International Journal of Research & Development in Technology and Management Science –Kailash
Volume - 22| Issue 6 | 2016 | ISBN - 1-63102-468-X
European Article Number [EAN] - 978-1-63-102468-9
editor.ijrdtm@rtmonline.in | http://journal.rtmonline.in

Paper Id: **IJRDTM –0530150**

TIME TABLE GENERATION USING GENETIC ALGORITHM

by

Akanksha Bansal Chopra | Shyama Prasad Mukherji College (University of Delhi), New Delhi **|** akankshabansal.asm@gmail.com

Deepak Chopra | Nagarro Software Pvt. Ltd., Gurgaon **|** deepakchopra83@gmail.com

Manvi Gupta | Shyama Prasad Mukherji College (University of Delhi), New Delhi **|** gptmanvi03@gmail.com

ABSTRACT

Preparing time table manually in institutions with large number of students is a very tedious job which consumes a lot of time resulting in various clashes such as one teacher having more than one class at a time and room clash etc. Clashes are the consequence of the human errors which are difficult to prevent in the following process- If we use previous years' timetable and try to modify it by incorporating changes as per the requirements. To overcome the above listed problem, in this paper, an automated system is proposed.

KEYWORDS : Genetics, Cross-Over, Mutation, Selection, Fitness.

INTRODUCTION

Time Table generation software provides all the operations for proper management in an organization like colleges. It allows staff of the college to generate the time table for the organization. Human requires scheduling of time table in order to manage the time effectively. It is used for management in schools, colleges and other fields of teaching and working like crash courses, coaching centers, training programs. In olden days, time table scheduling was done manually involving single person or some group in task of scheduling it with their hands, which took lot of effort and time. While scheduling even the smallest constraints can take a lot of time and when there are n- number of constraints or the amount of data to deal with increases. In such cases previously designed time table is reused for whole generation without any changes, proving to be the same in such situations. Other cases that can cause problem is when the number of employers are less, which results in rescheduling of time table or they need to fill empty seats urgently.

In our work, the system will take various inputs like :Course subject, Department name, Name of teacher, Paper number, Number of sections per year, Number of days a teacher can take lectures, and Number of lectures a teacher can take in a day. Depending upon these inputs it will generate a best possible time table, making optimal use of all resources so it adhere to the given constraints or college rules.

Institutions/Colleges/Universities use such time tables regularly. They schedule their course to meet the need of current duration and facilities available to them. The requirement of new course addition and newly enrolled students to fresh batches should meet their schedule. The software focuses on serving the staff working in the organization. It makes their work easier as the software removes the clashes between the lectures and time slots. It allows them to keep the record of the lectures, add new lectures in the existing time slots. It has everything that a staff need to manage the time table, hence makes it a perfect software which makes the work easy and makes time table software easily manageable.

RELATED WORKS

The authors have discussed the purpose of generation of time table in various institutions like schools and colleges, timing charts for train and bus schedule [1]. The techniques of cross-over, mutation and the fitness function for efficient utilization of infrastructure are used and a model in generated using probabilistic operators.[2] Implemented genetic algorithm for solving time table issues and generation of exam time table using real student data from engineering department courses.[3]Authors in another paper has used "event-condition-action" model for the implementation and has analyzed the higher fitness function for the generation of automatic time table.[4]

GENETIC ALGORITHM

A genetic algorithm is a search technique used to find true or approximate solutions for various search problems. Genetic algorithms fall under the category of global search heuristics. It uses the concepts of "Natural Selection" and "Genetic Inheritance" given by Darwin in 1859.It was originally developed by John Holland in 1975.Genetic algorithm are a sub class of an evolutionary algorithms that use techniques invented by evolutionary biology such as inheritance, mutation, selection, and crossover (also called recombination).

