Time Table Scheduling Using Genetic Algorithm

Harshal R. Kotwal[1], Hitesh G. Firake[2], Anuradha B. Tomar[3], Radha A. Varma[4], Pooja Y. Warade[5]
[1]Assistant Professor, Dept of Computer Engineering
[2, 3, 4, 5]Dept of Computer Engineering
[1, 2, 3, 4, 5] SSBT's COET, Maharashtra, India,

Abstract- *There are various approaches used by the researchers to develop an automatic time-table scheduling. But traditional heuristic approaches are inefficient and inflexible when solving scheduling problems. The genetic algorithm can be used to design and develop the automatic time table scheduling for an institute. The crossover and mutation these are the parameters of the genetic algorithm that is used to find unique and strong generation along with predefined fitness functions. After analysis if there is higher fitness value then it means strong solution, after that based on fitness value then parents are selected to reproduced new generation.*

Keywords- Genetic Algorithm, Timetable Scheduling, Np-Hard problems.

I. INTRODUCTION

Scheduling of timetable is quite tedious job and it may lead to certain errors while generating the timetable. Time table generated must follow certain rules and constraints where as it may lead to a serious problem when it is related with high cost, this problem occur frequently. Thus our approach is to create a simple time table scheduler which will schedule teachers, students and classes according to the constraints and all of this are not repeated more than once per period. The scheduling of timetable uses the genetic approach in which the best solution is obtained by performing the crossover and mutation over initial population and using fitness value till when the best offspring is generated. Our approach making use of genetic algorithm has a large problem space.

The constraints are classified into Hard Constraints and Soft constraints. Hard constraint that must not be violated while, soft constraints can be violated if necessary.

1.1 Scope

This study aims to construct timetable Generation System generates timetable for each class and teacher, in keeping with the availability calendar of teacher's availability and capacity of physical resources such as classrooms, laboratories and computer room and rules applicable at deferent classes, semesters, teachers and subjects level. Best of all, this Timetable Generation System tremendously improves resource utilization and optimization.

1.2 Objective

The various objectives include:

- We are putting Genetic Algorithm which would automatically generate timetable for the different courses of the institute.
- Courses and lectures will be scheduled in according with all the possible constraints and the given inputs and thus, a timetable will be generated.
- The system will allow interaction between the staff and students and at the same time enable them to upload their changes.
- The necessary changes and the additional constraints for the next week timetable will also be considered by this timetable generation system.

II. LITERATURE REVIEW

Branimir Sigl, Marin Golub et.al [1] In this paper a genetic algorithm for solving timetable scheduling problem is described. The algorithm was tested on small and large instances of the problem. Algorithm Performance was significantly enhanced with modification of basic genetic operators, which restrain the creation of new conflicts in the individual.

Barkha and Ambika et.al [2] This paper 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.

Leon Bombrick et.al [3] This paper details the implementation of a computer program which employs Genetic Algorithms (GAs) in the quest for an optimal lecture timetable generator. GA theory is covered with emphasis on less fully encoded systems employing non-genetic operators.

The field of Automated Timetabling is also explored. A timetable is explained as, essentially, a schedule with constraints placed upon it. The effects of altered mutation rate and population size are tested. It is seen that the GA could be improved by the further incorporation of repair strategies, and is readily scalable to the complete timetabling problem.

III. METHODOLOGY

The detailed methodology adopted in this study is explained below. The methodology includes constraints and genetic operators. Further this data is used to develop genetic algorithm models for creating time table scheduler.

3.1 Constraints

Time table scheduler needs to follow certain constraints. These constraints are classified into two types i.e. soft and hard constraints.

3.1.1 Hard Constraint:

Hard constraints are the type of constraints that was be strictly followed by the system and violation of which leads to certain damage to the system. Following are some hard constraints which our system must follow:

1. There should not be any lecture or practical during lunch break.
2. A student should have only one class at time.
3. A teacher should have only one class at time.
4. A room should be booked only for one class at time.
5. No teacher can be present in two different classes in the same hour.

3.1.2 Soft Constraints:

Soft constraints are the type of constraints which can be avoided sometimes if needed it does not cause any damage to the system if get avoided. Below mentioned are some of soft constraints which our system follows.

1. Student should not have any free time between two classes on a day.
2. Faculty should have work load less than 5 hr.
3. No continues lecture for the same faculty on the same class.
4. Scheduling of teachers should be well spread over the week.

3.2 Genetic Operators

There are usually three operators:

1. **Selection:** During each generation a proportion of existing population is selected. To create a new generation the selection based on the fitness value and fitness function.
2. **Mutation:** Randomly selected bits in a solution are interchange to get another better solution. Shown in fig. 3.1.

