ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2016

Modified Genetic Algorithm to get near Optimal solution for Class Time Table Problem

Shafaque M Islam[#1], Uday Bhave[#2]

PG Student, Dept. of Computer Engg, Shah and Anchor Kutchhi Engineering College, Mumbai, India[#1]

Assistant Professor, Dept. of Computer Engg, Shah and Anchor Kutchhi Engineering College, Mumbai, India[#2]

ABSTRACT: Timetabling is the allocation of given resources to objects, subject to satisfy a set of desirable objectives as optimally as possible. The basic challenge of class time table is to schedule subject with respect to faculty over a limited time period so as to avoid conflicts and to satisfy a number of constraints. Scheduling even the smallest constraints can take a lot of time and it become worse when the number of constraints or the amount of data to deal with increases. Therefore the main focused of the author is on the generation of conflict free class time table by providing lab slots allocation from user .The system take various inputs like details of subjects, teachers available and all required constraints, depending upon these inputs it will generate a possible time table, making optimal utilization of resources. The genetic algorithm is one that seeks to find the most optimal solutions where the search space is great and conventional methods are inefficient. It works on a basis of the Darwinian evolution theory. The objective of the work is to create a model used to generate near optimal time table using genetic operators. Basic genetic algorithm is modified based on requirement and it is presented in this paper.

KEYWORDS: hard constraints; soft constraints; optimization; chromosome; genetic algorithm; crossover; mutation; diversity;

I. INTRODUCTION

A timetable is a system used to schedule time. A lecture timetable problem is concerned with finding the exact time allocation within limited time period of number of events (lectures-practicals) and assigning to them number of resources (teachers, rooms) while satisfying different constraints [1].

The time table problem has exponential number of the possible feasible timetables, even if the size of the problem input is not significantly large. This problem has therefore proven to be a very complex and time-consuming problem. Timetables are considered feasible provided the so-called hard constraints are respected. However, to obtain high-quality timetabling solutions, soft constraints (set of desirable conditions) must be full fill. It is a combinatorial optimization problem belonging to NP-hard class where the computational time grows exponentially as the number of variables increases. Although any given solution to the timetabling problem can be verified quickly, there is no known efficient way to locate a solution in the first place; indeed, the most notable characteristic of NP-complete problems is that no fast solution to them is known. That is, the time required to solve the problem using any currently known algorithm increases very quickly as the size of the problem grows.

GA differ from other search techniques in the following ways: [2]

- GAs are intrinsically parallel. Most other methods are serial and can only explore the solution space to a problem in one direction at a time, and if the solution they discover turns out to be suboptimal, there is nothing to do but abandon all work previously completed and start over. However, since GA has multiple offspring, they can explore the solution space in multiple directions at once.
- Due to the parallelism that allows them to implicitly evaluate many schemas at once, Gas are particularly well suited to solving problems where the space of all potential solutions is truly huge.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2016

Outline of this paper is : section 2 presents survey on literature . In section 3, we present our proposed system .section 4 discussed GA algorithm section 5 discussed the results and we give our concluding remark in section 6.

II. LITERATURE SURVEY

The advantage of GA is that they can explore the solution space in multiple directions at once. Therefore, if one path turns out to be a dead end, they can easily eliminate it and continue work on more promising avenues, giving them a greater chance each run of finding the optimal solution. If the hard constraints are breached the lecture timetable would not have a feasible solution. If the soft constraint is breached it does not necessarily affect the solution of the timetable, they are constraints that may be fulfilled or not fulfilled[3].

A timetable with binary representation is applied with several operators with the aim of preventing the violation of the fundamental constraints. The algorithm is guaranteed to always produce a feasible solution by satisfying the hard constraints. It utilizes one-point and two-point crossover operators and propagates distinctive timetable features to generate better solutions even for complex cases. It is found that GA influence can be affected by adjustment to its parameters [4].

The Parameters
Population Size = 50
Number of Generation = 150
One-point Crossover
Two-point Crossover
Adaptive Mutation
Tournament Selection
Crossover probability = 80%
Mutation rate = 20%

Table 1 [4]: The assigned parameters

III. PROPOSED SYSTEM

Scheduling timetable being tedious task for manual computation, the use of automated genetic algorithm can profoundly increase the efficiency of the process. Genetic algorithm (GAs) is non-deterministic and is used to solve mainly NP-hard problem like timetable scheduling problem [2].For solving timetabling problem, meta heuristic approach is needed that fits requirements which is possible by using fitness function and other operators provided by genetic algorithm. To get near optimal time table solution genetic algorithm is used in this project.

