Imperial Journal of Interdisciplinary Research (IJIR)
Vol-2, Issue-5, 2016
ISSN: 2454-1362, http://www.onlinejournal.in

Genetic Algorithm With Meta-Heuristic Approach For Generating Timetable

Anand Bali[1], Anam Shaikh[2], Tazeen Ansari[3] & Faizan Baig[4]
[1,2,3,4]Dept. of Computer Engineering, M.H.S.S.C.O.E.

Abstract*: A college timetable consists of set of lectures and classrooms. Creating such timetables manually is complex and tedious. By automating the process of timetable generation, we can save a lot of valuable time of authorities who are involved in scheduling and maintain course timetables. Therefore, there is a need to develop a practical approach for building timetabling system for courses, which can be made to fit to any colleges timetabling problem in which genetic algorithm can be used. The university timetable problem asks us to search some classrooms and timeslots, which satisfy the given hard constraints in our project we make use of genetic algorithm with meta heuristic approach for generating timetable. In each step we calculate a fitness function for every individual and select the best population in order to produce children for next generation and give an optimal solution.*

Keywords- *Question Genetic Algorithm, Heuristic Approach, Fitness Function.*

Introduction

There are many colleges that perform their administration task automatically but still the timetable-scheduling task is done manually. Since the timetable, generation is quite burdening and time-consuming task. [1]
The lecture-timetable scheduling includes a constraint that fulfills the problem in which we find a solution for the given constraints. In timetable scheduling we find some free classrooms with time slots, which meets the constraints forced on offered courses, lecturers, classrooms and so on. The computation time for timetabling increases exponentially as the variables increases in number. There are various approaches made in the past for constructing timetables for various educational structures. Timetabling problems can be solved by various techniques such as branch and bound, mathematical programming, tabu search and simulated annealing, in order to schedule timetable, in our project we are introducing a System which would automatically generate realistic timetable for the institute. teachers will be allotted to various courses in accordance with all possible constraints needed to be fulfilled to generate proper timetable. The timetable scheduling is dependent on two things Hard constraints and Soft constraints. [4]

Problem Formulation

Generating timetable is an NP-complete scheduling problem. It is not considered as a standard job-shop issue as the allocation of classrooms required to be considered, it is highly constrained, but above all the problem will not be same for different universities. It is arduous to develop a universal program, considering every imaginary aspects of the timetabling problems. Although constructing timetable manually is very tedious task, it is still used by most of the universities, due to the lack of proper computing solution.

A timetabling algorithm can use various strategies to get a better solution while satisfying all the hard constraints. Violations can either be avoided from the outset or penalized to lead the algorithm towards better solutions and introduce repair mechanisms.
The considered university is of 3-year duration, which has the following attributes for its timetable scheduling.
1. The classes for students are scheduled in the weekday's that can be specified as 5 to 6 working days
2. Different types of lectures
• Theory lecture
• Tutorial
3. The size of theory lectures can vary from 40-70.
4. 1 Hour is a minimum timeslot interval. Theory classes takes 1 timeslot and tutorial takes 1 timeslot.

Once a lecturer agrees to offer a course for a specific semester of a department, an offered course Yi takes place in the scheduling timetable problem, which is expressed as a tuple of attributes. Except Rooms and timeslots, all attributes of Yi are determined at the time the course is decided to be offered. Both Rooms and timeslots are list fields to contain appointed time slots and classrooms for the course Yi. To show an

attribute of an offered course Yi, we use the notation Yi. attr. [2]

Problem Definition

The Genetic Algorithm is our main component which generates the timetable of every semester as the output.
The project takes number of inputs from the user such as Courses, Rooms, Teachers and Days as well as various Constraints, Rules and Facts which are stored in XML based knowledge base. [3]
This knowledge base acts as input to our Genetic Algorithm residing on system. Our knowledgebase is in the center, because it is between our Genetic algorithm and GUI front end.
Timetable Generation involves Following steps:

