ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 2, February 2015

Automatic Timetable Generation using Genetic Algorithm

Dipesh Mittal[1], Hiral Doshi[2], Mohammed Sunasra[3], Renuka Nagpure[4]

Bachelors of Engineering, Dept., of Information Tech, Atharva college of Engineering, University of Mumbai, India[1,2,3]

Assistant Professor, Dept., of Information Technology, Atharva college of Engineering, University of Mumbai, India[4]

Abstract: Timetable creation is a very arduous and time consuming task. To create timetable it takes lots of patience and man hours. Time table is created for various purposes like to organize lectures in school and colleges, to create timing charts for train and bus schedule and many more. To create timetable it requires lots of time and man power .In our paper we have tried to reduce these difficulties of generating timetable by Genetics Algorithm. By using Genetic algorithm we are able to reduce the time require to generate time table and generate a timetable which is more accurate, precise and free of human errors. The first phase contains all the common compulsory classes of the institute, which are scheduled by a central team. The second phase contains the individual departmental classes. Presently this timetable is prepared manually, by manipulating those of earlier years, with the only aim of producing a feasible timetable.

Keywords: Genetic algorithm, timetable, constraints, chromosomes.

I. INTRODUCTION

The class timetabling problem is a typical scheduling problem that appears to be a tedious job in every academic institute once or twice a year [3]. In earlier days, time table scheduling was done manually with a single person or some group involved in task of scheduling it manually, which takes a lot of effort and time. Planning timetables is one of the most complex and error-prone applications.

Timetabling is the task of creating a timetable while satisfying some constraints. There are basically two types of constraints, soft constraints and hard constraints. Soft constraints are those if we violate them in scheduling, the output is still valid, but hard constraints are those which if we violate them; the timetable is no longer valid [1].

The search space of a timetabling problem is too vast, many solutions exist in the search space and few of them are not feasible. Feasible solutions here mean those which do not violate hard constraints and as well try to satisfy soft constraints. We need to choose the most appropriate one from feasible solutions. Most appropriate ones here mean those which do not violate soft constraints to a greater extent [1]. Using Genetics Algorithm, a number of trade-off solutions, in terms of multiple objectives of the problem, could be obtained very easily. Moreover, each of the obtained solutions has been found much better than a manually prepared solution which is in use.

II. LITERATURE SURVEY

Genetic algorithms are general search and optimization algorithms inspired by processes and normally associated with natural world. Genetic algorithm mimics the process of natural selection and can be used as a technique for solving complex optimization problems which have large spaces [10]. They can be used as techniques for solving complex problems and for searching of large problem spaces. Unlike many heuristic schemes, which have only one optimal solution at any time, Genetic algorithms maintain many individual solutions in the form of population. Individuals (parents) are chosen from the population and are then mated to form a new individual (child). The child is further mutated to introduce diversity into the population [10].Rather than starting from a single point within the search space, GAisinitialized to the population of guesses. These are usually random and will bespread throughout the search space. A typical algorithm then uses three operators, selection, crossover and mutation, to direct the population toward convergence at global optimum.A GA, as shown in figure 1 requires a processof initializing, breeding, mutating, choosing andkilling. It can be said that most methods called GAs have at least the following elements in common: Population of chromosomes, Selection according to fitness, Crossover to produce new offspring, and random mutation of new offspring.

```
Create a population of creatures.
Evaluate the fitness of each creature.
While the population is not fit enough:
{
Kill all relatively unfit creatures.
While population size< max;
{
Select two population members.
Combine their genetic material to create a new creature.
Cause a few random mutations on the new creature.
Evaluate the new creature and place it in the population.
}}.
```

Fig 1: Top Level description of a GA [6]

A. GA Operators

1) *Chromosome representation:*Chromosome is a set of parameters which define a proposed solution to the problem that the genetic algorithm is trying to solve. The chromosome is often represented as a simple string. The fitness of a chromosome depends upon how well that chromosome solves the problem at hand.

