AUTOMATIC TIMETABLE GENERATION USING GENETIC ALGORITHM

K. Sangeeta, S. Sravya, P. Hari Krishna, V. Krishna, S. V. Pavan Kalyan

Department of CSE,

Aditya Institute of Technology and Management, Srikakulam, Andhra Pradesh, India.

ABSTRACT

The generation of timetables has always been tedious right from time and except being tedious, the timetable created has always been full of series of errors and mistakes. Such an oversized amount of techniques are recommended to solving this problem. During this paper, genetic Algorithm was employed by creating a gaggle of your time series randomly from a given time and courses in other to hunt out an answer to the timetable problems. The courses thus formed are evaluated with the assistance of the evaluation function. Soul logs into the system then the administrator input the courses with their codes and also the unit. At that time, the admin will keep adding until the amount of courses needed has been inputted. The admin can remove a course that has been inputted within the case of error. After inputting the courses, it moves to the subsequent page where all the lecture halls or rooms which are able to be used are inputted. After inputting these, the system then generates the timetable system. This method (genetic algorithm) used helps in reducing to barest minimum, errors and mistakes in encountered in developing an automatic timetable.

Keywords: Genetic algorithm, timetable, constraints, chromosomes.

INTRODUCTION

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

Timetabling is that the task of constructing a timetable while satisfying some constraints. There are basically two varieties of constraints, soft constraints and hard constraints. Soft constraints are those if we violate them in scheduling, the output continues to be valid, but hard constraints are those which if we violate them; the timetable is not any further valid.

The search space of a timetabling problem is solely too vast, many solutions exist within the search space and few of them are not feasible. Feasible solutions here mean those which don't violate hard constraints and further attempt to satisfy soft constraints. We would wish to decide on the foremost appropriate one from feasible solutions. Most appropriate ones here mean those which don't violate soft constraints to a greater extent. Using Genetics Algorithm, form of trade-off solutions, in terms of multiple objectives of the matter, might be obtained very easily. Moreover, each of the obtained solutions has been found far better than a manually prepared solution which is in use.

Genetic Algorithm Operators

1) Chromosome representation:
A chromosome representation is critical to clarify each individual within the GA population. The representation scheme determines how the matter is structured within the GA and also determines the genetic operators that are used. Each chromosome is formed from a sequence of genes from a predefined alphabet. One useful representation of chromosome for function optimization involves genes from an alphabet of floating point numbers with values limited by an upper and a boundary. The fitness of a chromosome depends upon how well that chromosome solves the matter at hand.

2) Initial population:
The process begins with a bunch of individuals which is called a Population. Each individual is additionally a solution to the matter you'd wish to resolve.
An individual is characterized by a bunch of parameters (variables) said as Genes. Genes are joined into a string to form a Chromosome (solution).
In a genetic algorithm, the set of genes of a personal is represented employing a string, in terms of an alphabet. Usually, binary values are used (string of 1s and 0s). We are saying that we encode the genes during a chromosome.

3) Selection:
The idea of selection phase is to select the fittest individuals and permit them to pass their genes to the following generation.
Two pairs of individuals (parents) are selected supported their fitness scores. Individuals with high fitness have more chance to be selected for replica.

4) Crossover:
Crossover is that the foremost important introduce a genetic algorithm. For each pair of parents to be mated, a crossover point is chosen willy-nilly from within the genes.
Offspring are created by exchanging the genes of parents among themselves until the crossover point is reached then
The new offspring are added to the population.

5) Mutation:
In certain new offspring formed, variety of their genes is also subjected to a mutation with an occasional random probability. This implies that variety of the bits within the bit string is also flipped.
Mutation is a genetic operator accustomed maintain genetic diversity from one generation of a population of genetic algorithm chromosomes to the subsequent. It is analogous to biological mutation. Mutation alters one or more gene values in an exceedingly chromosome from its initial state. In mutation, the solution may change entirely from the previous solution. Hence GA can come to raised solution by using mutation. This operator randomly flips variety of the bits in an exceedingly chromosome .For example the string 111000 is also mutated in its second position to yield 110110. Mutation can occur at each bit position in an exceedingly string with some probability, usually very small.