The timetable is constructed for engineering course. The course is divided into semesters (for fall semester: semester 1 & semester 3 and for spring semester: semester 2 & semester 4) and each semester has multiple divisions. There are 8 periods within a day and each period consists of 60 minutes. In every semester the course contains theoretical subjects and laboratory work also. In the course, faculty teaches multiple subjects in different division and semester. Practical period's assignment is taken by user. These practical periods remain static during time table generation. Based on fixed practical periods, lectures are allocated to faculty on available period.

Scope of this project is limited to following points:

- Time Table is generated for multiple divisions.
- Practical periods are taken as input.
- Soft constraints violation weight – age penalty value are taken as input
- Practical periods are considered as static during time table generation
- System generate clash free time table meeting all hard constraints.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2016

A. ***STRUCTURE OF THE AUTOMATED TIME TABLE GENERATOR***

Before a genetic algorithm can be put to work on any problem, a method is needed to encode potential solutions to that problem in a form that a computer can process. Data encoding is the first step before starting GA. It consists into getting a way to transform a solution into a chromosome. It is used to simplify the treatment and to improve speed of the algorithm.As shown in Fig 1: Chromosome is represented simply as a vector of periods (9.00 to 4.00), where each time table that represents solution is a number of genes and individual gene represent one subject with respective faculty. There is a vector of periods for each day for different classes.

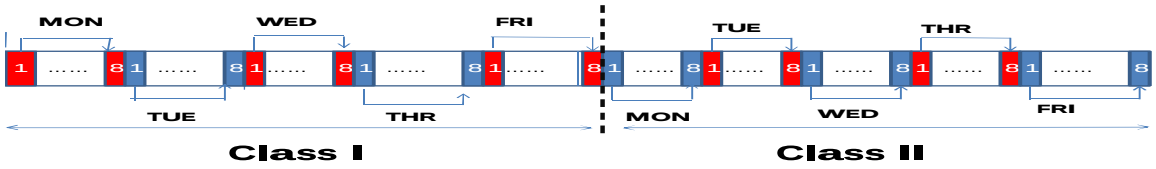


Fig 1. Data Represented in Vector form

B. ***CONSTRAINTS***

The constraints associated with lecture timetabling are dependent on how Engineering College wants timetable structured. The constraints involved in building the class time table for engineering college (through the study of constraints requirements of engineering colleges) are outlined below:

Hard Constraint:Hard constraints are those that must be fulfilled.

- Total number of Lectures of each Subject should be allotted as per curriculum.
- Same faculty should not be assigned to a same period in different class.
- No two faculties have lectures in the same period, in the same class.
- Empty periods should not be present on any day.

Soft Constraint:Soft constraints can be violated if necessary.

- Consecutive lectures of same faculty in the different class for same subject.
- No consecutive lectures of same faculty on different classes for different subjects.
- No consecutive lectures of different faculty in the same class for same subject.
- Not more than one lecture of same faculty on same subject on same day in same class.
- If there is a load at 9.00 am then there should not be any load beyond 4.00 pm.
- Maximum 2 hours lecture load per day but not more than 5 hours total load per day.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2016

IV. GENETIC ALGORITHM

Some important definitions:
• Gene: a Gene is a part of solution. It contains some data.
• Chromosome: a Chromosome (or individual) is a solution to the given problem, composed by genes.
• Population: set of solutions.
• Generation: a "round" of the algorithm, with a new population, after evolution.

A. *ALGORITHM OF THE PROPOSED TIME TABLE GENERATOR*

1. Generate initial population of generation 0 : pop_size

   Check each chromosome for Hard constraints satisfaction.

2. Calculate fitness of each chromosome

   If chromosome violating any soft constraints than penalty weight – age assign to it

3. Apply tournament selection on current population

   Select portion of current population :selected_pop

4. Apply reproduction operators on selected population

   4.1 apply crossover on 80% of selected population

   ie (selected_pop * 0.8)

   4.2 apply mutation on remaining population

   ie (selected_pop) – (selected_pop * 0.8)

5. Repeat step 2 to 4 while not generation _count>=50

B. *FITNESS EVALUATION*

The next step is the evaluation of fitness for each chromosome in the population. The fitness is calculated based on soft constraint violation of the class schedule; the score for fitness is thus attached to each chromosome for further processing of the main genetic operations (crossover and mutation).