A.BASICS OF GENETIC ALGORITHM

The initiation of a population of randomly generated individuals usually starts the evolution.The fitness of every individual in the population is evaluated in each generation, multiple number of individuals are selected from the current population based on their fitness, then they are recombined and mutated to form a new generation of a population.The newly generated population is then used in the next iteration of the algorithm.When either a maximum number of generations has been produced or a satisfactory fitness level has been reached for the population, then the algorithm is terminated.If the algorithm terminates due to a maximum number of generations, an optimal solution may or may not have been reached.

B.IMPLEMENTATION OF GENETIC ALGORITHM IS AS FOLLOWS

Implemented as a computer simulation in which the initial variables (parent variables) are combined by implementing algorithm techniques(cross-over) to form new generations (children variable) to give the best optimal solution.

C. NECESSITIES

A genetic illustration of the solution domain, and a fitness perform to gauge the solution domain. A standard illustration of solution is as an array of bits. Arrays of different sorts and structures are employed in basically an equivalent manner. The major property that creates these genetic representations convenient is that their elements are simply aligned as a result of their fastened size, that facilitates easy crossover operation.

Variable length representations can also be used, however crossover implementation is additional complicated during this case. Tree-like representations area unit explored in Genetic programming. Chromosomes could be:

- Bit strings (0101 ... 1100)
- Real numbers (43.2 -33.1 ... 0.0 89.2)
- Permutations of element (E11 E3 E7 ... E1 E15)
- Lists of rules (R1 R2 R3 ... R22 R23)
- Program elements (genetic programming)

D.FITNESS FUNCTION

The fitness function is outlined over the genetic illustration and measures the standard of the pictured solution. The fitness function is usually depends on problem. For instance, within the knapsack problem we wish to maximize the entire price of objects that we are able to place in a knapsack of some fastened capability. An illustration of solution may be an array of bits, where every bit represents a unique object, and also the price of the bit (0 or 1) represents whether or not or not the article is within the knapsack.

Not each such illustration is valid, because the size of objects might exceed the capability of the knapsack. The fitness of the solution is that the total of values of all objects within the knapsack if the illustration is valid, or zero otherwise. It's onerous or maybe not possible to outline the fitness expression; in these cases, interactive genetic algorithms are used.

E.VARIOUS CATEGORIES

E.1.INITIALIZATION

Initially several individual solutions are generated randomly to associate to an initial population. The population size depends on the character of the matter however generally contains thousands of potential solutions.

Traditionally, the population is generated randomly, covering the whole vary of potential solutions (the search space).

E.2.SELECTION

During every sequent generation, a proportion of the present population is chosen to breed a new generation. Individual solutions area unit designated through a fitness-based method, wherever fitter solutions (as measured by a fitness function) area unit generally to be designated. Certain selection ways rate the fitness of every solution and preferentially choose the most effective solutions. Different ways rate solely a random sample of the population, as this method is also terribly long. Most functions are stochastic and designed in order that a little propotion of less work solutions are designated. This helps keep the range of the population massive, preventing premature convergence on poor solutions. In style and well-studied choice ways embody roullette choice & tournament choice.

E.3.REPRODUCTION

The next step is to come up with a second generation population of solutions from those picked through genetic operators: crossover (also referred to as recombination), and/or mutation. For each new solution to be created, a combination of "parent" solutions is chosen for breeding from the pool selected antecedent.

By producing a "child" solution victimization on top of strategies of crossover and mutation, a new solution is made which usually shares several of the characteristics of its "parents". New parents are selected for every kid, and the method continues till a new population of solutions of applicable size is generated.

E.4.CROSSOVER

In crossover, you select a locus that you swap the remaining alleles from one parent to the is can be advanced and is best understood visually. As you'll be able to see, the children take one section of the chromosome from every parent. The point at that the chromosome is broken depends on the indiscriminately elite crossover point.

This explicit methodology is termed single point crossover as a result of just one crossover point exists. Generally sole child one or child two is formed, however often each offspring are created and placed into the new population. Crossover doesn't forever occur, however. Sometimes, no crossover happens and therefore the parents are derived on to the new population. The probability of crossover occurring is sometimes 60% to 70%.