6) Fitness Function:
The fitness function determines how fit a personal is (the ability of a personal to compete with other individuals). It gives a fitness score to each individual. The probability that a personal are selected for replica relies on its fitness score.
The fitness function is usually problem dependent particularly, within the fields of genetic programming and genetic algorithms, each design solution is sometimes represented as a string of numbers remarked as a chromosome. After each round of testing, or simulation, the thought 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 point how close it came to meeting the specification, and this is often generated by applying the fitness function to the test, or simulation, results obtained from that solution.
Fitness is given by

Fitnessvalue/population fitness x no. of children.

LITERATURE SURVEY

Usually timetable is scheduled manually in schools, colleges and in universities [1,2,3,5,6,8,10,11]. Scheduling timetable manually is hectic and time consuming. As timetable is made by the human being there's also possibilities of error within the timetable.

To overcome these problems people usually taking the previous years' schedule and modifying it but still it is a tedious job to incorporate changes. To overcome all these problems we propose to make an automated system.

Genetic Algorithm is a Meta heuristic one. It guarantees in obtaining a solution and the resultant obtained is more optimal. In Heuristic method the chances of getting optimal solutions are less but in case of Genetic Algorithm the chances are more.

The system will take various inputs like details of students, subjects and class rooms and teacher's availability, load distribution, lab availability depending upon these inputs it will generate a possible schedule, making optimal utilization of all resources in a way that will best suit any of constraints or college policies. List of subjects may include electives as well as core subjects. The case is similar to colleges and other educational institutions. So our aim is to develop a general purpose which can efficiently generate optimal solutions.

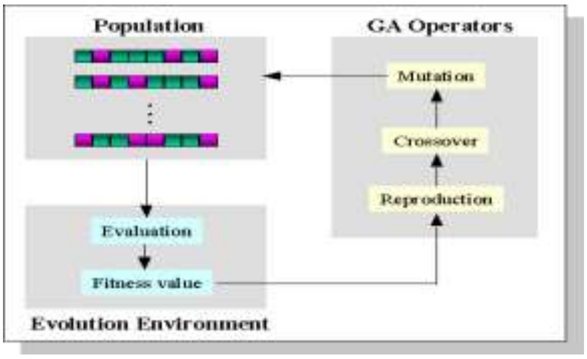
Observations about existing system are:
1. Generating Timetable manually is extremely susceptible to human errors.
2. Stress on human to satisfy all the constraints.
3. Problem in satisfying time critical constraints.
Hard constraints involved in existing system [4]:
1. No participant is also in additional than two rooms at the identical period.
2. No room should be double booked.
3. The space capacity should be large enough to hold each.

Evolutionary Algorithms (EAs) are a class of direct, probabilistic search and optimization algorithms gleaned from the model of organic evolution [13]. A Genetic Algorithm (GA) is a type of EA and is regarded as being the most widely known EA in recent times [12].
A GA differs from other search techniques in the following ways:
GAs optimizes the trade-o between exploring new points in the search space and exploiting the information discovered thus far.
GAs has the property of implicit parallelism. Implicit parallelism means that the GAs effect is equivalent to an extensive search of hyper planes of the given space, without directly testing all hyper plane values. Each schema denotes a hyper plane.

GAs are randomized algorithms, in that they use operators whose results are governed by probability. The results for such operations are based on the value of a random number. This means GAs use probabilistic transition rules, not deterministic rules.
GAs operates on several solutions simultaneously, gathering information from current search points to a direct subsequent search. Their ability to maintain multiple solutions concurrently makes them less susceptible to the convergence problem of local maxima and noise.
GAs work with a coding of the parameter set, not the parameters themselves. GAs search from a population of points, not a single point.
GAs use (objective function) information, not derivatives or other auxiliary knowledge. It is demonstrated that the literature is currently converging on the use of constraint based solution algorithms and implementations. It is also noted that the next most commonly reported implementation involves the use of hybrid algorithms [7].