How Fitness is Calculated

Weight-age is assigned to each soft constraint violation by the current solution based on their impact on organization. Following weight-age are given for already defined soft constraints based upon interview taken with time table committees of some colleges.
Weight-age is classified with following categories:

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2016

Category	Weight-age
VH – Very High	10 : w1
H- High	8 : w2
L – Low	6 : w3
VL – Very Low	4 : w4
Moderate	-4 : w5

- Step 1: initialize the chromosome fitness value to zero
- Step 2: check for No consecutive lectures of same faculty on different classes for different subjects. Increment fitness value by w1 if not.
- Step 3: check for No consecutive lectures of different faculty in the same class for same subject. Increment fitness value by w2 if not.
- Step 4: check for faculty load at 9.00 am then there should not be any load beyond 4.00 pm. Increment fitness value by w2 if not.
- Step 5: check for Maximum 2 hours lecture load per day but not more than 5 hours total load per day of faculty. Increment fitness value by w3 if not.
- Step 6: check for Not more than one lecture of same faculty on same subject on same day in same class. Increment fitness value by w4 if not.
- Step 7: check for Consecutive lectures of same faculty in the different class for same subject. Decrement fitness value by w5 if yes.

If soft constraints are violated than assigned weight-age are used to calculate fitness value. Solution which has high fitness value will be consider as bad solution and the solution which has low fitness value will be consider as good one.

C. ***SELECTION***

Selection is process of selecting individual for reproduction. This selection is done randomly with probability depending on the fitness of the individuals so that the best ones are often chosen for reproduction than poor ones.
[5] In tournament selection, n individuals are selected randomly from the larger population, and the selected individuals compete against each other. The individual with the highest fitness wins and will be included as one of the next generation population. Tournament selection also gives a chance to all individuals to be selected and thus it preserves diversity. The tournament selection has several advantages which include efficient time complexity, especially if implemented in parallel, low susceptibility to takeover by dominant individuals, and no requirement for fitness scaling or sorting.

```
Procedure :Tournament Selection
  While Population Size< (Pop_Size)/2 do
        Select two chromosome randomly from current generation : Pi, Pj
        Calculate fitness of selected chromosomes : Fi , Fj
        If Fi  <Fj
              Select  : Pi      else  Select :Pj
        End if
 End While
Return selected chromosomes for next generation
```

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2016

D. ***CROSS OVER***

Choosing two individuals to swap segments of their code, producing artificial "offspring" that are combinations of their parents. To generate new offspring, one point crossover is applied on selected parents.

```
Procedure : Crossover
        Crossover_size := 0.8 * selected population
        While  Not (Crossover_size)
                Select two chromosome randomly from selected popluation: tti ,ttj
                Select random day  :dayid
                L1:       Get period for first crossover point of dayid from tti : Pi
                          Get period for first crossover point of dayid from ttj :Pj
              If  period Pi isPractical or period PjisPractical then  goto  L1
              If  period Pi ==break then
                 L2:      Select two chromosome randomly from selected popluation: tti ,ttj
                           Select random day  :dayid
                           Get last two periods for second point crossover from tti :   P1i , P2i
                           Get last two periods for second point crossover from ttj :    P1j , P2j
                         If  any period P1i , P2i , P1j, P2j  isPractical then  goto  L2
                                  Else swap P1i,P1j and  P2i , P2j
                  Else   swap Pi and  Pj
                 End if
        End Loop
        Return chromosome after crossover for next generation
```

E. ***MUTATION***

Mutation operator randomly selects a section of the chromosome and then randomly regenerates all the probabilities from that section. The purpose of mutation in GAs is preserving and introducing diversity. Mutation should allow the algorithm to avoid local minima by preventing the population of chromosomes from becoming too similar to each other, thus slowing or even stopping evolution.

```
Procedure : Mutation
While Not Mutation_size
  For each selected chromosome
        Calculate fitness  : Fitness
        L1 : Select two periods that violates soft constraints : p1 , p2
        New Chromosome = swap (p1 , p2)
        Calculate fitness after mutation :TempFitness
        If TempFitness< Fitness //mutation improve fitness
                Fitness = TempFitness
                Return  New Chromosome for next generation
        Else
                Revert back swapping
                Got to  L1
        End if
End Loop
```

