International Journal of Engineering Research-Online
A Peer Reviewed International Journal
Articles available online http://www.ijoer.in

Vol.3., Issue.5., 2015
(Sept.-Oct.)

RESEARCH ARTICLE

ISSN: 2321-7758

FREETAM
Automated Timetable Planner

SAMEER DESWAL[1], SAVITA BISHNOI[2]

[1]Computer Science & Engineering,Rohtak Institute of Technology & Management ,Rohtak, Haryana, India,

[2]Assistant Professor, CSE Department, Rohtak Institute of Technology & Management, Rohtak, Haryana, India

ABSTRACT

Creating timetable for any institution is quite a complex task because it includes various constraints to consider like suitable teacher should be present in a class at proper time. Freetam will reduce the effort or time frame of generation of timetable and it will generate timetable by including constraints.

INTRODUCTION

Timetable construction is NP-complete problem and it is part of constraint satisfaction problem. Constraint programming is a natural tool for describing as well as solving a lot of problems from various areas. Its major advantage is its capability of precise declarative description of a problem using relations between variables. It is based on a strong theoretical basis and it has wide practical applications in areas of evaluation, modeling, and optimization.

Timetabling is one of the typical examples of constraint programming application. The task is to allocate activities in time and space respecting various constraints and to satisfy as nearly as possible a set of desirable objectives. A typical constraint is the request that activities which are using the same resource (e.g., a room, a teacher, a section) can not overlap in time or that a resource is of a certain capacity, restricting e.g. how many activities can use it at the same time. In addition, there are usually relations between activities and constraints restricting what resources an activity should or can use. There are different types of timetabling problems for example examination, transport, employee rostering, course timetable etc. In this thesis we will concentrate on course timetable.

The early techniques used in solving timetabling problems were based on a simulation of the human approach in resolving the problem. These included techniques based on successive augmentation that were called direct heuristics. These techniques were based on the idea of creating a partial timetable by scheduling the most constrained lecture first and then extending this partial solution lecture by lecture until all lectures were scheduled. The next step was for researchers to apply general techniques like integer and linear programming; graph coloring and network flow to solve the timetable problem. The major general techniques that seemed to have been prevalent in the 1970's and 1980's have their roots in artificial intelligence and are based on algorithms supported by simulated annealing, Tabu search and genetic algorithm methods.

Methodology

The generation techniques developed are such as Integer Programming/Linear Programming Constraint Satisfaction Programming Genetic and Evolutionary Algorithms Simulated Annealing. The Linear and Integer Programming techniques, the first applied to timetabling, were developed from the broader area of mathematical programming. Mathematical programming is applicable to the class of problems characterized by a large number of variables. The construction of a linear programming model involves three successive problem- solving steps. The first step identifies the unknown or independent decision variables. Step two requires the identification of the constraints and the formulation of these constraints as linear equations. Finally, in step three, the objective function is identified and written as a linear function of the decision variables.

Evolutionary Algorithms (EAs) are a class of direct, probabilistic search and optimization algorithms gleaned from the model of organic evolution. A Genetic Algorithm (GA) is a type of EA and is regarded as being the most widely known EA in recent times. GAs are a class of stochastic search algorithms based on biological evolution whose search strategy mimics natural selection by using an automated version of the "survival of thefittest".

Tabu Search (TS) is a meta-heuristic technique that guides a local heuristic search procedure to explore the solution space beyond local optimality.

Prototype Generic Algorithm

1. Define Planning Entity
    a. Define Planning Variables
    b. Provide planning variable range provider (Optional)
2. Define Planning Solution
    a. Include Collection of Planning Entities
    b. Provide planning range provider (Optional)
3. Choose Constructive Heuristic
4. Start generating solution for problem
    a. Generate Solution
    b. Run score calculator over solution
    c. Get score
    d. Add solution to space with score value
5. Repeat Step 4 till reach optimal score value
6. Then use Meta Heuristic for finding a proper solution
    a. Move Planning entities based on variable then
    b. Check score
    c. if optimal update best score solutions till
7. Reach expected score value
8. Show the solution

Scoring Algorithm

1. Run Solution through all rules written.
2. If rule is getting violated update the score value in handler of score
    a. If the rule is dedicated to hard constraint then assign negative value to hard score.
    b. If rule is dedicated to soft constraint then assign negative value to soft score.
3. Else move to next rule.
4. Repeat step 2-3 till all rules applied.
5. Last value of handler is the score value of solution
    a. Either of type integer/Long
6. Or some HARD/SOFT combination

