ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

Genetic Approach Time Table Generator – A Case Study

Asif Ansari[#1], Prof Sachin Bojewar[#2]

PG Scholar, AlamuriRatnamala Institute of Engineering & Technology, Mumbai, India[#1]

Asst. Professor, Vidyalankar Institute of Engineering & Technology, Mumbai, India[#2]

ABSTRACT: A college timetable is a temporal arrangement of a set of classes and classrooms in which all given constraints are satisfied. Timetabling has long been known to belong to the class of problems called NP hard. This project introduces a practical timetabling algorithm capable of taking care of both Hard and soft constraints effectively, used in an automated timetabling system.
The Genetic Algorithm is main component of project which produces the XML based weekly timetable sheet as the output. The project takes various inputs from the user such as Teacher List, Course List, Semester List, Room List, Day List and Timeslot as well as various rules, facts and constraints using Genetic Algorithm, which are stored in XML based knowledge base. This knowledge base serves as input to our Timetable Generator Algorithm

KEYWORDS: NP Hard, XML, Genetic algorithm, hard and soft constraints.

I. INTRODUCTION

The algorithm are written in JAVA, which makes our project platform independent. Further benefits of choosing these frameworks are explained in later part of report with practically acceptable results.
The problem consists of the following entities:
Days, Timeslots, and Periods. We are given a number of teaching days in the week (typically 5 or 6). Each day is split in a fixed number of timeslots, which is equal for all days. A period is a pair composed by a day and a timeslot. The total number of scheduling periods is the product of the days times the day timeslots.

Courses and Teachers. Each course consists of a fixed number of lectures to be scheduled in distinct periods, it is attended by given number of students, and is taught by a teacher. For each course there is a minimum number of days that the lectures of the course should be spread in, moreover there are some periods in which the course cannot be scheduled.

Rooms. Each room has a capacity, expressed in terms of number of available seats. All rooms are equally suitable for all courses (if large enough).

Curricula. A curriculum is a group of courses such that any pair of courses in the group have students in common. Based on curricula, we have the conflicts between courses and other soft constraints.
The solution of the problem is an assignment of a period (day and timeslot) and a room to all lectures of each course.

II. LITERATURE SURVEY

There are many heuristic algorithms like simulated annealing, tabu search, genetic algorithm which can be useful to achieve a good quality solution, but using a combination of these algorithms was apparently more useful.

2.1 MATHEMATICAL PROGRAMMING

Mathematical programming involves the optimization of a function given a set of constraints that have to be met. Integer linear programming has been used to solve university timetabling problems but given the nature of high

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

computational demand it is not recommended for large timetabling problems [2,3]. Integer programming was used at Shahrood University of Technology in Iran to solve a course timetabling problem where about 200 subjects over two terms (one year) were scheduled in the faculty of science [4].

2.2 BRANCH AND BOUND

A branch and bound algorithm searches the entire search space but uses bounds on the function to be optimized and the current best solution to search through the search space. A dynamically search tree is created using unexplored subspaces which are represented as nodes. There are three major components that are used: node selection, bound calculation and branching [5, 6].

Branch and bound was used by Busetti to solve a timetabling problem at Purdue University with the option of users having the ability to make adjustments interactively [7]. Another use of branch and bound was used by Friedrich, Hofsäβ, and Wekeck in scheduling transit assignments and reduced the computational time from 32.5 hours (shortest path search) to 1.6 hours (branch and bound search) [8]. According to Fang branch and bound is not suitable for large timetabling problems [3].

2.3 SIMULATED ANNEALING

Simulated annealing is the process of heating a metal to a high degree and then letting it cool down to a minimum temperature to form a solid. By simulating this process, simulated annealing can find a global minimum. One of the advantages of using simulated annealing is that it avoids being trapped in a local minimum. The algorithm uses a random search that only accepts changes if it decreases the objective function that we are trying to minimize but also uses a probabilistic function that allows it to increase the objective function. Using simulated annealing requires the following: a representation of possible solutions, a generator of random solutions, an objective function and an annealing schedule (temperature rules) [7, 9].