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

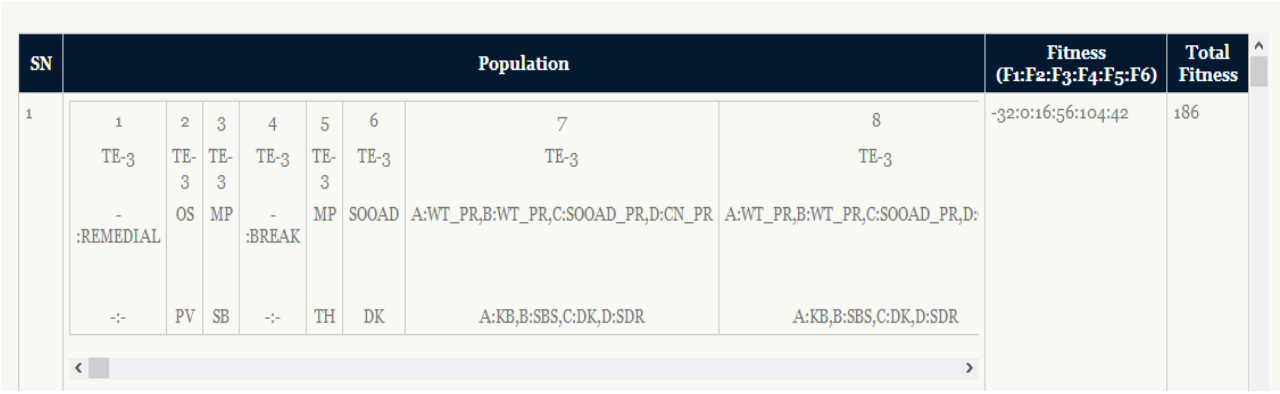
International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2016

V. RESULT ANALYSIS

Output screen of class time table generation :



SN	Population								Fitness (F1:F2:F3:F4:F5:F6)	Total Fitness
1	1	2	3	4	5	6	7	8	-32:0:16:56:104:42	186
	TE-3	TE-3	TE-3	TE-3	TE-3	TE-3	TE-3	TE-3		
	-:REMEDIAL	OS	MP	-:BREAK	MP	SOOAD	A:WT_PR,B:WT_PR,C:SOOAD_PR,D:CN_PR	A:WT_PR,B:WT_PR,C:SOOAD_PR,D:		
	-:-	PV	SB	-:-	TH	DK	A:KB,B:SBS,C:DK,D:SDR	A:KB,B:SBS,C:DK,D:SDR		

Fig 2 . Chromosome 1 and respective Fitness calculation

Figure 2 show solution of class time table and their respective fitness. As soon as any soft constraints are get violate, respective weight-age assign to solution to calculate fitness value.

Soft Constraint	Weight-age	Number of times violated	Fitness value
Soft Constraint 1	-4	8 times satisfied	-32
Soft Constraint 2	10	not violating	0
Soft Constraint 3	8	2 times	16
Soft Constraint 4	4	14 times	56
Soft Constraint 5	8	13 times	104
Soft Constraint 6	6	7 times	42

Table 1. Fitness Calculation of Choromosome1

Table 1.illustrate the fitness value of chromosome based on soft constraints violation weight-age assignment . So Total Fitness of chromosome 1 =186.

The experiments aim to analyse the result of the proposed Genetic algorithm with different parameters. Among chromosomes with fitness value, the chromosome with lowest fitness value would be the better one. Real data are used in these experiments. The experiments focus on the best fitness that could be obtained with fixed population size and generation iterations. The result of implementing the experiment is represented by the axes (X, Y). The X axis represents generation iterations and the Y axis represents lowest fitness value of the respective generation.

Experiment 1: The parameters are: population size = 50, number of generation = 20, One-point crossover, mutation, and tournament selection. Figure 2 shows the performance of the GA on Experiment 1.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2016

Gen No	Fitness	Gen No	Fitness
1	309.76	11	243.04
2	301.64	12	231.08
3	293.8	13	218
4	289.6	14	188.84
5	284.88	15	182
6	277.28	16	182
7	269.76	17	182
8	264.88	18	182
9	257.4	19	182
10	251.8	20	182

Table 2. Result For Experiment 1

Table 2, Illustrate the result of Experiment, class time table generator run 20 generation. Average Fitness is calculated for each generation.

