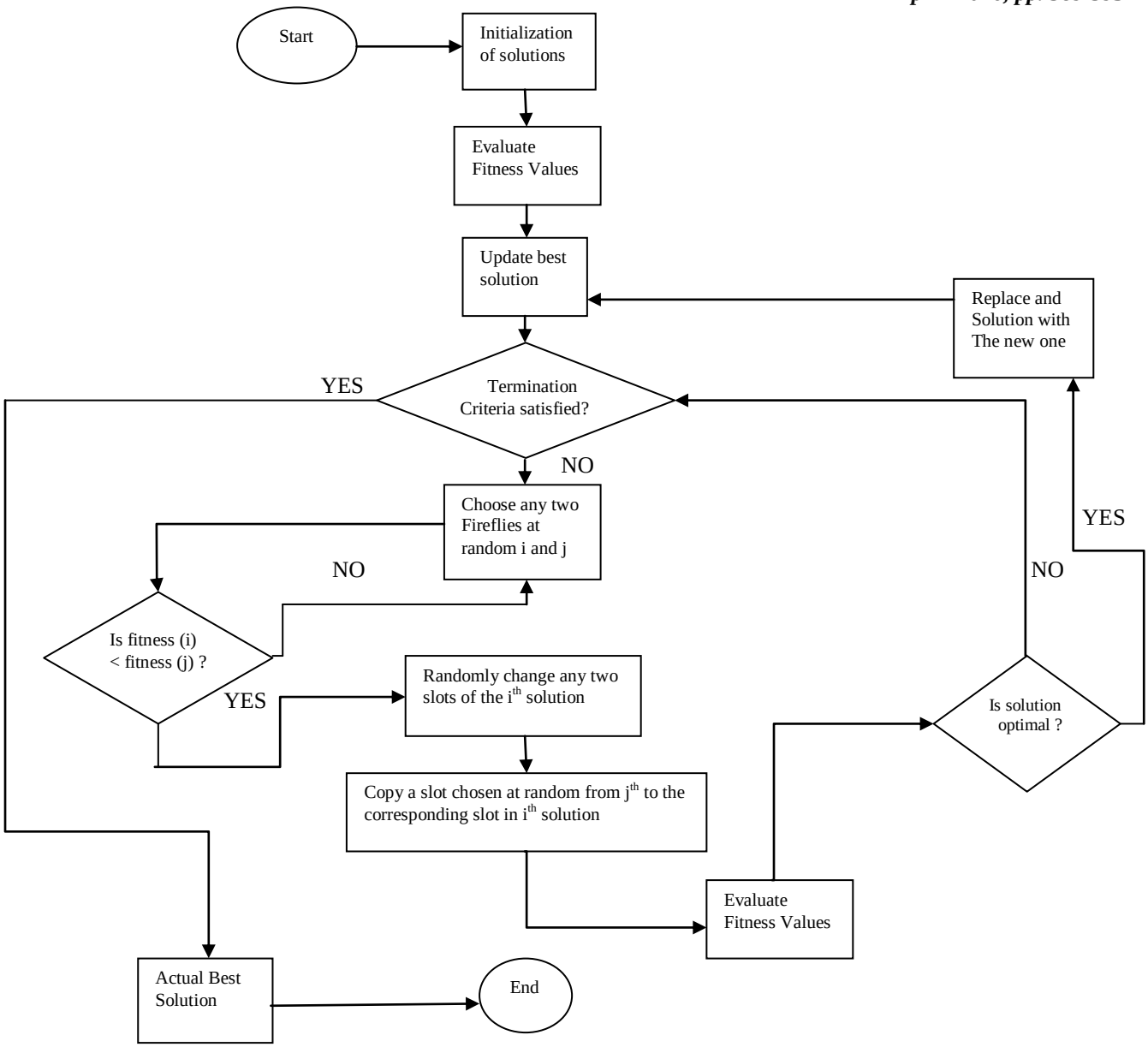

Figure 1. Flowchart of Timetable Generation using Firefly Algorithm

VI. RESULTS AND DISCUSSION

The timetable generation code was executed several times to get an optimal result using Firefly Algorithm. 10 different bees are used to produce 10 new firefly positions or candidate solutions. At the start of iteration, best firefly position or candidate solution is selected and that position is regarded as the current best solution. Table 2 shows the best candidate solution with their fitness function value at a specific iteration number. Here we have taken 20 test data at different iteration number. As shown in the figure Test data 1 i.e., the best candidate solution after iteration number 1 produces fitness function value of 20.3333. Then Test Data 2 i.e., the best candidate solution after iteration number 5 produces fitness function value of 24.5000. So there was an overall increase in 20.49% in fitness function value. Similarly after running the code for several iterations, it was found that the fitness function value of the candidate solution reaches its optimum value after 400 iterations. In this case the optimal fitness function value was found to be 45.1667.

Table 1 gives the information about preference values for each subject given by each teacher. T1 to T10 represents teachers whereas S1 to S20 represents subjects to be taught by the teachers.