III. GENETIC ALGORITHM

The Genetic Algorithm is main component ofthe which produces the timetable even / odd semester sheet as the output.
The project takes various inputs from the user such as Teacher List, Course List, Semester List, Room List and Day

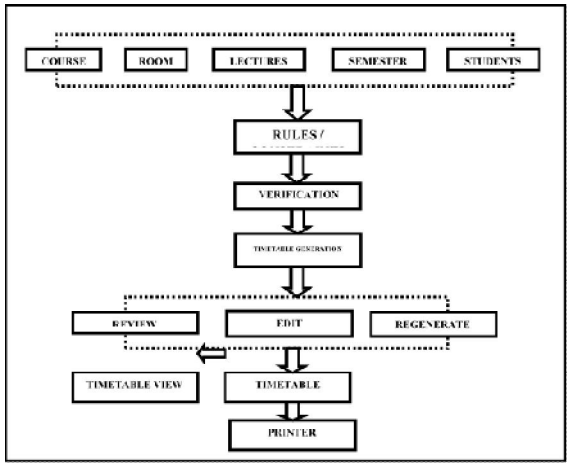

Figure 1: Proposed time table

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

IV. METHODOLOGY

With Timetable Generation System, the process involves just a few simple steps:

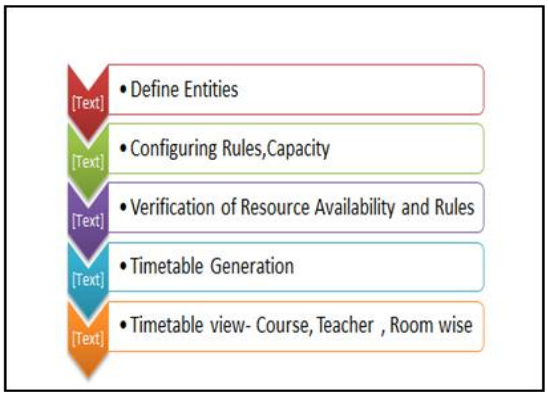

Figure 2: Timetable Generation process

V. EXISTING SYSTEM

Timetabling is the whole process concerned with making a timetable having events arranged according to a time when they take place which must be subject to the timing constraints of each entity placed in the table. These courses are usually taught by varied lecturers in different departments who may also wish to specify some timing constraints on their courses. Given all the courses and course details, the educational administrative staff in college is charged with the responsibility of creating a near optimal timetable which would serve as a guide for academic activities in the college.

The traditional manual timetabling system is time-consuming, resource-intensive, involves many steps and requires re-processing the same data several times.

VI. PROPOSED SYSTEM

The System proposes an optimized technique to automate time table generation system. Time table generation system involves various challenging constraints of resources including faculties, rooms, time slots etc. The proposed technique filters out the best of active rules and Genetic algorithm to generate the optimized solution. Genetic Algorithm and Active Rules together form a complete sphere for developing a system, which needs to satisfy various constraints. Active Rules provide “event-condition-action” model for the implementation of any rule based system.

Planning timetable is one of the most complex and error prone applications. There are still serious problems like generation of high cost time table are occurring while scheduling and these problems are repeating frequently. Therefore there is a great requirement for an application distributing the course evenly and without collisions. The aim is here to develop a simple, easily understandable, efficient and portable application which could automatically generate good quality time table within a second. Active rules are described for the knowledge of intelligent agents (i.e. Constraints), GAs are described and their use in optimizing rule based agent is proposed, methods are apply to the problem of optimizing some results of this application are presented and finally, some conclusion and possible direction for future research are presented.

The structure of time table generator consist Input Data Module, relation between the input data module, time slots module, applying active rules and GA module then extract the reports.