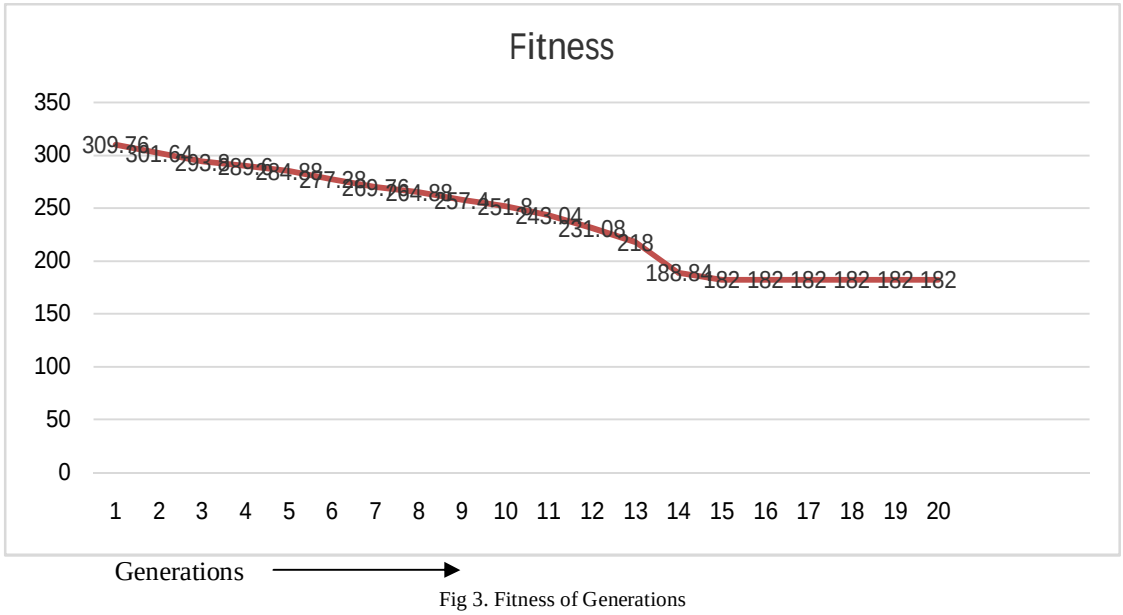


Fig 3. Fitness of Generations

As Shown in Fig 3 as number of generation increases, fitness value get reduces. At some point, fitness value get remain stable.

Experiment 2: It focused on soft constraints violation weight age value of chromosomes which has lowest fitness value of each generation iteration. Table 3 shows individual soft constraints violation weight-age of that chromosome which is most fittest (good solution) in the respective generation.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2016

Gen NO	SC1	SC2	SC3	SC4	SC5	SC6
1	-28	30	16	48	88	36
2	-28	30	16	48	88	36
3	-28	30	16	48	88	36
4	-28	30	16	48	88	36
5	-28	30	16	48	88	36
6	-28	30	16	48	80	36
7	-28	30	16	48	80	36
8	-28	30	16	48	80	36
9	-28	30	16	48	80	36
10	-28	30	16	48	80	36
11	-28	30	16	48	80	36
12	-28	30	16	48	80	36
13	-28	30	16	48	80	36
14	-28	30	16	48	80	36
15	-28	30	16	48	80	36
16	-28	30	16	48	80	36
17	-28	30	16	48	80	36
18	-28	30	16	48	80	36
19	-28	30	16	48	80	36
20	-28	30	16	48	80	36