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 2, February 2015

2) *Initial population:*The first step in the functioning of a GA is the generation of an initial population. Each member of this population encodes a possible solution to a problem. After creating the initial population, each individual is evaluated and assigned a fitness value according to the fitness function. It has been recognized that if the initial population to the GA is good, then the algorithm has a better possibility of finding a good solution and that, if the initial supply of building blocks is not large enough or good enough, then it would be difficult for the algorithm to find a good solution.

3) *Selection:*This operator selects chromosomes in the population for reproduction. The fitter the chromosome, the more times it is likely to be selected to reproduce [11].

4) *Crossover:*In genetic algorithms, crossover is a genetic operator used to vary the programming of a chromosome or chromosomes from one generation to the next. It is analogous to reproduction and biological crossover, upon which genetic algorithms are based. Cross over is a process of taking more than one parent solutions and producing a child solution from them. There are methods for selection of the chromosomes. This operator randomly chooses a locus and exchanges the subsequences before and after that locus between two chromosomes to create two offspring. For example, the strings 10000100 and 11111111 could be crossed over after the third locus in each to produce the two offspring 10011111 and 11100100. The crossover operator roughly mimics biological recombination between two single−chromosome organisms [11].

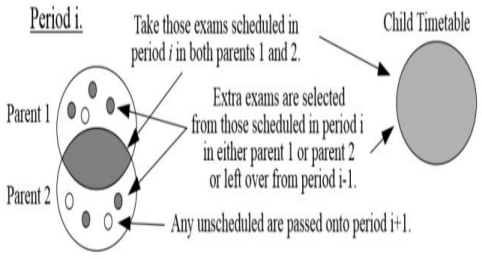

Fig 2: Crossover Operator [9]

5) *Mutation:*Mutation is agenetic operator used to maintain genetic diversity from one generation of a population of genetic algorithm chromosomes to the next. It is analogous to biological mutation. Mutation alters one or more gene values in a chromosome from its initial state. In mutation, the solution may change entirely from the previous solution. Hence GA can come to better solution by using mutation. This operator randomly flips some of the bits in a chromosome. For example, the string 00000100 might be mutated in its second position to yield 01000100. Mutation can occur at each bit position in a string with some probability, usually very small [11].

6) *Fitness Function:*The fitness function is defined over the genetic representation and measures the quality of the represented solution. The fitness function is always problem dependent In particular, in the fields of genetic programming and genetic algorithms, each design solution is commonly represented as a string of numbers referred to as a chromosome. After each round of testing, or simulation, the idea is to delete the 'n' worst design solutions, and to breed 'n' new ones from the best design solutions. Each design solution, therefore, needs to be awarded a figure of merit, to indicate how close it came to meeting the overall specification, and this is generated by applying the fitness function to the test, or simulation, results obtained from that solution.

III. PROPOSED APPROACH

In order to deal with timetabling issues we are proposing a system which would mechanically generate timetable for the institute. Course and lectures will be scheduled in accordance with all possible constraints and given inputs and thus a timetable will be generated.

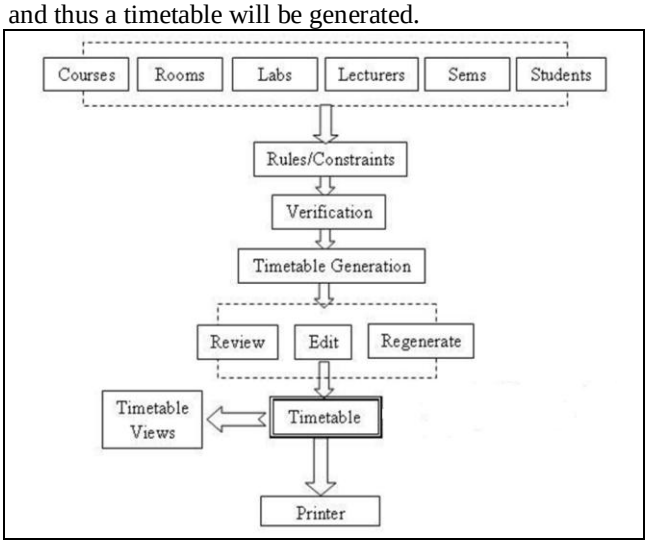

Fig 3: General view of Timetable Generator [4]

Structure of time table generator consists of input data, relation between the input data, system constraints and application of genetics algorithm.