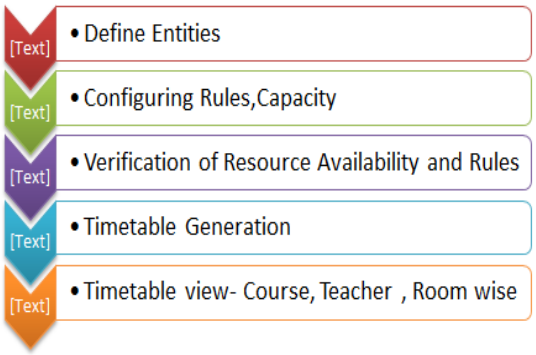

Figure 1: - Timetable Generation process

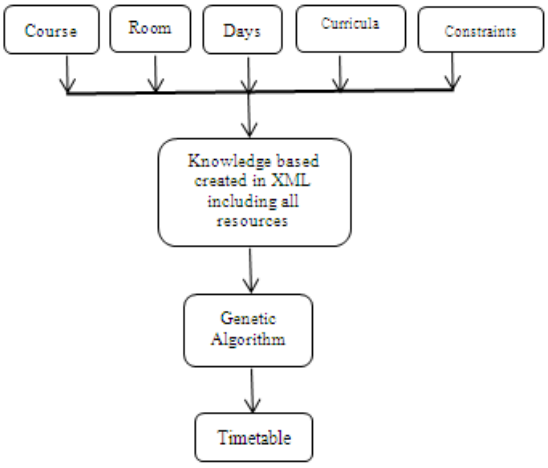

Figure 2: Proposed System for timetable Generation.

Implementation and Working

Timetable generation system consists of various challenging constraints of resources including rooms, faculties, time slots etc. The proposed method filters out the best of Genetic Algorithm and Active-Rules to generate the better solution. Active Rules and GA form a complete sphere for developing a system, which needs to satisfy various constraints. Active Rules gives “event-condition-action” model for the implementation of any rule based system. [5]

The task of the work is to generate a model which will be used to generate the schedule using the operators.
The structure of time table generator consists Time slots Module, Data Module, Genetic Algorithm module, relation between the input data module and applying active rules then generate the reports. [6]

1.1. Genetic Algorithm

Genetic Algorithms are adaptive heuristic search algorithm based on the evolutionary ideas of natural selection and genetics. As such, they represent an intelligent exploitation of a random search used to solve optimization problems. Although randomized, GAs are by no means random, instead they exploit historical information to direct the search into the region of better performance within the search space. The basic techniques of the GAs are designed to simulate processes in natural systems necessary for evolution, especially those follow the principles first laid down by Charles Darwin of "survival of the fittest.". Since in nature, competition among individuals for scanty resources results in the fittest individuals dominating over the weaker ones.

GAs simulate the survival of the fittest among individuals over consecutive generation for solving a problem. Each generation consists of a population of character strings that are analogous to the chromosome that we see in our DNA. Each individual represents a point in a search space and a possible solution. The individuals in the population are then made to go through a process of evolution.

GAs are based on an analogy with the genetic structure and behavior of chromosomes within a population of individuals using the following foundations:

- Individuals in a population compete for resources and mates.
- Those individuals most successful in each 'competition' will produce more offspring than those individuals that perform poorly.
- Genes from `good' individuals propagate throughout the population so that two good parents will sometimes produce offspring that are better than either parent.
- Thus each successive generation will become more suited to their environment.[7]

Generation
{
Select good solutions to breed new population
Parents generate new solution
Evaluate new solutions for fitness
Old population replaced by new population
}
The randomly assigned initial pool is considered to be poor. However, successive generations improve, for a number of reasons:

After an initial population is randomly generated, the algorithm evolves the through three operators:
selection which equates to survival of the fittest;
crossover which represents mating between individuals;
mutation which introduces random modifications.

1. Selection Operator

- Give priority to better individuals, letting them to pass on their genes to the next generation.
- Nature of each individuals depends upon their fitness level.

2. Crossover Operator

- Prime distinguished factor of Genetic Algorithm from other optimization techniques
- Selection operator chooses two individuals from the population
- A crossover site along the bit strings is randomly chosen
- Two strings exchange their values up to this point
- If S1=11111 and s2=00000 and the crossover point is 2 then S1'=00111 and s2'=11000
- The two new offspring are added to the next generation of the population
- This process is likely to create better individuals by combining the portions of good individuals

3. Mutation Operator

- With some low probability, a portion of the new individuals will have some of their bits flipped.
- Its purpose is to inhibit premature convergence and preserve diversity within the population.
- Mutation alone causes a random walk through the search space
- Selection and Mutation generates noise-tolerant, hill-climbing algorithms, parallel. [7]

