Journal of Theoretical and Applied Information Technology
15th January 2017. Vol.95. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

AUTOMATIC GENERATION AND OPTIMIZATION OF COURSE TIMETABLE USING A HYBRID APPRAOCH

[1]RAJESH KUMAR SAHOO, [2]DEEPTIMANTA OJHA, [3]DURGA PRASAD MOHAPATRA, [4]MANAS RANJAN PATRA

[1]Department of Computer Sc. & Engineering, A.B.I.T., Cuttack

[2]Department of Computer Sc. & Engineering, A.B.I.T., Cuttack

[3]Department of Computer Sc. & Engineering, N.I.T., Rourkela

[4]Department of Computer Sc. & Engineering, Berhampur University, Berhampur

E-mail: [1]rajesh_sahoo@rediffmail.com, [2]deeptimantaojha@gmail.com,

[3]durga@nitrkl.ac.in,[4]mrpatra12@gmail.com

ABSTRACT

Course timetable generation problem is a NP-hard problem where we have to take care of different constraints. Optimization problem is a technique of finding an alternative solution having cost effective or highly achievable performance subjected to given constraints, Optimization aims at maximizing desired factors and minimizing or reducing the undesired factors. This paper focuses the hybrid approach produced by combining the concept of Bee colony Optimization (BCO) and Firefly Algorithm (FA) collectively termed as BCFA for finding the optimal solutions of course time table. There are three objectives for construction of the paper, first objective is to get an overview on timetabling problem, second objective is the BCFA and its variations with other timetable generation algorithms and the third objective is to compare the result of BCFA with other evolutionary algorithms. The proposed approach aims at constructing or generating the course timetable and optimizing that timetable .

Keywords: *Bee Colony Optimization (BCO), Firefly Algorithm (FA), Particle Swarm Optimization (PSO), course timetable, hybrid approach*.

1. INTRODUCTION

The process of constructing time table manually for schools, colleges and universities is very time taking and requires lots of effort as we have to take care of various constraints and preferences given by various teachers. The resources are also not been properly utilized. In order to solve all these problems, eradicate all these drawbacks to produce a satisfactory result we develop an automated timetable generation system. The systems will the user for various inputs like total number of subjects to be taught, total number of teachers available, subject limits given by each teacher, subject preference given by each teacher, etc and by taking all these inputs it will generate possible time tables making optimal use of all resources provided to it. In 1996 Wren describes the timetable problem as the allocation of different subjects to different teachers which are subjected to various constraints. It also satisfies a set of objectives i.e., a timetable specifies at which location and time the teachers are allocated. The timetable must satisfy a number of requirements and desires of all people as much as possible. In a college, different courses are available, so there is no conflict of free timeslots available for every student within that time. Therefore a faculty member tries to find the timetable with the minimum chances of conflicts [9]. An appropriate timetable is then chosen from the optimal solutions generated. Timetabling is defined as a task to create a timetable without violating various constraints provided by the user. Basically constraints can be divided into two types, soft constraints and hard constraints. If we violate some constraints in scheduling but the output is still valid, then they are termed as sot constraints but hard constraints are defined as those constraints if we violate them then the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

timetable is no longer valid. BCFA is suitable for getting optimal result in less time.

The content of this paper is organized in the following structure: Section-2 describes the basic concept; Section-3 illustrates literature survey. Section-4 describes the proposed approach, necessity, working of proposed approach and methodology, Section-5 discusses the simulation result, Section-6 is all about discussions and future scope, Section-7 describes the conclusion.

2. BASIC CONCEPTS

2.1 Automated Timetable Generation

Timetable generation is the method of developing an optimized course timetable. The timetable generated should satisfy all the constraints provided by the user. While developing a time table many important factors are taken into account like subject preferences of each teacher, maximum number of subjects a teacher want to take, etc. Automated timetable generation helps to minimize the time, effort and cost in developing suitable course time table since manual developing of time table was ver tough and was not precise either.

2.2. Overview of Bee Colony Algorithm

It is a nature-based meta-heuristic algorithm which emphasizing on foraging behavior of honey bees for finding nectars or food sources. This proposed technique generates the optimal number of test cases which is robust and have high fitness value.

