IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-7, Issue-1, Jan 2019 E-ISSN: 2347-2693

Technique of Crossover and Mutation to solve School Time Table Problem using Genetic Algorithm

Mohd. Irfan

Department of Computer Science and Applications, DeshBhagat College, Bardwal-Dhuri, India

Available online at: www.ijcseonline.org

Accepted: 21/Jan/2019, Published: 31/Jan/2019

Abstract— Creating school time table lies under the category of NP hard problems. If we try to create a time table using exhaustive approach i.e applying all possible combinations, then it can take a huge amount of time to solve the problem. So Genetic Algorithm is a good way to solve such problems. Instead of using traditional techniques of Crossover and Mutation in genetic algorithm we have applied a better technique of crossover and mutation which reduces the search space by deducting invalid time tables from the search space. We have also done some experiments which show the variation of number of generations to solve the problem with mutation rate.

Keywords—NP hard problem**,**exhaustive approach

I. INTRODUCTION

Time Table problem is NP hard problem. It consumes too much time of teachers to create school time table. There may be too many constraints which are needed to be fulfilled. The constraints can be divided into two categories namely hard constraints and soft constraints. No teacher can be in two classes at same time, same room cannot be assigned to a class at one point of time are examples of hard constraints. A teacher should not be assigned more than three periods continuously, main subjects should be taught in the first three periods etc. are examples of hard constraints. The hard constraints need to be 100% fulfilled while soft constraints are tried to be fulfilled up to maximum level.

In this paper we are only focussed upon fulfilling the hard constraint i.e no teacher can be in two classes at same time.

If there are nine period, 10 classes & 9 subjects for each class then there are $(9!)^{10}$ ways to arrange subjects for which we need to search a very large search space, but still there would be a few time tables which fulfil our above mentioned constraint.

To avoid trying so many possibilities we can use Genetic Algorithm which uses heuristic search to find the solution. Genetic Algorithm uses concept of Natural Selection in which we create population of chromosomes using random arrangement of Genes. The best chromosomes crossover to create new chromosomes which may have better fitness. Sometimes chromosomes mutate which may result in a more fit chromosome.

In Time Table problem if we use traditional techniques to crossover and mutate then there may be chromosomes which are too unfit to be valid. In figure: 1 after crossover and mutation, invalid time table may be generated, because same subject (S11) is scheduled to teach twice and subject S34 which is of class 3 is scheduled in class 1 and S12 a subject of class 1 is scheduled in class 3.Here j^{th} subject of i^{th} class is indicated as Sij.To avoid such invalid time tables we are using a technique of crossover and mutation. In this paper we have discussed the Technique.

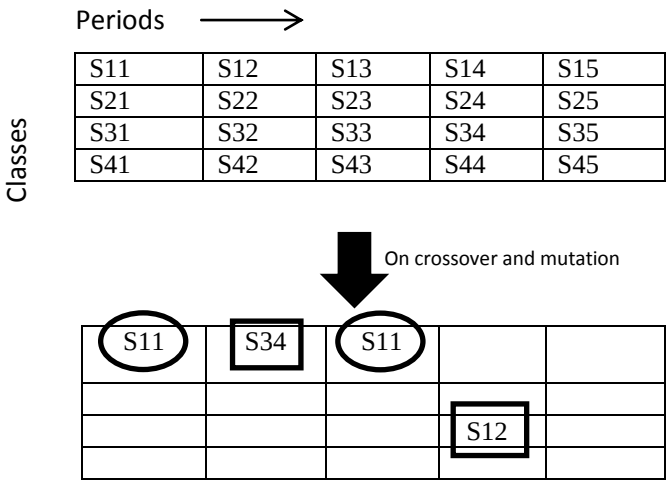

Figure:1

II. METHODOLOGY

While applying Genetic algorithm, we assume time table as a chromosome, a pair of a subject and a teacher where both are tied together as a gene. The fitness evaluation, selection, crossover and mutation is done according to the flow chart as shown in figure: 2.