1.2. Algorithm

// Initialize generation 0:
k: = 0;
POPk: = Randomly generated population of individuals.
// Evaluate POPk:
Calculate fitness(j) for each j ∈ POPk;
do
{// Create generation k + 1:
// 1. Copy:
Select (1 −) × n members of POPk and insert into POPk+1;
// 2. Crossover:
Select × n members of POPk; pair them up; generate offspring; include the offspring into POPk+1;
// 3. Mutate:
Select μ × n members of POPk+1; invert a randomly-selected bit in each;
// Evaluate POPk+1:
Calculate fitness(j) for each j ∈ POPk;
// Increment:
k: = k + 1;
}
While fitness of fittest individual in population POPk is not high enough;
return the fittest individual from population POPk;

1.3. Hard constraints

Lectures: lectures of a course must be planned and scheduled they must be assigned to distinct periods.

Room Occupancy: More than one lecture cannot take place in the same classroom in the same duration.

Conflicts: Lectures of courses in similar curriculum or taken my same teacher must not conflict and should be scheduled in different sessions.

Availabilities: If teacher in not available

1.4. Soft constraints:

Room Capacity: For every lecture, the number of students that appear the course must be less or equal to the number of seats of all the classrooms in which the lecture is held.

Minimum Working Days: The lectures of every course must be spread in minimum span of days.

Room Stability: Rooms should be stable i.e. every lectures should be held in same room.

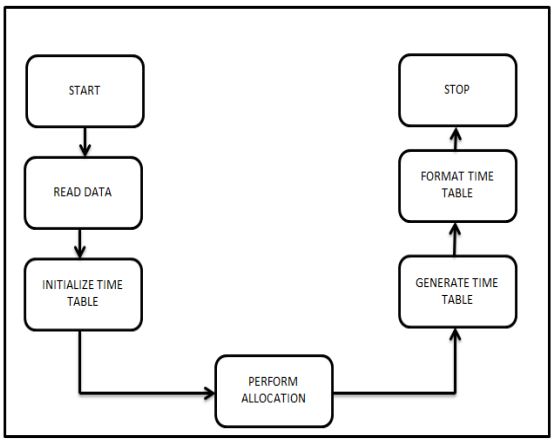

Figure 3: - Shows the actual process of timetable generation.

1.5. Input Format

Every item is in a single file, which contains a file header and four sections: courses, rooms, curriculum, and constraints. The header gives all scalar values and each section gives the arrays for that particular aspect of the problem. The format for the input is shown below.

Name: SABOO COMP
Courses: 10
Rooms: 3
Days: 6
Periods_per_day: 4
Curriculum: 3
Constraints: 0
COURSES:
DWM SAIKA 3 6 60
HMI AHLAM 3 6 60
PDS NAZNEEN 3 6 60
ML WAHIDA 3 6 60
CC ASADULLAH 1 6 60
DF ASADULLAH 3 6 60
SPCC NAFEESA 3 6 60
DD SAIKA 3 6 60
SOAD SHABANA 3 6 60
CN ASADULLAH 3 6 60
ROOMS:
404 60
405 60
406 60
CURRICULUM:
COMP8 DWM HMI PDS ML CC DF
COMP6 SPCC DD SOAD CN
UNAVAILABILITY_CONSTRAINTS:
END.

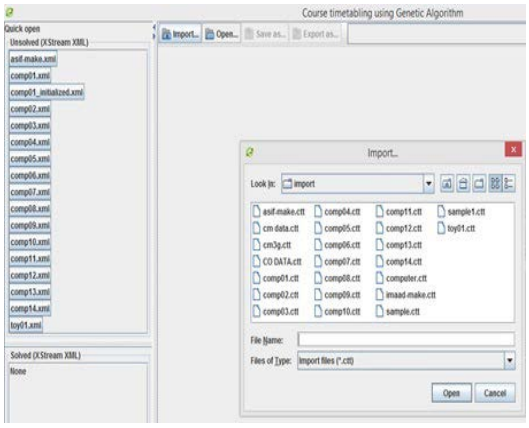

Figure 4: - The input format is stored in an xml file and is imported using the above dialog.