2.3. Overview of firefly algorithm

The Firefly Algorithm is a bio-inspired heuristic algorithm which is a population based stochastic method which is derived and motivated by flashing or mating behavior of fireflies. The attractiveness of fireflies is having a mutual relation with the brightness. The current best solution is represented through the fireflies with high intensity of light or attractiveness. The firefly will move randomly to search new better firefly in a particular iteration. The position of all fireflies represents a possible set of solutions and their light intensities represent corresponding fitness values or quality of the candidate solutions.

2.4. Overview of BCFA Algorithm

The proposed hybrid Firefly Algorithm (BCFA) is created or developed by merging the Bee Colony Optimization Algorithm with the approach used in firefly Algorithm. Here total population of the candidate solution is subdivided into two parts. One part of the solution undergoes Bee Colony Optimization (BCO) and another part undergoes Firefly Algorithm (FA).

3. LITERATURE SURVEY

According to Sophia et al. [7] construction of timetable should be done in such a manner that it satisfies all operational rules of the teachers and students. Adriano Denise [1] focused on how the lecturer time table is generated by using PSO and GA.GA gives the better result in comparison to PSO. Fen Irene et al. [8] described University Course Timetabling Planning UCTP) through hybrid particle swarm optimization which allocate weekly lessons where all students can attend their lessons without overlapping. Emilio Fortunato et al. [4] defined the objective function derivative which is needed for the initial position of particles in PSO. Shu-Chuan [5] described how the discrete PSO algorithm is used for scheduling exam timetable. In this case some soft constraints, such as preferences have to be handled. Lastly, this paper attempts to solve many problems faced by administrative staff, such as handling preferences as it may vary in every semester. Ojha et al.[11] explained how course time table is optimized by firefly algorithm in 400 iterations. Ojha et al. [12] explained how automated generation time table is optimized by Bee colony optimization algorithm in 250 iterations. Betar and Khader [2] focused on how the problem of course time table is generated through harmony search method and also find the solution very close to the optimal solution. Bhaduri, A [3] explained the genetic algorithm based time table research where the local neighborhood search is used to explore the candidate solution by using genetic algorithm. According to Lai et al. [10] the problems of examination timetable and course time table which is done by artificial Intelligence and computational intelligence methods with a small scale transaction. Sahoo et al. [14] explained how the automated test cases are generated and optimized by using different meta-heuristic algorithms like harmony search, particle swarm optimization and bee colony algorithms. According to this paper, bee colony algorithm generates the optimized test cases in very less iteration as compared to harmony search and particle swarm algorithm. Suresh et al. [15] represented that genetic algorithm (GA) is used

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

to generate the test data automatically through basis path testing. According to this paper indicates that GA is more effective and efficient to generate the test data automatically. According to Montero et al.[17] four parameters are used in PSO to generate the optimum solution for generation of automated time table of computer center. Mudjihartono Paulus et al. [18] focused on the time table generation from PSO which can solve many problems faced by administrative staff, like preferences available in every semester. Hertz [19] described the course time table problem using tabu search method and emphasizing to handle the problems to generate the course time table in large-scale with scheduling concept. Mooney et al. [20] explained how course time table is generated through a nonlinear integer programming model, by setting preference values, status, priority and classroom utilization rate.

4. PROPOSED APPROACH

This paper proposed a methodology for generating candidate solutions for course timetable and these possible solutions are optimized by various evolutionary algorithms like Firefly Algorithm (FA), Bee colony algorithm (BCA) and hybrid approach by combining FA and BCA collectively known as BCFA. These approaches are used to evaluate its efficiency and effectiveness for generating the optimized course time table.

4.1. Necessity Of Proposed System

The proposed system is intended to generate an optimized course time table with existing approaches of Firefly Algorithm, Bee Colony Algorithm and a new hybrid approach BCFA. In Bee colony algorithm (BCA) food source positions may be initialized by the system and each food source maintains its positions. In Hybrid Firefly Algorithm, the position coordinates of all fireflies represents a possible set of solutions and their light intensities represent corresponding fitness function values. This paper also finds out the effectiveness of the proposed approach through the teacher number and subject number to design and optimize the course time table.

4.1.1. Bee colony optimization

Bee Colony Optimization algorithm or simply BCO algorithm was developed by Karaboga[6] in the year 2005. BCO algorithm is based on foraging behavior shown by the honey bees. BCO is a bio-inspired meta-heuristic algorithm which is derived and motivated by the behavior shown by the honey bees to find adequate food source in the environment. All the food source positions represent a possible set of solutions and the amount of nectar represent corresponding fitness values or quality of the food source. There are mainly three types of bees used in basic Bee Colony Optimization Algorithm: Employed Bees, Onlooker Bees and Scout Bees. Initially a set of new food sources or candidate solutions are produced by Employed Bees while searching for food sources in the surrounding. Onlooker Bee chooses an Employed Bee to improve the quality of food source with the help of Roulette Wheel selection method. Scout Bees are used when a food source is exhausted. Scout Bees replaces that food source with a randomly generated new food source. In the timetable generation algorithm Scout Bees are not used. The steps for implementing Bee Colony Optimization are as follows:

1. Initial food sources are created randomly and their corresponding fitness values are evaluated. This information is given to all Employed Bees. The generation of new candidate solution is called as Employed Bee Phase.
2. In Onlooker Bee phase, selected food sources are improved to increase their fitness values. The selection is done using Roulette Wheel Selection method.
3. At end of iteration the candidate solution with best fitness value is memorized.

4.1.2. Firefly algorithm

Firefly Algorithm was proposed by Xin-She Yang[13] in 2008. The Firefly Algorithm can be defined as an evolutionary algorithm which is population based stochastic motivated by the flashing or mating behavior of fireflies. The position coordinates of all fireflies represents a possible set of solutions and their light intensities represent corresponding fitness function values.
Firefly algorithm is based on three ideologies:

1. All fireflies are considered to be unisex i.e., they are attracted to each other without considering their sex or gender.
2. Attractiveness between any two fireflies is related to their brightness and its value decreases as distance among these two fireflies increases.
3. Brightness produced by a firefly is always associated with an objective function.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

4.2. Working of the Proposed Approach

tno – denotes total number of teachers currently available.

sno – denotes total number of subjects to be taught.

pno - denotes maximum subject preferences that a teacher can give.

tlim - denote an array which gives information about maximum number of subjects a teacher is willing to teach.

smat - denotes a 2-D array used to store preferences of all teachers for a particular subject where row represents the subjects and column represents the teachers.

The fitness value or quality of each candidate solution is given by

for j=1 to sno
fx(i)=fx(i)+pref(i,j)*prob(i,j) (1)
end for

Where,
i – denotes i^{th} candidate solution
fx(i) – denotes fitness function value
pref(i,j) – denotes preference value of teacher 'i' for that particular subject 'j'.

prob(i,j)-denotes probability of selecting a particular teacher 'i' for teaching the subject 'j' and can be evaluated as

$$prob(i,j)=1/tlim(i,j) \quad (2)$$

4.3. Methodology

For the mathematical function:

$$fx = fx + pref*prob \quad (3)$$

Here the fitness function value of a particular candidate solution is determined with the help of preference value provided by a teacher denoted by "pref" and probability of allocating a teacher to subject denoted by "prob". The factor "prob" is calculated by taking inverse of "tlim" value given by the user which is the number of subjects a teacher is willing to take. So if the "tlim" value provided a teacher is less than the "tlim" value provided by another teacher then the first teacher is given more preference than the second teacher and hence "prob" value will higher than the second teacher. So the "prob" factor value always lies in between 0 and 1.i.e,

$$0 \leq prob \leq 1$$

Greater the value of "pref" and "prob" higher will the fitness function value of the particular candidate solution and hence greater will be the chances that the solution will move toward optimization.

Initially a set of candidate solutions are 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 solutions are ranked according to their fitness function value. Then half of the worst solution are discarded and best half solutions undergoes two phases in which one copy undergoes Bee Colony Optimization and another copy undergoes Firefly algorithm optimization.