A. Input Data

The input data contains:

1) *Professor:*Data describes the name of lecturers along with their identification number.

2) *Subject:* Data describes the name of courses in the current term.

3) *Room:*Data describes the room number and their capacity.

4) Time intervals: It indicates starting time along with duration of a lecture.

B. System Constraints

System constraints are divided into 2 categories:

1) *Hard Constraints:*The timetable is subjected to the following four types of hard constraints, whichmust be satisfied by a solution to be considered as a valid one:

a. A student should have only one class at a Time.

b. A Teacher should have only one class at a time.

c. A room should be booked only for one class at a time.

d. Some classes require classes to have particular equipment. For example, audio visual equipment, projectors etc.

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 2, February 2015

*2) Soft Constraints:*These are the constraints that are of no great concern but are still taken into contemplation. They don't need to be satisfied but the solutions are generally considered to be good if they are satisfied.

a. Courses must be eventually distributed.

b. Students should not have any free time between two classes on a day.

c. Scheduling of teachers should be well spread over the week.

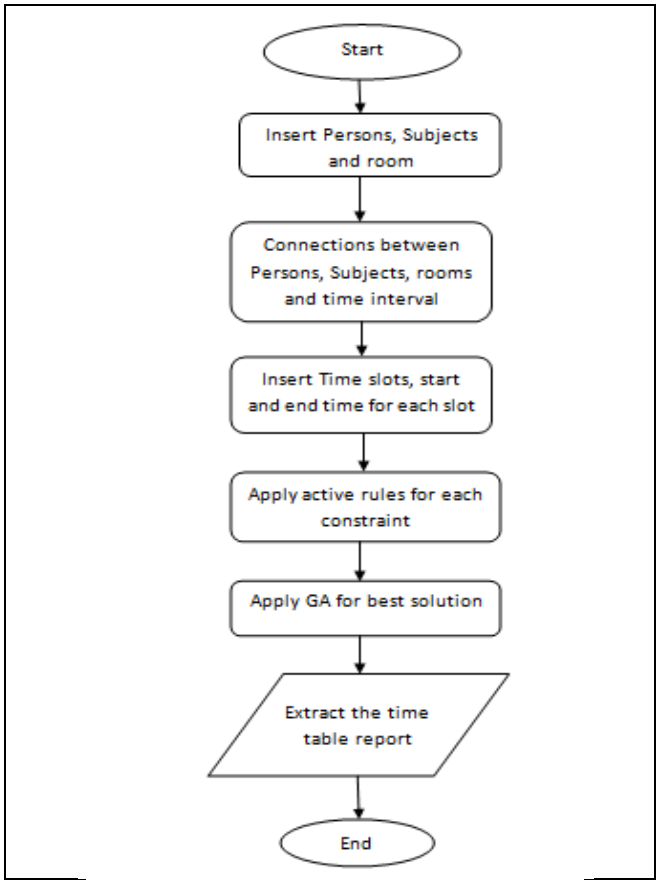

Fig 4: The structure of time table generator [7]

IV. FUTURE SCOPE

To generate timetable for the institute which will be less time consuming and free of human errors along with high level of efficiency and precision. Moreover improve the overall process of timetable generation with help of genetics algorithm along with the assistance of technology.

V. CONCLUSION

As discussed, an evolutionary algorithm, genetics algorithm for time tabling has been proposed. The intention of the algorithm to generate a time-table schedule automatically is satisfied. The algorithm incorporates a number of techniques, aimed to improve the efficiency of the search operation. By automating this process with the help of computer assistance timetable generator can save a lot of precious time of administrators who are involved in creating and managing various timetables of the institutes. Also the timetables generated are much more accurate, precise than the ones created manually.