Table 1 Preference Table given by each teacher for a particular subject

	T1	**T2**	**T3**	**T4**	**T5**	**T6**	**T7**	**T8**	**T9**	**T10**
S1	1	2	6	1	1	1	1	1	1	1
S2	1	1	1	1	1	4	1	6	1	1
S3	1	3	3	1	1	1	1	1	1	6
S4	4	1	1	1	1	1	2	1	2	1
S5	3	1	1	1	1	6	1	5	1	1

S6	1	1	5	1	4	1	1	4	3	1
S7	6	1	1	1	1	1	1	1	1	5
S8	1	1	1	1	1	1	3	1	4	1
S9	1	4	4	1	3	1	1	2	1	1
S10	1	1	1	1	1	5	1	1	1	1
S11	5	1	1	2	2	1	1	3	1	4
S12	1	1	1	1	1	1	5	1	1	3
S13	1	1	1	3	1	3	1	1	5	1
S14	1	5	1	1	1	1	1	1	1	2
S15	1	1	2	4	1	1	1	1	1	1
S16	2	1	1	6	1	1	1	1	1	1
S17	1	6	1	1	1	1	4	1	1	1
S18	1	1	1	5	1	1	1	1	6	1
S19	1	1	1	1	5	2	1	1	1	1
S20	1	1	1	6	1	1	6	1	1	1

Higher the preference value greater is their desire to take that subject.

For example, let us consider the first column of the preference table described above. It shows preferences given by Teacher number 1 for each subjects starting from 1 to 20.The preference value for Subject Number 7 given by Teacher Number 1 is 6 (which is the maximum value in this case).So T1 desires to take subject number 7 than any other subjects . Preference value '1' indicates that the particular teacher is least interested in taking that subject.

Table 2 shows possible solutions and their corresponding fitness function values at different iteration number

Table 2 Possible solutions at different iterations with their fitness function value

Iteration Number	**S 1**	**S 2**	**S 3**	**S 4**	**S 5**	**S 6**	**S 7**	**S 8**	**S 9**	**S 10**	**S 11**	**S 12**	**S 13**	**S 14**	**S 15**	**S 16**	**S 17**	**S 18**	**S 19**	**S 20**	**Fitness Function Value**
1	3	1	3	7	3	7	8	2	10	2	5	6	8	2	6	5	9	9	5	4	20.3333
5	3	4	1	9	7	1	1	8	5	6	1	1	1	2	4	5	2	10	5	4	24.5000
10	3	4	1	9	7	1	1	8	5	6	10	1	1	2	4	5	2	4	5	4	27.5000
20	9	8	2	5	6	6	1	8	5	6	10	7	1	2	4	5	2	4	3	4	30.0000
30	3	4	2	1	7	1	1	8	5	2	10	1	4	2	4	5	2	9	5	4	30.5000
50	9	8	2	1	6	6	1	8	5	6	10	7	4	2	4	5	2	4	5	4	33.3333
75	9	8	10	1	6	6	1	8	5	1	3	7	4	2	4	5	2	4	5	4	34.6667
100	9	8	10	1	6	3	1	8	3	1	1	7	4	2	4	5	2	4	5	4	37.3333
120	3	8	10	1	6	5	1	8	3	8	1	7	4	2	4	5	2	9	5	4	39.0000
150	3	8	10	1	6	3	1	8	3	6	1	7	4	2	4	5	2	9	5	4	39.1667
175	3	8	10	1	6	3	1	8	2	8	1	7	4	2	4	5	2	9	5	4	40.1667
200	3	8	10	1	6	3	1	8	2	8	1	7	4	2	4	5	2	9	5	4	40.1667
225	3	8	10	1	6	3	1	8	2	6	1	7	4	2	4	8	2	9	5	4	40.6667
250	3	8	10	1	6	3	1	8	2	6	1	7	4	2	4	8	2	9	5	4	40.6667
275	3	8	10	8	6	3	1	7	2	6	1	7	4	2	3	5	2	9	5	7	42.6667
300	3	8	10	8	6	8	1	7	2	6	1	7	4	2	6	5	2	9	5	7	42.8333
325	3	8	10	1	6	4	1	9	2	6	8	7	9	2	4	5	2	4	5	7	43.3333
350	3	8	10	1	6	3	1	9	2	6	1	7	9	2	4	5	2	4	5	7	44.8333
400	3	8	10	1	6	8	1	7	3	6	1	7	9	2	4	5	2	9	5	4	45.0000
500	3	8	10	1	6	8	1	9	2	6	1	7	9	2	4	5	2	4	5	7	45.1667

Let us consider the first row

3 1 3 7 3 7 8 2 10 2 5 6 8 2 6 5 9 9 5 4

It tells you that Subject Number 1 i.e., S1 is allocated to teacher number 3 i.e., T4, S2 is allocated to T1, S3 is allocated to T3 and so on. The fitness function of above given example is 20.3333.