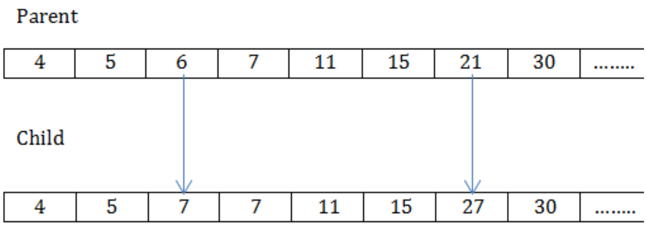

Fig. 3.1 Mutation.

3. **Crossover:** Substring from the pair of solution is swapped to form a new pair of solution. Shown in fig. 3.2.
4.

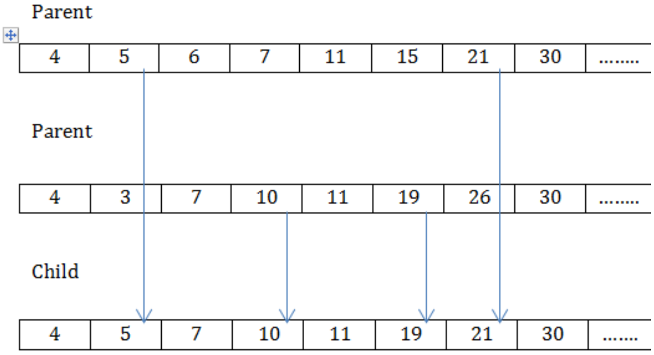

Fig. 3.2 Crossover.

3.3 Genetic Algorithm

Genetic algorithm attempt to incorporate ideas of natural evolution an initial population is created consisting of randomly generated rules. Based on the survival of the fittest a new population is formed to consist of the fittest rules in the current population as well as offspring of these rules. Offspring are generated by applying the genetic operators such as crossover and mutation.

According to J.S.Kim et.al Genetic algorithms are adaptive search techniques that can find the global optimal solution from an initial population of sample solutions by manipulating and generating recursively a new population of generation. Genetic algorithm provides operators such as reproduction, crossover and mutation which are develop by using natural solution.

In genetic algorithm initially the initial population is generated. For each generation select good solution to find new population create new solution from parent check the solution for the fitness if sufficient quality is not reached and minimum number of iterations are not completed then replace the old population with the new one. This creates random initial pool. Following operation are performed on the replaced solution, shown in fig 3.3.

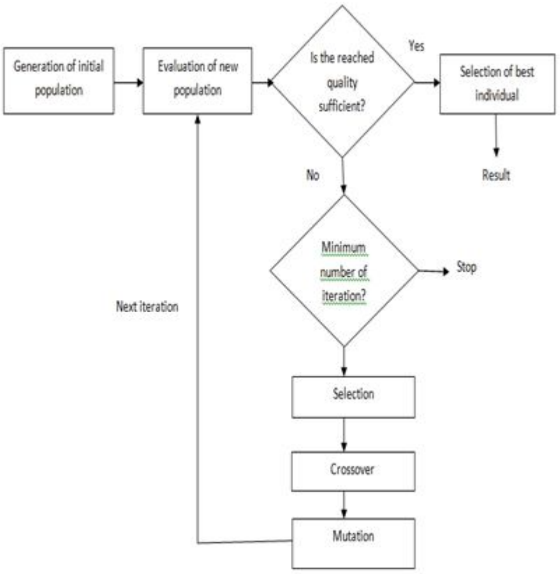
Generation of initial population
Evaluation of new population
Is the reached quality sufficient?
Yes
Selection of best individual
Result
No
Minimum number of iteration?
Stop
Next iteration
Selection
Crossover
Mutation

Fig. 3.3 Workflow of genetic algorithm.

V. CONCLUSIONS

An implementation of the GA developed model for Time table scheduling has resulted in Optimized output With reduced cost.This project report described how set of active rules can be used to express the knowledge of intelligent and how a genetic algorithm can be used to dynamically prioritize rules in the face of dynamically evolving environments. In this time table generation approach, there are many good solutions and the genetic algorithm will find one of them.In extreme cases where there is only one good solution the genetic algorithm may fail, but again it can be restarted by the Active Rules with many chances to find a better solution.

REFERENCES

[1] ”Solving Timetable Scheduling Problem by Using Genetic Algorithms”, Branimir Sigl, Marin Golub, Vedran Mornar Faculty of Electrical Engineering and Computing, University of Zagreb Unska 3, 10000 Zagreb, Croatia

[2] Use of Active Rules and Genetic Algorithm to Generate the Automatic Time-Table.”Bharkha Narang(Asst. Prof.) Ambika Gupta(M.Tech Scholar), Rashmi Bansal(M.Tech Scholar)”Proceedings of 2nd International Conference on Emerging Trends in Engineering and Management, ICETEM 2013.

[3] J.Leon Bambrick Supervisor Dr B Lovell Lecture Timetabling Using Genetic Algorithms Available : http://secretgeek.net/content/bambrilg.pdf