VII. INPUT FORMAT

Each instance is in a single file, containing a file header and four sections: courses, rooms, curricula, and constraints. The header provides all scalar values and each section provides the arrays for that specific aspect of the problem. The exact format is shown by the following example

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

Name	Courses	Rooms	Days	Period/ Day	Curricula	Constraints

Subject Name	Lecturer Name	Hours/Week	Working Days	Strength of class

Class Room No	Capacity

Course Name	Number of Courses	List of Courses(subject)

Figure 1: Input Format

VIII. GENETIC ALGORITHM

Genetic algorithms are methods of solving problems based upon anabstraction of the process of Natural Selection. They attempt to mimic nature by evolving solutions to problems rather than designing them. Genetic algorithms work by analogy with Natural Selection as follows. First, a population pool of chromosomes is maintained. The chromosomes are strings of symbols or numbers. There is good precedence for this since humans are defined in DNA using a four-symbol alphabet. The chromosomes are also called the genotype (the coding of the solution), as opposed to the phenotype (the solution itself). In the Genetic algorithm, a pool of chromosomes is maintained, which are strings. These chromosomes must be evaluated for fitness. Poor solutions are purged and small changes are made to existing solutions and then allow "natural selection" to take its course, evolving the gene pool so that steadily better solutions are discovered. The basic outline of a Genetic Algorithm is as follows: Initialize pool randomly The randomly assigned initial pool is presumably pretty poor. However, successive generations improve, for a number of reason.