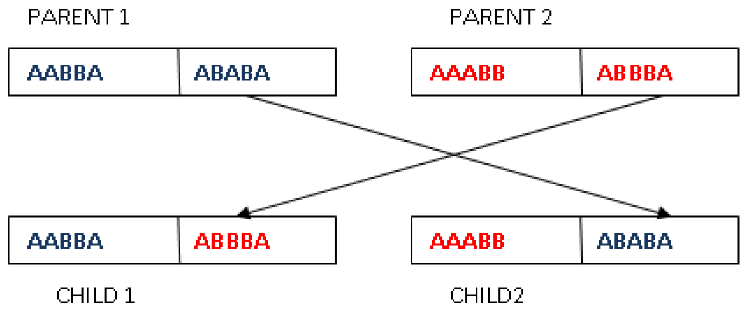

Figure1:CROSS-OVER

E.5.MUTATION

After selection and crossover, you currently have a brand new population full of people. Some are directly traced and other are made by crossover. In order to confirm that the people aren't all precisely the same, you permit for a little probability of mutation. You loop through all the alleles of all the people, and if that allele is chosen for mutation, you'll be able to either amend it by a little quantity or replace it with a brand new cost. The likelihood of mutation is typically between one and 2 tenths of a percent. Mutation is fairly easy ,you only amend the chosen alleles supported what you are feeling is critical and move on Mutation is, however, very important to make sure genetic diversity at intervals of the population.

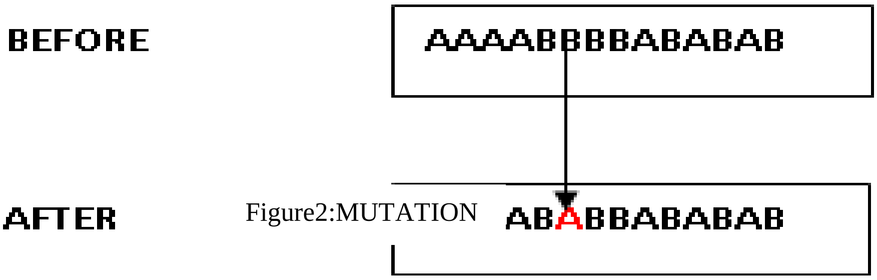

Figure2:MUTATION

IMPLEMENTATION OF GENETIC ALORITHM IN TIME TABLE

- Genetic algorithm removes clashes between the time slots .
- It uses the techniques of cross over which helps to generate the time table.
- It ensures that no teacher and student have more than one lecture at the same time.

CALCULATIONS

The calculations for the implementation of time table using genetic algorithm are done as follows:

- ✓ Number of teachers in department be:n;
- ✓ Number of days a teacher can take lecture be:x;
- ✓ Number of lectures in a day a teacher can take be:y;
- ✓ Now ,workload can be calculated on the basis of the above defined variables.

$$\boxed{\text{Workload of a week}(w)=n*x*y;}$$

WORKFLOW

The given workflow depicts the relationship with database and shows the flow of inputs.