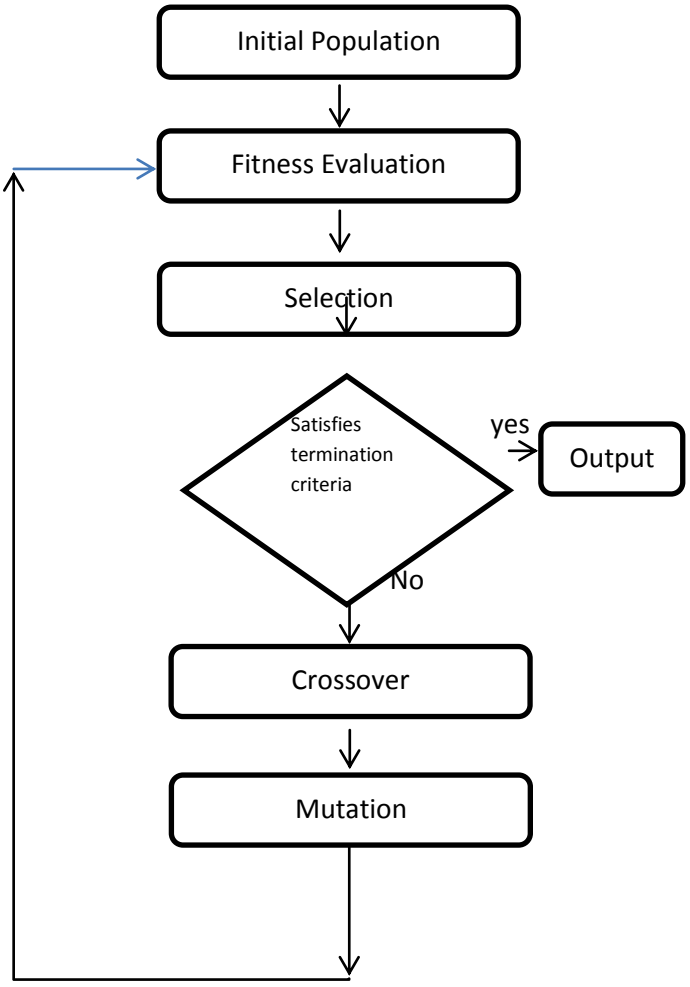

Figure: 2

In our Technique of Crossover, we use horizontal cut on a random point on time table as shown in figure: 3.

S11	S12	S13	S14	S15

Horizontal cut →

S21	S22	S23	S24	S25
S31	S32	S33	S34	S35
S41	S42	S43	S44	S45

Figure: 3

In our Technique of mutation we swap two subjects of the same class. Firstly, a class is selected randomly, and then two subjects of that class are selected at random and swapped as shown in figure:4.

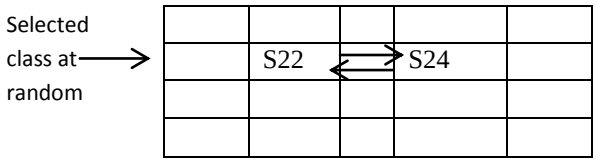

Figure: 4

III. EXPERIMENTAL SETUP

We have taken 9 periods, 10 classes and 9 subjects for each class for the purpose of our experimentation and all the 9x10=90 subjects are tried to divide evenly among 10 teachers. Using the methodology mentioned above we have applied some experiments. We have used 10 Chromosomes for initial population, one point random crossover and different mutation rates ranging from 10% to 100%. We have applied genetic algorithm as described in the methodology.

IV. RESULTS AND DISCUSSION

We have got the results of the experiment shown in table:1 for different mutation rates.