Proposed System

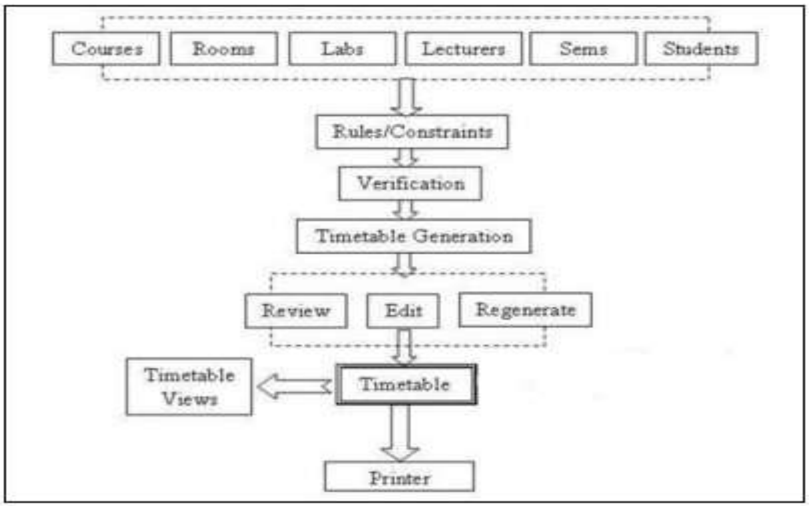
Genetic algorithm mimics the tactic of survival and may be used as a technique for solving complex optimization problems which have large spaces. Rather than starting from one point within the search space, GA is initialized to the population of guesses. These are usually random and might be spread throughout the search space.



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

1) Professor: Data describes the name of lecturers along with their identification.
2) Subject: Data describes the name of courses within this term.
3) Room: Data describes the space number and their capacity.

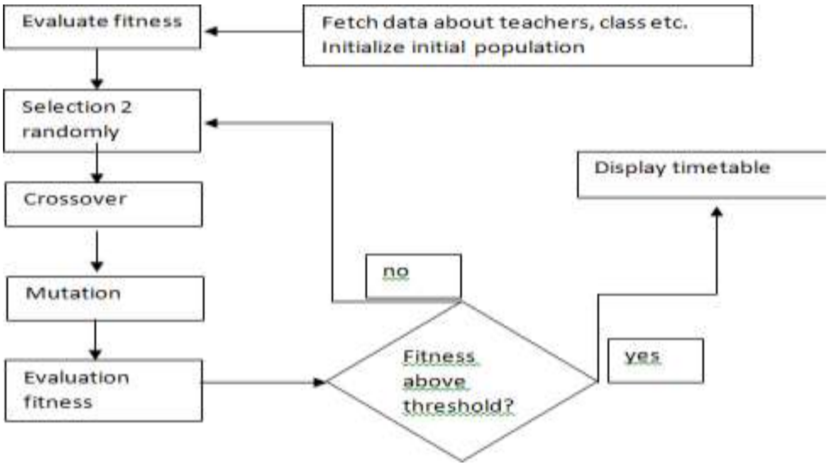
4) Time intervals: It indicates get-go along with duration of a lecture.



System Constraints:
System constraints [4] are divided into 2 categories:

1) Hard Constraints: The timetable is subjected to the following four types of hard constraints, which must be satisfied by a solution to be considered as a sound one:
a. A student should have just one class at a Time.
b. a lecturer should have just one class at a time.
c. a locality should be booked only for one class at a time.
d. Some classes require classes to have particular equipment. as an example, audio visual equipment, projectors etc.
These are usually random and might be spread throughout the search space.