IX. RESULT AND DISCUSSION

Output screen for generating timetable

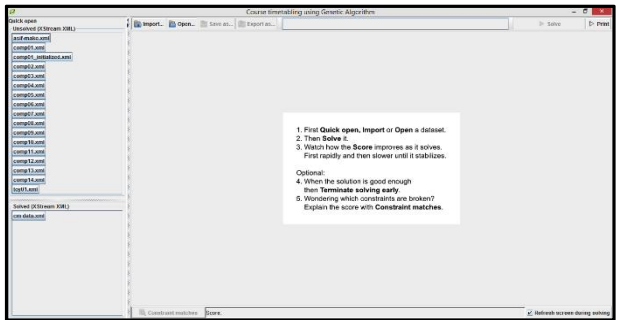

Figure 5: Output screen for generating timetable

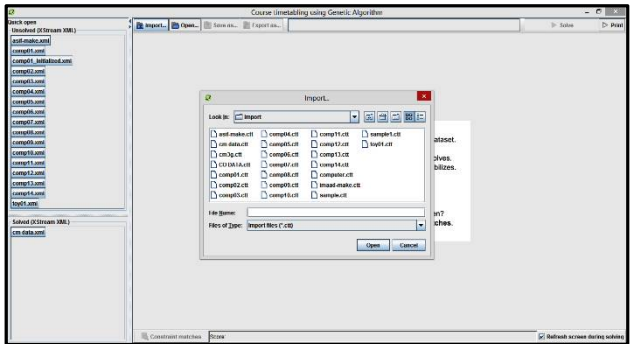

Figure 6: Importing data

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

Panel showing Room, Teachers and Curricula

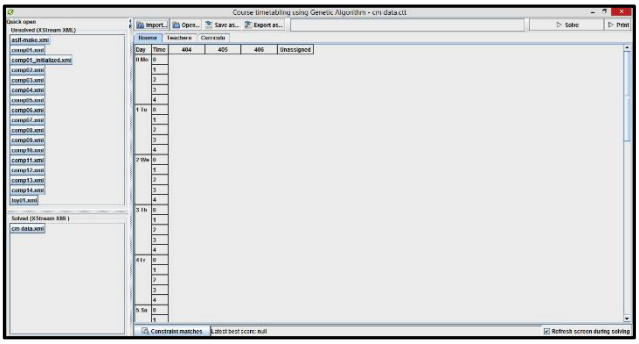

Figure 7: Panel showing Room, Teachers and Curricula

Teacher wise allotment

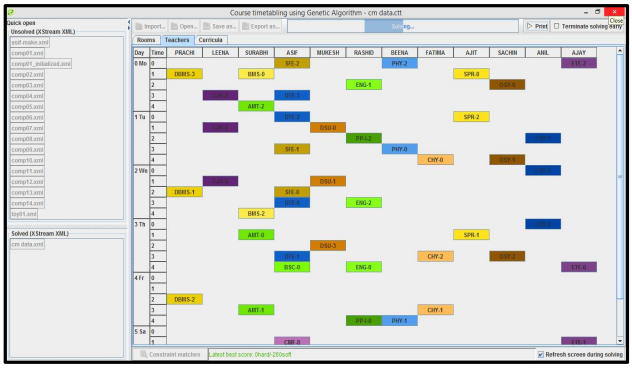

Figure 8: Teacher wise allotment

Curricula wise allotment

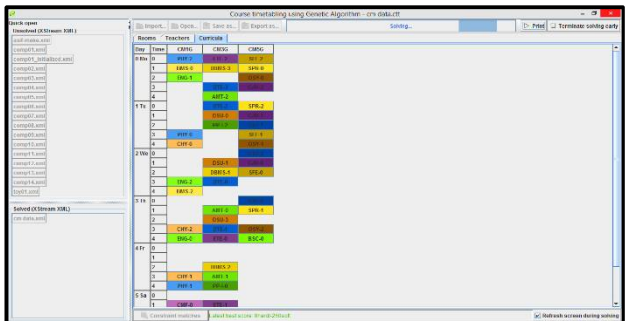

Figure 9: Curricula wise allotment

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

Print command

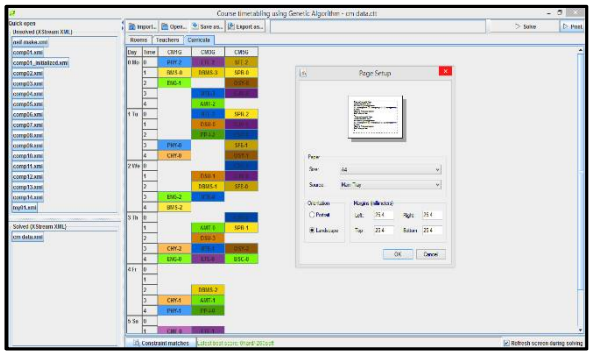

Figure 10: Print command

The print command can be given according to the room wise, teacher wise or curricular wise

X. CONCLUSIONS AND FUTURE SCOPE

Generally, this system can be considered a useful system since it helps to improve their process of preparing the timetable. The GA in timetabling framework has been shown to be successful on several real problems .It has been shown that the genetic algorithm perform better in finding areas of interest even in a complex, real-world scene. One could argue that the genetic algorithm can find a local optimum and then stop. This is always a danger with a genetic algorithm, but again it depends on the search space. In this time table generation approach, there are many good solutions and the genetic algorithm will find one of them.

REFERENCES

1. Michael R. Garey and David S. Johnson. Computers and intractability: a guide to the theory of np-completenesspage no 13-14. 1979.
2. S. Petrovic S. and E. Burke. University timetabling. In Handbook of scheduling: Algorithms, models, and performance analysis, pages 1-23. CRC Press, Boca Raton, FL, 2004.
3. H. L. Fang. Genetic algorithms in timetabling scheduling. Ph.D. thesis, University of Edinburgh, Edinburgh, UK, 1994.
4. S. A. MirHassani. A computational approach to enhancing course timetabling with integer programming. Applied Mathematics and Computation, 175: 814-822, 2006.
5. M. L. Pinedo. Planning and scheduling in manufacturing and services. In Springer Series in operations research, pages 3-8. Springer, New York, 2005.
6. J. Clausen. Branch and bound algorithms- principles and examples, 1999. http://citeseerx.ist.psu.edu/
7. F. Busetti. Simulated annealing overview, 2009. http://www.cs.ubbcluj.ro/~csatol/mestint/pdfs/Busetti_AnnealingIntro.pdf.

BIOGRAPHY

Asif Ansari, works a lecturer with Babasaheb Gawade's Institute of Technology from last 12 years. A very hard working and student friendly professor with good grip and command over latest trends and development in technology. At present perusing his masters from Mumbai University.