Table: 1

Mutation rate	Average number of generations to solve the Time Table
10%	7159
20%	3610
30%	2473
40%	1853
50%	1485
60%	1197
70%	1066
80%	952
90%	839
100%	767

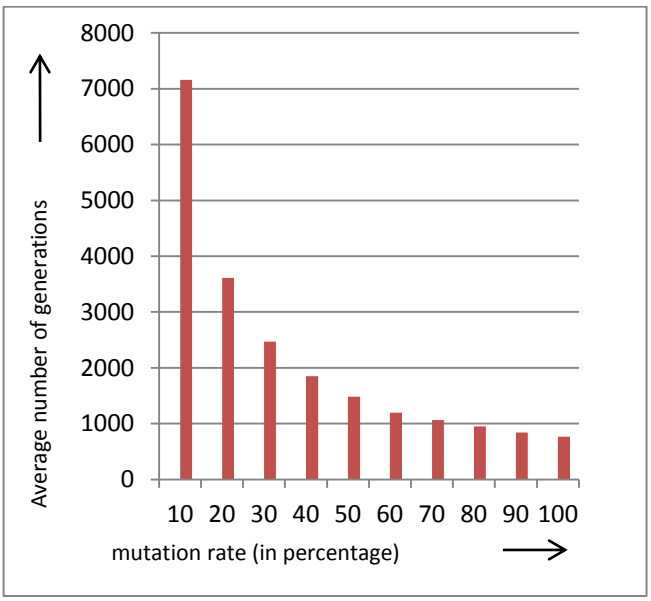

Figure: 5

From the data in Table:1 and Figure: 5 it is found that as we increase mutation rate we get better results i.e our problem gets solved in lesser number of generations. The graph as shown in Figure:6 shows the number of generations for experiments done 1000 times at 100% mutation rate.

It is also found that the time table is solved in average time of 12.987 milliseconds when 1000 experiments at 100% mutation rate are done on a PC having 4GB Ram, Intel i-5 processor and Windows 10 operating system.

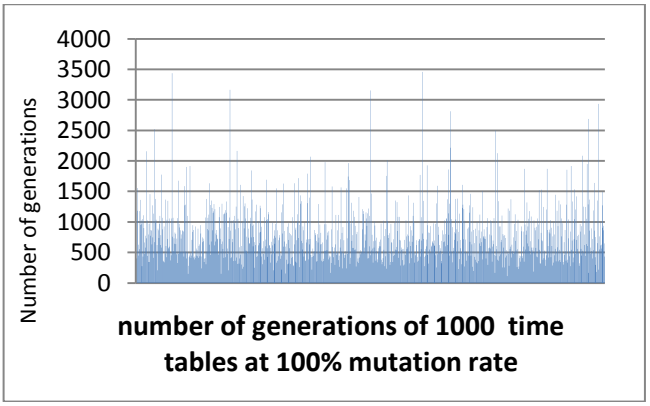

Figure: 6

V. Conclusion and Future Scope

We have observed from the experiments that using our technique of crossover and mutation, solution is found in a few generations which take a very small amount of time. We have also observed that as we increase mutation rate we get better results. But we cannot go beyond 100% mutation rate.

Further to enhance the work we can apply mutation by selecting multiple rows or classes which may solve the problem in lesser number of generations.

VI. References

[1] Man, K. F., Tang, K. S., &Kwong, S. (1996). *Genetic algorithms: concepts and applications [in engineering design]. IEEE Transactions on Industrial Electronics, 43(5), 519–534*.doi:10.1109/41.538609

[2] E. Canth-Paz, "*A summary of research on parallel genetic algorithms*," Illinois Genetic Algorithms Lab., Univ. Illinois at Urbana-Champaign, IlliGAL Rep. 95007, July 1995.

[3] Manoj Garg and Dinesh Kumar, "*Simple GA & Hybrid GA for Basis Path Testing under BDFF*", International Journal of Scientific Research in Computer Science and Engineering, Vol.4, Issue.6, pp.28-35, 2016

[4] S. Dubey, R. Jhaggar, R. Verma, D. Gaur, "*Encryption and Decryption of Data by Genetic Algorithm*", International Journal of Scientific Research in Computer Science and Engineering, Vol.5, Issue.3, pp.47-52, 2017

[5] https://en.wikipedia.org/wiki/Genetic_algorithm

Author Profile

Mohd. Irfan completed his M.Sc. I.T Degree from Punjabi University, Patiala (Punjab) in 2016.He is working as an Assistant Professor in Department of Computer Science and Applications at Desh Bhagat College, Bardwal-Dhuri(Punjab)