In Phase 1, each solution undergoes through Employed Bee Phase and Onlooker Bee Phase. In Employed Bee Phase, any two slots of the candidate solution are chosen randomly and replaced with new random values. Then any one slot chosen randomly from the candidate solution is replaced with the value of that slot position of the current best solution. If the fitness value of the new solution is found to be better than the old solution then it is replaced. After Employed Bee Phase is completed, it undergoes Onlooker Bee Phase. But firstly relative fitness function value of each candidate solution is calculated. If any candidate solution is having relative fitness function value less a constant "pa" then that solution undergoes alteration by randomly replacing a slot in the candidate solution. Usually the value of pa lies in between 0 and 1. In this case the value of pa is taken as 0.1. If the fitness function value of the new solution is found to be better than the old solution then it is replaced.

In phase 2, any two fireflies are chosen at random. If the fitness function value of first firefly at i^{th} position is greater than the second firefly at j^{th} position then halt the process and try other pairs of fireflies otherwise move the i^{th} solution toward the j^{th} position solution do the following. Firstly any two slots of the candidate solution are chosen randomly and replaced with new random values. Then any two slots chosen randomly from the candidate solution are replaced with the value of that slot position of the j^{th} solution. If fitness function value of the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

new solution is found better than old solution then it is replaced. At the end of iteration, best solution is calculated and memorized.

4.4. Pseudo Code of BCFA

Initialize the number of generation.
Initialize population size=10.
Evaluate its fitness function value 'fx'.
Find the initial best solution and memorize it.

While generation<MAX do
Rank the solutions
Discard the bottom half solutions having worst fitness values.
Top half best solutions undergo updating in two phases separately.
Make two copies of best solutions.
One copy undergoes Bee Colony Optimization i.e., Phase 1
Another copy undergoes Firefly Algorithm i.e., Phase 2

Phase 1
For i=1 to bsize(number of bees)
//Employed Bee Phase
Randomly change any two slots of the candidate solution.
Copy a slot which is random chosen from the current best solution to the corresponding slot in the candidate solution.
Evaluate its fitness value
If(fitness(new)>fitness(old))
then replace the older solution
End If
//Probability Calculation Phase
Calculate the probability of occurrence of each solution
//Onlooker Bee Phase
If P> a random value in the range of [0,1]
Produce a new candidate solution
Evaluate its fitness value
If(fitness(new)>fitness(old))
then replace the old solution with the new one
End If
End If
End For

Phase 2
For i=1 to fsize(number of 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_value (new) > fitness_value (old))
then replace the older solution
End If
Find the current best solution
End If
End For
End While
Memorize the best candidate solution

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The flowchart of automated timetable generation using BCFA is depicted in the figure 1. .