Table3. Result For Experiment2

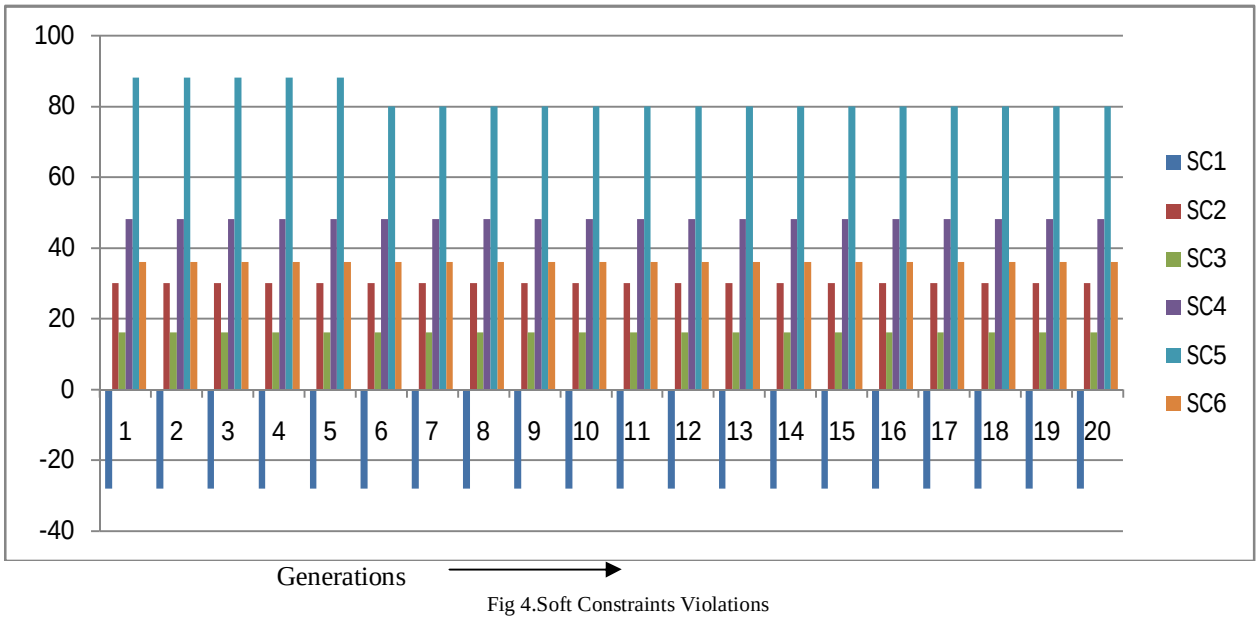


Fig 4.Soft Constraints Violations

Figure 4 shows graph based on data of Table 3. In each successive generation iteration, each soft constraints violation weight age value may get change. That result into change in the fitness value. The Table3, details the results of different generations and elucidates near optimal solution.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2016

VI. CONCLUSION

Genetic algorithm cannot be expected reliably to find optimum solutions, but it can yield excellent near optimal solutions, which are adequate for most practical problems. Basic genetic algorithm is modified based on requirements of the time table generation problem. The algorithm generates initial population only when all hard constraints have been satisfied. The fitness of algorithm is improved by applying selection and reproduction operations. Through tournament selection best fitted individuals selected for next generation. In reproduction, two different kind of crossover is applied to reduce violation of soft constraints. Mutation improves fitness of individual by swapping gens which violates soft constraints and result into good solution. The proposed genetic algorithm gives very good results related to manually generated time table.

REFERENCES

[1]. Ashish Jain1, Dr. Suresh Jain and Dr. P.K. Chande "Formulation of Genetic Algorithm to Generate Good Quality Course Timetable " International Journal of Innovation, Management and Technology" , Vol. 1, No. 3, August 2010.
[2]. PratibhaBajpai ,Dr.Manoj Kumar "An Approach to solve Global Optimization Problem" , Indian journal of computer science and engineering , vol 1.
[3] M. O. Odim, B. O. Oguntunde, O. O. Alli "On the Fitness Measure of Genetic Algorithm for Generating Institutional Lecture Timetable" Vol. 4, No. 4 April 2013.
[4] 1 Omar Ibrahim Obaid, 2 MohdSharifuddin Ahmad, 3 Salama A. Mostafa, 4 Mazin Abed Mohammed "Comparing Performance of Genetic Algorithm with Varying Crossover in Solving Examination Timetabling" , Journal of Emerging Trends in Computing and Information Sciences, VOL. 3, NO.10, Oct 2012.
[5]. NorainiMohdRazali John Geraghty, " Genetic Algorithm Performance with Different Selection Strategies in Solving TSP" Proceedings of the World Congress on Engineering 2011 Vol II WCE 2011, July 6 - 8, 2011, London, U.K.
[6]. AsifAnsari , Prof Sachin Bojewar," Genetic Algorithm to Generate the Automatic Time-Table – An Over View", International Journal on Recent and Innovation Trends in Computing and Communication , Volume: 2 Issue: 11 November 2014.
[7]. PratibhaPajpai, Dr.ManojKumar , "Genetic Algorithm - an Approach to Solve Global Optimization Problems",PratibhaBajpai et al./Indian Journal Of Computer Science and Engineering Vol 1 No 3 199-206.