We have used C# along with .NET framework to develop our application. We have used real data of various departments of our institute to test the method and how effectively it is functioning.

The project reduces time consumption and the pain in framing the timetable manually. The benefits of this approach are simplified design and reduced development time.

ACKNOWLEDGEMENT

We are thankful to Prof. NeelimaPathak, Prof. SumitaChandak and Prof. RenukaNagpure, Department of Information Technology, Atharva college of Engineeirng.

REFERENCES

[1] M. Doulaty, M. R. FeiziDerakhshi, and M. Abdi, "Timetabling: A State-of-the-Art Evolutionary Approach", *International Journal of Machine Learning and Computing*,Vol. 3, No. 3, June 2013.

[2] Anirudha Nanda, Manisha P. Pai, and AbhijeetGole, "An Algorithm to Automatically Generate Schedule for School Lectures Using a Heuristic Approach*", International Journal of Machine Learning and Computing*, Vol. 2, No. 4, August 2012

[3] DilipDatta, Kalyanmoy Deb, Carlos M. Fonseca, "Solving Class Timetabling Problem of IIT Kanpur using Multi-Objective Evolutionary Algorithm".*KanGAL* 2005.

[4] AnujaChowdhary, PriyankaKakde, ShrutiDhoke, SonaliIngle,RupalRushiya, Dinesh Gawande, 'Time table Generation System", *International Journal of Computer Science and Mobile Computing*, Vol.3 Issue.2, February- 2014.

[5] MughdaKishorPatil, RakheShrutiSubodh, Prachi Ashok Pawar, NaveenaNarendrasinghTurkar, "Web Application for Automatic Time Table Generation", *International Journal of current Engineering and Technology*, E-ISSN 2277-4106, P-ISSN 2347-5161.

[6] Sandeep Singh Rawat, Lakshmi Rajamani, "A Time table Prediction for Technical Educational System using Genetic Algorithm", *Journal of Theoretical and Applied Information Technology*,2005-2010JATIT.

[7] BharkaNarang, Ambika Gupta, RashmiBansal, "Use of Active Rules and Genetic Algorithm to Generate the Automatic Time-Table", *International Journal of Advances in Engineering Sciences*,Vol.3(3).

[8] Alberto Colorni, Marco Dorigo, Vittorio Maniezzo, "A Genetic Algorithm to Solve the Time Table Problem", *Computonal Optimization and Applications Journal*.

[9] Rupert Weare, Edmund Burke and Dave Elliman, "A Hybrid Algorithm for Highly Constrained Timetabling Problem", *Computer Science Technical Report* No. NOTTCS-TR-1995-8.

[10] D. Abramson, J.Abela,"A Parallel Genetic Algorithm for Solving the School Timetabling Problem", *15 Australian Computer Science Conference, Hobart*, Feb 1992.

[11] Melanie Mitchell, "An Introduction To Genetic Algorithm", *A Bradford Book The MIT Press*, Fifth printing 1999.

[12] David A.Coley, "An Introduction To Genetic Algorithms For Scientists And Engineers",World*Scientific Publishing Co. Pvt. Ltd*.,1999.

BIOGRAPHIES

Dipesh Mittal, currently in the final year of Bachelors of engineering in Information Technology from Atharva college of Engineering which is affiliated to University of Mumbai. Will be pursuing Masters in Information Management with concentration in Business Intelligence in the coming future.

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 2, February 2015

HiralDoshi, currently in the final year of Bachelors of engineering in Information Technology from Atharva college of Engineering which is affiliated to University of Mumbai. Will be pursuing Masters in Computer Science in the coming future.

Mohammed Sunasra, currently in the final year of Bachelors of engineering in Information Technology from Atharva college of Engineering which is affiliated to University of Mumbai. Will be pursuing Masters in Computer science in coming future.

RenukaNagpure, currently working as Assistant Professor in the department of Information Technology in Atharva college of Engineering which is affiliated to University of Mumbai. Have done her masters in Computer Science. Also have published several papers such as one in national conference and six in International Journals.

DOI 10.17148/IJARCCE.2015.4254