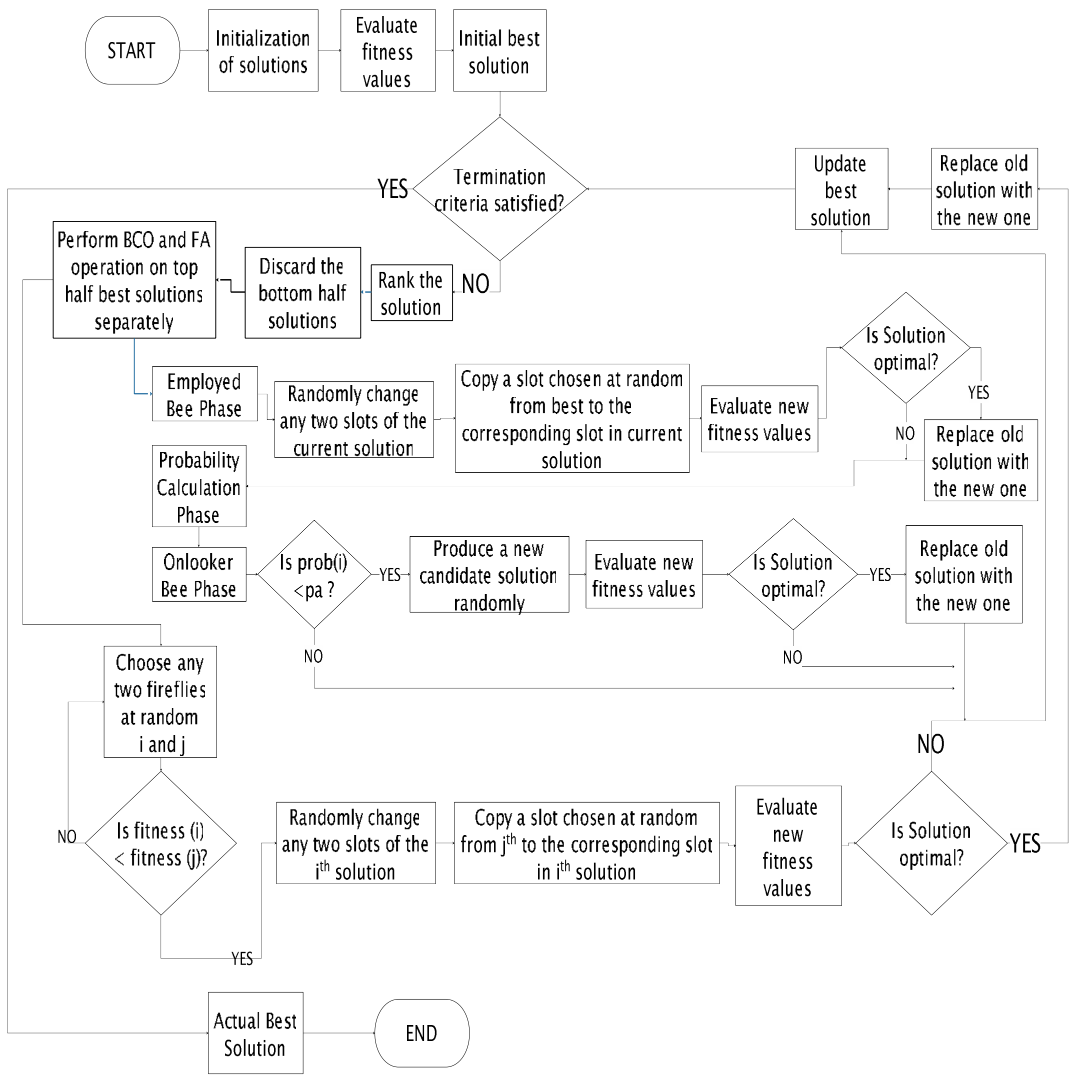

Figure 1:- Flowchart of Hybrid BCFA

5. SIMULATION RESULTS

The timetable generation code was executed several times to get an optimal result using Bee Colony Optimization Algorithm. 10 different bees are used to produce 10 new food source positions or candidate solutions. In iteration, best food source position or candidate solution is selected and that position is memorized.

Table 2 shows the best candidate solution with their fitness function value at a specific iteration number. Here we have taken 20 test cases at different iteration number. After running the code for several iterations, it was found that the fitness function value of the candidate solution reaches its optimum value after 120 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.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

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

Table 2 shows possible solutions and their corresponding fitness function values at different iteration number.
Higher the preference value greater is their desire to take that subject.

Table 2:- Possible Solutions At Different Iterations With Their Fitness Value

It. No	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 Value
1	3	8	5	8	5	9	1	6	4	6	1	7	4	2	7	2	2	4	5	10	24.8333
5	3	8	5	8	6	9	1	9	4	6	1	7	4	2	7	2	2	4	5	10	29.8333
10	3	8	2	9	1	3	1	6	2	6	1	7	4	2	8	5	7	4	5	10	31.6667
20	3	6	3	1	8	8	1	9	2	6	1	7	9	2	4	5	10	4	5	4	36.6667
30	3	6	10	1	8	8	1	9	2	6	1	7	9	2	4	5	3	4	5	7	42.0000
50	3	6	10	1	8	8	1	9	2	6	1	7	9	2	4	5	2	4	5	7	43.6667
75	3	6	10	1	8	8	1	9	2	6	1	7	9	2	4	5	2	4	5	7	43.6667
100	3	8	10	1	8	3	1	9	2	6	1	7	9	2	4	5	2	4	5	7	44.3333
120	3	8	10	1	6	8	1	9	2	6	1	7	9	2	4	5	2	4	5	7	45.1667
150	3	8	10	1	6	8	1	9	2	6	1	7	9	2	4	5	2	4	5	7	45.1667
175	3	8	10	1	6	8	1	9	2	6	1	7	9	2	4	5	2	4	5	7	45.1667
200	3	8	10	1	6	8	1	9	2	6	1	7	9	2	4	5	2	4	5	7	45.1667
225	3	8	10	1	6	8	1	9	2	6	1	7	9	2	4	5	2	4	5	7	45.1667
250	3	8	10	1	6	8	1	9	2	6	1	7	9	2	4	5	2	4	5	7	45.1667
275	3	8	10	1	6	8	1	9	2	6	1	7	9	2	4	5	2	4	5	7	45.1667
300	3	8	10	1	6	3	1	9	2	6	1	7	9	2	4	5	2	4	5	7	45.1667
325	3	8	10	1	6	8	1	9	2	6	1	7	9	2	4	5	2	4	5	7	45.1667
350	3	8	10	1	6	8	1	9	2	6	1	7	9	2	4	5	2	4	5	7	45.1667
400	3	8	10	1	6	8	1	9	2	6	1	7	9	2	4	5	2	4	5	7	45.1667
500	3	8	10	1	6	8	1	9	2	6	1	7	9	2	4	5	2	4	5	7	45.1667