Paper Id: **IJRDTM –0530150**

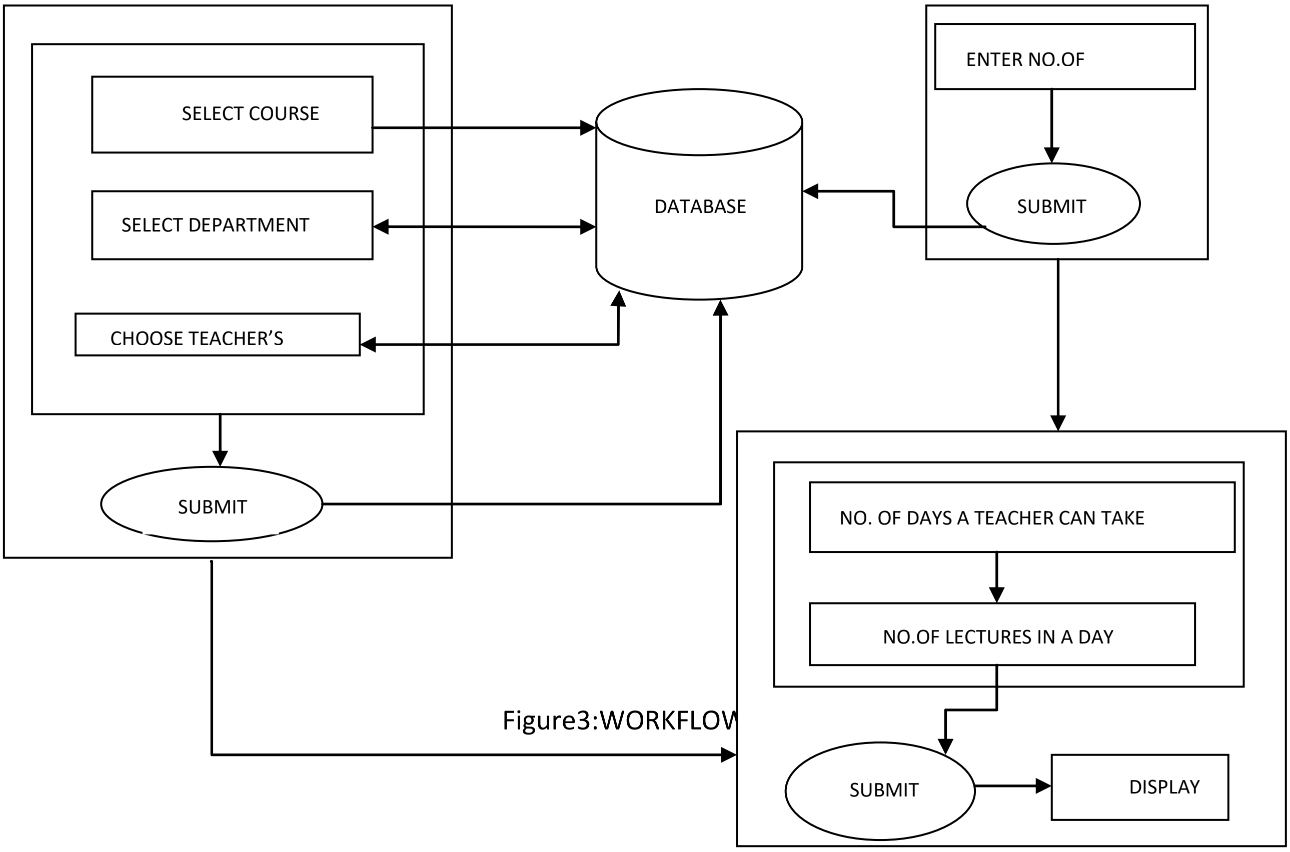

Figure3:WORKFLOW

DRAMATIC REPRESENTATION TO IMPLEMENT THE TIME TABLE

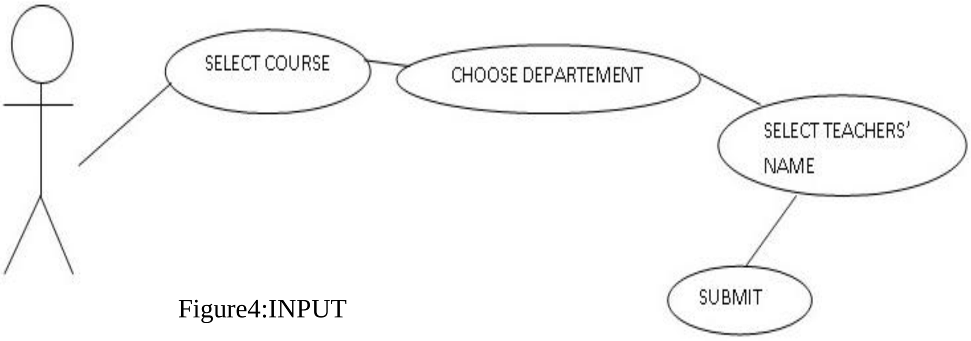

Figure4:INPUT

In the above given figure the user selects the course then chooses the department after that respective teachers' names will be displayed of the concerned department, lastly select the name of the teacher then click on submit button.