Table 3 shows the final allocation of each subject to the corresponding teachers. According to the table, Subject number 4, 7 and 11 are assigned to Teacher number 1 .Then Subjects number 14 and 17 are assigned to teacher number 2 and so on. The optimized fitness function value is found to be 45.1667. The implementation of timetable management system through Firefly Algorithm is done using Matlab 7.0 and the result is depicted in the form of a graph in figure 2.

Table 3 Subject Allocation to Each Teacher

Teacher Number	**Subjects allocated to each teacher**
T1	S4,S7,S11

T2	S14,S17
T3	S1
T4	S15,S18
T5	S16,S19
T6	S5,S10
T7	S12,S20
T8	S2,S6
T9	S8,S13
T10	S3

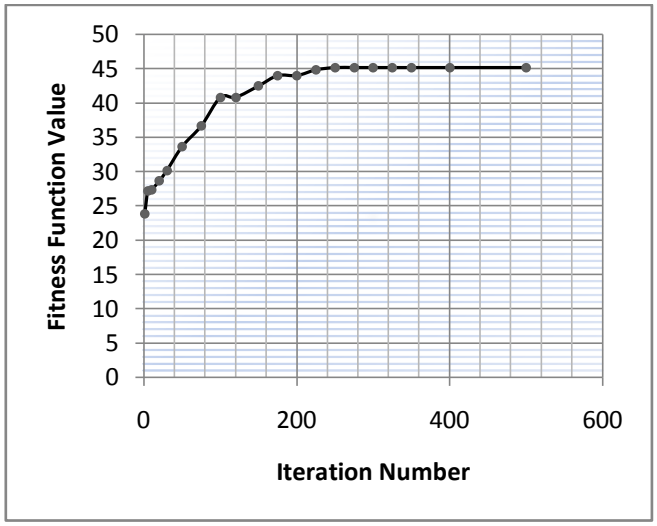

Figure 2. Graphical representation of Fitness Function Value v/s Iteration Number

VII. CONCLUSIONS AND FUTURE SCOPE

Generally in FA, fireflies attract each other to produce the optimal solution. Firefly Algorithm can solve non-convex problems with complex non-linear constraints. FA is promoting technique for solving complex problem such as course timetabling problem. So this work discusses FA to find solutions and this solution is useful to solve for designing the university course timetable. The solution is found with the characteristics of the proposed problem and also is able to improve the satisfaction of the teachers and classes toward the schedule in time table. Any conflicts between the faculty member's schedules, the class schedules, or the classroom schedules are also reflected in this work. The future scope is to optimize the course time table by using Particle swarm optimization along with comparison.

REFERENCES

[1] Adrianto Dennise, "Comparison Using Particle Swarm Optimization and Genetic Algorithm for Timetable Scheduling", Journal of Computer Science 10 (2),2014 pp- 341-346.

[2] Al-Betar, M.A. and A.T. Khader, 2009. A hybrid harmony search for university course timetabling. Proceedings of the Multidisciplinary International Conference on Scheduling: Theory and Applications Aug. 10-12, Dublin, Ireland, pp: 157-179.

[3] Bhaduri, A., 2009. University time table scheduling using the genetic artificial immune network. Proceedings of the International Conference on Advances in Recent Technologies in Communication and Computing, Oct. 27-28, IEEE Xplore Press, Kottayam, Kerala, pp: 289-292. DOI: 10.1109/ARTCom.2009.117

[4] Campana Emilio Fortunato, Diez Matteo, Fasano Giovanni, and Peri Daniele, "Initial Particles Position for PSO, in Bound Constrained Optimization", 2013, ICSI, Part I, LNCS 7928, Springer-Verlag Berlin Heidelberg.

[5] Daskalaki Sophia, Birbas Theodore, Housos Efthymios, "An Integer Programming Formulation for a Case Study in University Timetabling", European Journal of Operating Research: Elsevier, 2004.

[6] D. Dervis Karaboga, An Idea Based On Honey Bee Swarm for Numerical Optimization 2005, Technical Report-TR06,Erciyes University, Engineering Faculty, Computer Engineering Department.

[7] Ioannis X. Tassopoulos, Grigorios N. Beligiannis, "A hybrid particle swarm optimization based algorithm for high school timetabling problems", in applied soft computing 12(2012),pp- 3472-3489.

[8] Lai, L.F., C.C. Wu., N.L. Hsueh, L.T. Huang and S.F Hwang, 2008. An artificial intelligence approach to course timetabling. Int. J. Artif. Intell. Tools, 17: 223-240. DOI: 10.1142/S0218213008003868.

[9] Montero Elizabeth, Riff Maria-Christina, Altamirano Leopoldo, "A PSO Algorithm to Solve a Real Course+Exam Timetabling Problem",2011 ICSI: International Conference on Swarm Intelligence, France.

[10] X. S. Yang, Firefly Algorithm: Stochastic Test Functions and Design Optimisation, Int. J. Bio- Inspired Computation, 2010, Vol. 2, No. 2, pp.78–84.