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

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.

Let consider the best solution after first iteration

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

In the above example Subject Number 1 i.e., S1 is allocated to teacher number 3 i.e., T4, S2 is allocated again to T8, S3 is allocated to T5 and so on. The fitness function of above-given example is 24.8333.

Table 3 shows the final allocation of each subject to the corresponding teachers.

Table 3:- Subjects Allocations

Teacher Number	Subjects allocated to each teacher
T1	S4,S7,S11
T2	S14,S17
T3	S1,S9
T4	S15,S18
T5	S16,S19
T6	S5,S10
T7	S12,S20
T8	S2,S6
T9	S8,S13
T10	S3

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 graphical representation of Fitness value v/s iteration number curve is depicted in the fig 2.

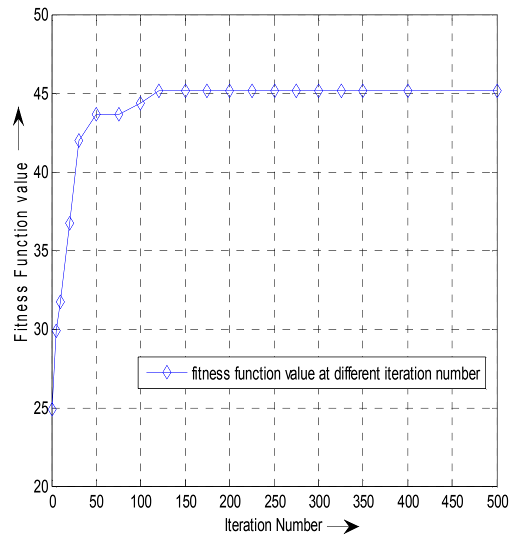

Figure 2:- Iteration Number v/s Fitness Function value graph

After obtaining the result from hybrid BCFA, the result was compared with the result obtained from Particle Swarm Optimization (PSO), Bee Colony Optimization (BCO) and Firefly Algorithm (FA) and a comparison graph was obtained which is described in fig 3.

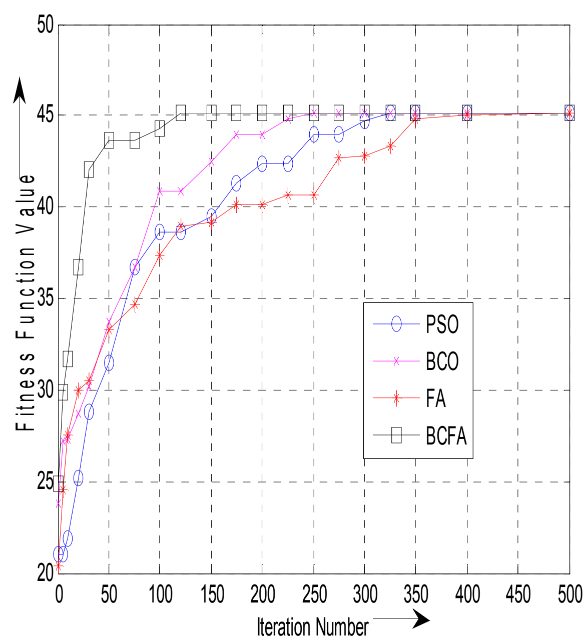

Figure 3:- Comparison of Fitness Value v/s Iteration Number curve of PSO, BCO, FA and Hybrid BCFA