Result & Statistics

Figure 1 Time Spent

Table 1 Algorithm Comparison

Algorithm	Scalable ?	Optimal ?	Easy to use ?	Tweakable ?	Requires CH?
Exhaustive Search (ES)					
Brute Force	0/5	5/5	5/5	0/5	No
Branch And Bound	0/5	5/5	4/5	2/5	No
Construction heuristics (CH)					
First Fit	5/5	1/5	5/5	1/5	No
First Fit Decreasing	5/5	2/5	4/5	2/5	No
Weakest Fit	5/5	2/5	4/5	2/5	No
Weakest Fit Decreasing	5/5	2/5	4/5	2/5	No
Strongest Fit	5/5	2/5	4/5	2/5	No
Strongest Fit Decreasing	5/5	2/5	4/5	2/5	No
Cheapest Insertion	3/5	2/5	5/5	2/5	No
Regret Insertion	3/5	2/5	5/5	2/5	No
Metaheuristic (MH)					
Local Search					
Hill Climbing	5/5	2/5	4/5	3/5	Yes
Tabu Search	5/5	4/5	3/5	5/5	Yes
Simulated Annealing	5/5	4/5	2/5	5/5	Yes
Late Acceptance	5/5	4/5	3/5	5/5	Yes
Step Counting Hill Climbing	5/5	4/5	3/5	5/5	Yes
Evolutionary Algorithms					
Evolutionary Strategies	4/5	3/5	2/5	5/5	Yes
Genetic Algorithms	4/5	3/5	2/5	5/5	Yes

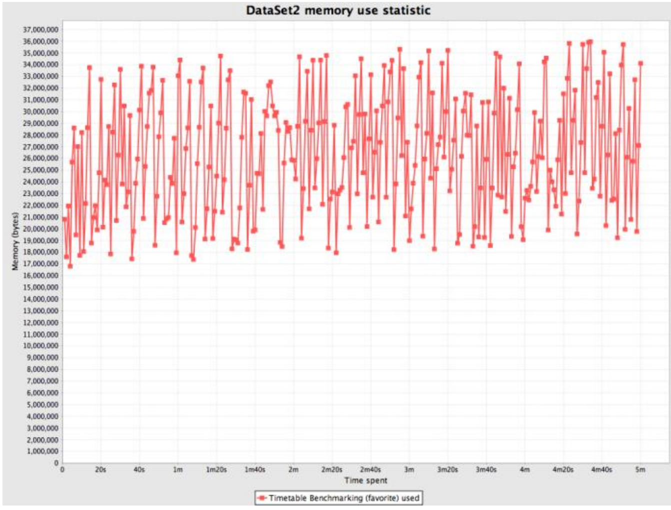

Figure 2 Memory Usage

CONCLUSION & FUTURE SCOPE

In this paper I have examined the timetable scheduling problem. It began with a discussion into the terminology and size of the research area. A strong distinction between the terms scheduling and timetabling was made to avoid confusion in the work presented.

The future research is the application and further development of the benchmark proposed in the work. Now I have provided the prototype so in future, morework could be done regarding timetable problem and implement the research to develop a system, which generate the timetable within proper time and with feasible constraints.

References

[1] Abramson, D. 1991. Constructing School Timetables Using Simulated Annealing: Sequential and Parallel Algorithms Management-Science.

[2] Abramson, D. and Dang, H. 1993. School Timetables: A Case Study in Simulated Annealing," Applied Simulated Annealing, Lecture Notes in Economics and Mathematics Systems, Springer-Verlag, Ed.

V. Vidal.

[3] Abramson, D., Amoorthy, M. K. , and Dang, H. May 1999. Simulated annealing cooling schedules for the school timetabling problem. Asia - Pacific Journal of Operational Research; Singapore.

[4] Boizumault, P. ,Delon Y. and Peridy L. 1995. Constraint Logic Programming for Examination Timetabling, The Journal of Logic Programming.

[5] Gunadhi-H, Anand-VJ, and Wee-Yong-Yeong, 1994. A DSS approach to timetable scheduling. Proceedings of the Second Singapore International Conference on Systems. SPICIS `94. Japan- Singapore AI Centre, Singapore; 1994; 576 pp.

[6] Glover F. 1993. Future path for integer programming and links to artificial intelligence. Computer & Ops. Res.

[7] Glover F. , 1986. Future Paths for Integer Programming and Links to Artificial Intelligence. Computers and Operations Research.

[8] Glover, F. , 1997. A Template for Scatter Search and Path Relinking. Lecture Notes in Computer Science.

[9] Glover, F. , 1989. Tabu search- Part I. ORSA Journal on computing.

[10] Goldberg, 1989. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley.