ISSN 2395-695X (Print)
ISSN 2395-695X (Online)
International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)
Vol. 2, Special Issue 15, March 2016

TIME TABLE AUTOMATION SYSTEM USING BACKTRACKING TECHNIQUE

S.ALAGU LAKSHMI
N.DURGA
IT/FINAL YEAR
SRIVIDYA COLLEGE OF ENGINEERING
AND TECHNOLOGY
VIRUDHUNAGAR

Ms.S.SUJITHA,M.E.,
Assistant Professor/CSE&IT
SRIVIDYA COLLEGE OF
ENGINEERING ANG
TECHNOLOGY
VIRUDHUNAGAR

ABSTRACT- An information system based upon the need to support timetable production and maintenance is presented. Given the very practical outcomes expected of timetable research, the information system was designed to enable the whole range of administrative functions performed by teachers to be either directly supported or readily modified to prove such support. The implementation of this particular system is given and resulting timetables are presented and discussed. The system generated manual and automated timetables and these were produced by trailing a number of objective functions. It was noted that the determination of the optimal objective function is dominated by specific individual institutional criteria. It is suggested that this would make a more than significant project for future information systems research. 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 Backtracking Technique 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.

INTRODUCTION

The general area of scheduling has been the subject of intense research for a number of decades. Scheduling and timetabling are typically viewed as two separate activities, with the term scheduling used as a generic term to cover specific types of problems in this area. Consequently, timetable construction can be considered as a special case of generic scheduling activity. In the most general terms, scheduling can be described as the constrained allocation, of resources to objects, being placed in space-time in such a way as to minimize the total cost of a set of the resources used. Time table construction is the allocation, subject to constraints, of given resources to objects being placed in space-time in such a way as to satisfy or nearly satisfy a desirable set of possible.

This paper describes a constraint-based model and algorithm for a train timetable and route generation program. This program determines how trains can be efficiently dispatched from a depot to meet rush hour service requirements while satisfying resource constraints. Our research focused mainly on generating a timetable and routing for the building up of train services. However, it can also be applied to the reverse case of breaking down the service after rush hour ends. The research was performed using data from one of the world's busiest subway systems. The constraint-based algorithm described in this paper was tested on one of the busiest lines within this subway system. The subway authority currently uses a rule-based expert system to generate the train timetable based on headway requirements, i.e., the time between the arrivals of two consecutive trains. The routing is then generated using a separate semi-automatic system which is based on heuristics. This routing program determines how each train should travel to get to its destination. Subway trains normally travel in a cyclic manner from one final terminus to other and back. The key problem is what happens when the train reaches the final terminus and needs to reverse while at the same time trains are being dispatched from the depot to the terminus. The timetable and route generation is currently performed by separate systems. The timetable generation system must therefore ensure there is enough slack or buffer for the routing system to be able to generate the routes. This extra time buffer causes inefficienct..

EXISTING SYSTEM

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. 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. Rather than starting from a single point within the search space, GA is initialized to thepopulation of guesses. These are usually random and willbespread 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.

PROPOSED SYSTEM

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. Structure of time table generator consists of input data, relation between the input data, system constraints and application of Backtracking Technique . 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, which must 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. 2) Soft Constraints: These are the constraints that are of no great concern but are still taken into contemplation.

MODULES

- Dataset Upload
- Preprocessing and updating db
- Backtracking Technique
- Generation of timetable for every table

Dataset upload

A dataset (or data set) is a collection of data, usually presented in tabular or text form. When in the form of a table, each column represents a particular variable and each row corresponds to a given member of the dataset in question. A dataset is used to document information about the creation of instance records, and to link external resources to them (like the linkage and structure schema). A dataset can be seen as an aggregation of instance records used to keep a reference between the instance records and their source (provenance). A dataset can be split into multiple dataset slices. Each slice can be written in a separate file. Each slice of a dataset shares the same <id> of the dataset. Datasets normally contain one or more data records from a single source representing the same type of instance(s). However, the flexibility of a dataset can accommodate any other less-usual use cases. Datasets may reside on the Web as well as be stored locally. Each dataset is uniquely identified with standard metadata characterizations. At minimum, datasets have a simple structure of attribute-value pairs for each instance record. However, they may also have more complex structure via schemas (ontologies) that also describe the relationships between concepts and attributes and may even relate those to external schema. We, upload the dataset which contains details related to the table such as faculties name , subject code and semester it divide as odd semester and even semester. Depending upon the content in the dataset we generate the table.

Preprocessing and updating db

- **In preprocessing step , we choose the selected dataset.**
- **Then applying preprocessing technique to remove unwanted data the dataset.**
- **Finally , we uploaded the error free dataset into the Database.**

Backtracking Technique

Backtracking is a general algorithm for finding all (or some) solutions to some computational problems, notably constraint satisfaction problems that incrementally builds candidates to the solutions, and abandons each partial candidate c ("backtracks") as soon as it determines that c cannot possibly be completed to a valid solution. The classic textbook example of the use of backtracking is the eight queen's puzzle that asks for all arrangements of eight chess queens on a standard chessboard so that no queen attacks any other. In the common backtracking approach, the partial candidates are arrangements of k queens in the first k rows of the board, all in different rows and columns. Any partial solution that contains two mutually attacking queens can be abandoned. Backtracking can be applied only for problems which admit the concept of a "partial candidate solution" and a relatively quick test of whether it can possibly be completed to a valid solution. It is useless, for example, for locating a given value in an unordered table. When it is applicable, however, backtracking is often much faster than brute force enumeration of all complete candidates, since it can eliminate a large number of candidates with a single test. Backtracking is an important tool for solving constraint satisfaction problems, such as crosswords, verbal arithmetic, Sudoku, and many other puzzles. It is often the most convenient (if not the most efficient) technique for parsing, for the knapsack problem and other combinatorial optimization problems. It is also the basis of the so-called logic programming languages such as Icon, Planner and Prolog. Backtracking depends on user-given "black box procedures" that define the problem to be solved, the nature of the partial candidates, and how they are extended into complete candidates. It is therefore a Meta heuristic rather than a specific algorithm – although, unlike many other meta-heuristics, it is guaranteed to find all solutions to a finite problem in a bounded amount of time. The term "backtrack" was coined by American mathematician D. H. Lehmer in the 1950s. The pioneer string-processing language SNOBOL

(1962) may have been the first to provide a built-in general backtracking facility.

Generation of timetable for every table

By applying Backtracking Technique , we generate the timetable for each class with condition of each subject should not appear at the same time for same year.We checking each condition for over all subjects and finally we generate timetable for separately.

CONCLUSION

As discussed, an evolutionary algorithm, back tracking 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.The next step in back tracking after encoding is to initiate the population.It is about creating an amount of individuals using hard constrains randomly. Small amount population can give a bad output as the amount of populationcan get smaller after the invalid chromosomes produced by the evolution get killed. In the other hand, largepopulation gives better results but the system will get slower as more chromosomes need to go through back tracking process.

REFERENCE

[1] Mei Rue Computer and Automation Engineering (ICCAE), 2010 The 2nd International Conference on (Volume:4)

[2] Bhaduri a "university timetable scheduling using genetic algorithm". Advances in Recent Technologies in Communication and Computing, 2009. ARTCom '09. International Conference

[3] Dipti Shrinivasan "automated time table generation using multiple context reasoning for university modules" Published in: evolutionary computation, 2002. cec '02. proceedings of the 2002 congress on (volume:2)

[4]Simon Robinson, Christian Nagel, Jay Glynn, Morgan Skinner, Karli Watson, Bill Evjen "Professional C#", 3rd Edition, Wiley Publishing,Inc 2004

[5] Silberschatz, H. Korth and S. Sudarshan, "Database System Concepts", 4rth Edition, McGraw-Hill International, 2002.