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

6. DISCUSSION AND FUTURE SCOPE

The objective of the paper is to generate optimized course timetable within less time. This paper discusses three traditional searching techniques namely Particle Swarm Optimization(PSO), Bee Colony Optimization(BCO) and Firefly Algorithm (PSO) and a hybrid approach by combining the concept of BCO and FA namely BCFA hybrid approach to generate an optimized course timetable. After obtaining the result it was found that the hybrid approach BCFA produces much better result than the traditional three optimization techniques i.e., PSO, BCO and FA. The hybrid BCFA produces the result in much less time and takes less iteration to achieve optimization. The future scope is to optimize course timetable generation using other evolutionary Hybrid algorithms like hybrid approaches of BCO-PSO, FA-PSO, BCO-CSA (Cuckoo Search Algorithm) and many more.

7. CONCLUSION

Generally PSO, BCO and FA lead the way to generate an optimal solution by applying it in course time table problem. A hybrid firefly algorithm (BCFA) gives better result as compared to others algorithms. It is also used for solving the course time table and also laboratory timetable problem. This work also finds the optimal solution to design a course time table. BCFA are designed on the basis of timeslots which reduces the time complexity of the problem. This proposed problem improves the teachers' satisfaction and class schedule of teachers.

8. 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.,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, 2009,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]. Chu Shu-Chuan, Chen Yi-Tin, Ho Jiun-Huei, "Timetable Scheduling Using Particle Swarm Optimization", Proceedings of the First International Conference on Innovation Computing, Information and Control 2006, ICICIC.

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

[8]. Ho Sheau Fen Irene, Safaai Deris, Hashim Zaiton Siti Mohd, "University Course Timetable Planning using Hybrid Particle Swarm Optimization", Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation, 2009, ISBN: 978-1-60558-326-6.

[9] 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.

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

[11].Ojha.D,Sahoo.R.K.,Dash.S,"Automatic Generation Of Timetable Using Firefly Algorithm",International Journal of Advanced Research in Computer science and software engineering,Vol.6,Issue-4,2016,pp.589-593.

[12].Ojha.D,Sahoo.R.K.,Dash.S,"Automated Time table Generation Using Bee Colony Optimization",International Journal of Applied Information Systems,Vol.10-No.9,2016,pp.38-43.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

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

[14] Rajesh Kumar Sahoo,Deeptimanta Ojha, Durga Prasad Mohapatra,Manas Ranjan Patra,"Automated Test case Generation and optimization:A Comparative Review",International Journal of Computer Science & Information Technology,Vol.8,No.5,2016.

[15] Yeresime Suresh,Santanu Ku.Rath,"A genetic Algorithm based approach for test data generation in basis path Testing",International Journal of Soft computing and Software Engineering(JSCSE),Vol.3,No.3,2013.

[16] Rajesh Kumar Sahoo,Durga Prasad Mohapatra,Manas Ranjan Patra,"A firefly Algorithm Based Approach for Automated Generation and Optimization of Test cases",International Journal of Computer Sciences and Engineering,vol.4,Issue-8,2016,ISSN:2347-2693.

[17] Elizabeth Montero,Maria-Cristina Riff,Leopodlo Altamirano,"A PSO algorithm to solve a real course+Exam time tabling problem",ICSI(2011),pp.241-248.

[18] Mudjihartono Paulus, Triadi Gunawan Wahyu, Jin AI The, "University Timetabling Problems With Customizable Constraints Using Particle Swarm Optimization Method", International Proceeding: International Conference on Soft Computing, Intelligent System and Information Technology (ICSIIT), Petra University, Indonesia. ISBN: 978-602-97124-0-7, 2010.

[19]Herz, A, "Tabu search for large scale timetabling problems". *Europian Journal Operation Reearch* **1991**, *54*, pp.39–47.

[20]Mooney, E.L.; Rardin, R.L.; Parmenter, W.J. "Large scale classroom scheduling", *IIE Trans.* **1995**, 2*8*, pp. 369–378.