Paper Id: **IJRDTM –0530150**

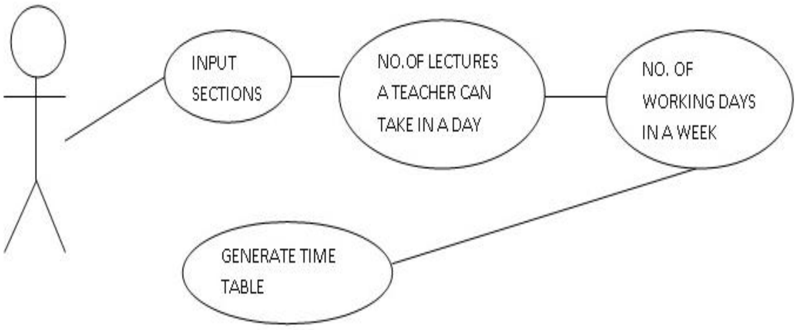
INPUT SECTIONS
NO.OF LECTURES A TEACHER CAN TAKE IN A DAY
NO. OF WORKING DAYS IN A WEEK
GENERATE TIME TABLE

Figure5:DISPLAY

After clicking on submit button user inputs the respective sections, enters the number of lectures a teacher can take in a day, number of working days in a week and finally he/she clicks on generate button and the final time table is generated.

CONCLUSION

A modified genetic algorithms based technique was developed to come up with a conflict free, optimal and effective lecture timetable with relation to all the thought of metrics. The framework of the modified genetic algorithms method may be classified into 2 modules: generation of associate initial diary based mostly on lecturer, course, cluster and timeslots.

The second module will deal with the issues and clashes of theory and practical lectures and appropriate techniques of genetic algorithm will be used to solve it thus providing an error-free software for implementation in the college. Generations of feasible and optimal lecture timetable was the basic feature of the software.

FUTURE WORK

The designed software generates the time table by removing the clashes between the time-slots for theory lectures only. In accordance with the designed software, incorporation of practical lectures in the software is suggested.

REFERENCES

[1]. Mittal, Dipesh., Doshi ,Hiral. , Sunasra, Mohammed., Nagpure ,Renuka., 2015. "Automatic Timetable Generation using Genetic Algorithm", International Journal of Advanced Research in Computer and Communication Engineering,4(2), pp. 245-248. (Source:www.ijarcce.com/upload/2015/february-15/IJARCCE4I.pdf).

[2]. Ansari, Asif., Prof .Bojewar ,Sachin.,2014." Genetic Algorithm to Generate the Automatic Time-Table – An Over View", International Journal on Recent and Innovation Trends in Computing and Communication,2(11),pp.3480-3483.(Source: http://www.ijritcc.org/download/1416628378.pdf).

[3]. Jha Kumar ,Sujit.,2014." EXAM TIMETABLING PROBLEM USING GENETIC ALGORITHM", International Journal of Research in Engineering and Technology, pp. eISSN: 2319-1163 | pISSN: 23217308.(Source:http://esatjournals.net /ijret/2014v03/i05/IJRET20140305120.pdf).

[4]. Narang,Bharkha.,Gupta,Ambika.,Bansal,Rashmi.,2013." Use of Active Rules and Genetic Algorithm to Generate the Automatic Time-Table", Proceedings of 2nd International Conference on Emerging Trends in Engineering and Management. (Source:http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.685.9109&rep=rep1&type=pdf).