2) Soft Constraints: These are the constraints that are of no great concern but are still taken into contemplation. They are doing not must be satisfied but the solutions are generally considered to be good if they're satisfied.
a. Courses must be eventually distributed.
b. Students mustn't have any free time between two classes on daily.



RESULTS AND FUTURE SCOPE

To generate timetable for our institute which can be less time consuming and freed from human errors together with high level of efficiency and precision. The timetable generated is dynamic which is fit for the students to attend the classes according to their wish.

To generate timetable for our institute which can be less time consuming and freed from human errors together with high level of efficiency and precision. Moreover improve the process of timetable generation with help of genetics algorithm together with the help of technology [9].

CONCLUSION

The timetables generated using the algorithm are far more neutral and efficient than those made by a human. The mechanical approach saves many some time and is far more convenient than the normal methods.

There are fewer errors in such timetables. Though a manually made timetable could also be more flexible than the machine generated but this approach saves many some time.

Thus, a combination of the algorithm along with manual approach is that the most effective. A timetable generated using the algorithm could also be tweaked by a human to make slight changes that may be preferred.

Acknowledgment:
We are thankful to Sri. K. Sangeeta, Department of Computer Science and Engineering, Aditya Institute of Technology And Management.

REFERENCES

[1] Automated Class Scheduling Generation in the Context of a University Schedule Information System, Kuldeep Singh Sandhu, School of Management, Nathan Campus, Griffith University, 21st September 2001.

[2] A Genetic Algorithm Based University Scheduling System, Edmund Burke, David Elliman and Rupert Weare, University Park, University of Nottingham, NG7 2RD 2007 Publisher IEEE.

[3] An Evolutionary Algorithm for solving School Time-tabling problem, Calogero Di Stefano Genetica S.r.l. Via San Dionigi 15,I-20139 Milan, Italy and Andrea G. B. Tettamanzi – Universita degli Studi di Milano Depatimento di Tecnologie dell Informazione Via Bramante 65, I-26013 Crema, Italy.

[4] Constraint Based Scheduling, A.M. Abbas - Department of Computer Science of Balamand Tripoli, Lebanon, E.P.K Tsang – University of Essex Colchestor, CO4 3SQ, England.

[5] Generating Complete University Course Schedules by Using Local Search Methods, Duong Tuan Anh, Vo Hoang Tam, and Nguyen Quoc Viet Hung, 2006 International Conference on research, Innovation and Vision for the Future–IEEE publisher.

[6] OPUS-College Schedule Module Design Document A. Cornelissen, M.J. Sprengers, B. Mader Radbound University Nijmegen, Nijmegen, July 6, 2010.

[7] Construction of Schedules Based on Periodic Event Scheduling, Jonas ChristoferVillumsen, IMM – Thesis-2006-52, Publisher: Informatics and Mathematical Modelling, Technical University of Denmark, 2006.

[8] Application to a University Course Scheduling Problem by General Project, Scheduler Masanori Horio, Atsuo Suzuki, The 7th International Symposium on Operations Research and its applications (ISORA'08), Lijiang, China, October 31- November 3, 2008.

[9] Dynamic Schedule tracker, Paula Okunie , Transit IDEA Project, March 2009, Transportation Research Board of the National Academics.

[10] Abramson, D. (1991). Constructing Schools Timetables Using Simulated Annealing: Sequential and Parallel Algorithms Management Science.

[11] Adejuwon, O. S. (2012). Development of a University timetable automation system.

[12] Anisha, J., Ganapathy, S. C., Harshita, G., & Rishabh, B. (2015). A Literature Review on Timetable generation algorithms based on Genetic Algorithm and Heuristic approach . International Journal of Advanced Research in COmputer and Communication Engineering , 159-163.

[13] Back, T. (1995). Evolutionary Algorithms in theory and practice.