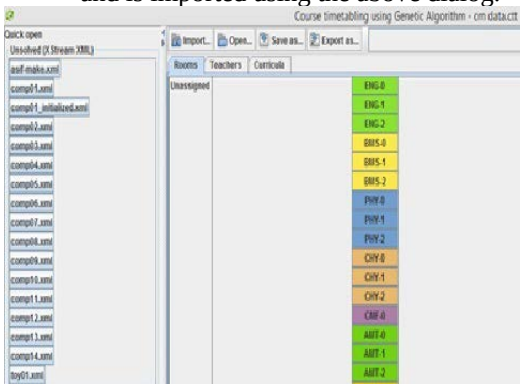

Figure 5: - After importing the file, the unassigned timetable for the curriculum will be as above.

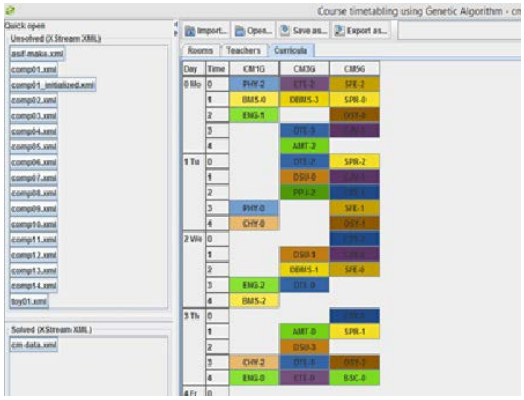

Figure 6: - Curriculum wise allotment

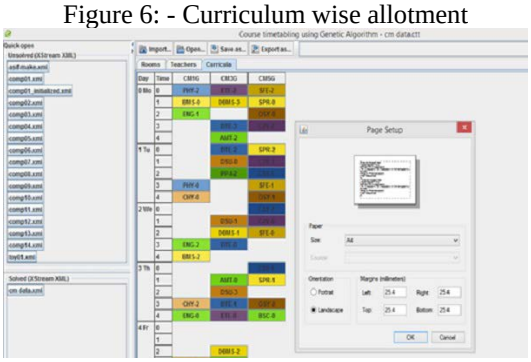

Figure 7: - Print command for the timetable

Future Scope

To schedule timetable for the university with no human errors and which will be less time consuming along with high level of flexibility and precision. Moreover, improve the overall process of timetable scheduling with help of genetic algorithm along with the use of different technologies.

For Further work there is need to implement different types of genetic algorithms, such as heuristic approach to generate the timetable. For example, the individual with overlapping populations such as steady state or incremental Genetic Algorithm. In such cases, that the Genetic Algorithm could be used for creating the nodes at real time (once an initial good state has been attained) and not just training them. Plan to investigate other techniques for finding the optimal rule set (for example, NN or other heuristic search methods like simulated annealing) and differentiate the results with manually generated results obtained by a statistical analysis of the network.

Conclusion

The Timetable Generation application will ease the process of timetable scheduling which may otherwise needed to be done non-automatically by the center staffs possibly leading to constraints problem that are difficult to determine when time table is scheduled. The intention of algorithm is to generate the timetable automatically is satisfied. To improve the quality of search operation, the algorithm incorporates many techniques. It also addresses the important hard constraint.

Acknowledgment

Our thanks to M.H. Saboo Siddik College of Engineering, Department of Computer Engineering, for giving us the initiative to do constructive work. We also thank anonymous reviewers for their constructive suggestions.

References

[1] J. J. Grefenstette, editor. Proceedings of the Second International Conference on Genetic Algorithms and their Applications. Practice and Theory of Automated Timetabling VI Proceedings of The 6thInternational Conference on the Practice and Theory of Auto

[2]http://documents.mx/documents/time5461de72b1af9f1f4e8b45cf.html

[3] Alberto Colorni, Marco Dorigo “A Genetic Algorithm to solve the time table problem” Available : http://citeseerx.ist.psu.edu

[4] Om Prakash Shukla, Amit Bahekar, Jaya Vijayvergiya, ”Effective Fault Diagnosis and Maintenance Optimization by Genetic Algorithm” Available : http://researchjournals.in/documents/published/2204.pdf

[5]http://www.ijritcc.org/download/1416628378.pdf

[6] Leon Bambrick, “Lecture Timetabling Using Genetic Algorithms” Available : http://secretgeek.net/content/bambrilg.pdf

[7]http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol1/hmw/article1.html