IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

TIMETABLE GENERATOR USING GENETIC ALGORITHM

Prof Archana Salaskar, Rupesh Malla, Anurag Singh, Parmatma Sahani, Mohammad Khorajiya

Associate Professor, Student, Student, Student, Student
Information Technology
[1]Vasantdada Patil Prathishtan College of Engineering, Mumbai, India

Abstract: The proposed solution is an automatic time table generator using a genetic algorithm approach to minimize the time and manpower involved in creating a time table. The system takes various inputs such as the number of subjects, teachers, workload of teachers, semesters, class numbers, and labs, and generates accurate and error-free time tables for teaching faculties. The genetic algorithm approach incorporates a range of strategies to ensure the precision and accuracy of the generated time table. Overall, the proposed solution aims to simplify the time-consuming process of generating a time table for educational institutions.

Index Terms **– real-time scheduling, genetic algorithm, time tabling, constraints**

1. INTRODUCTION

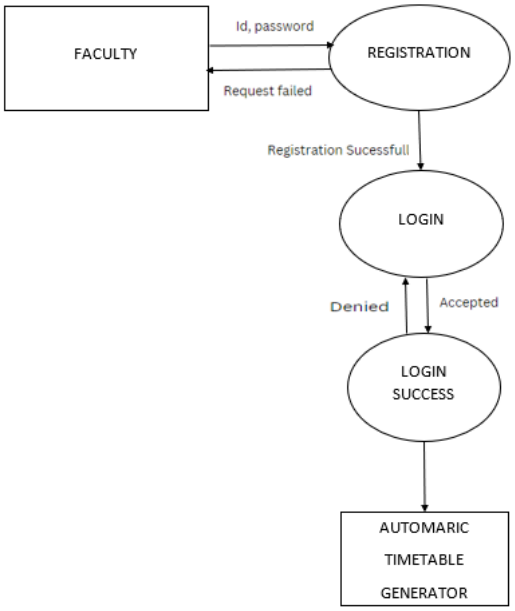
Time Table Scheduling is a problem which is faced by most institutes, using inheritable algorithm can break this problem. A typical scheduling problem that are or can be tedious job in every academic institute formerly or a time. Planning timetable are one of the most complex and prone to error work. In this system to solve it we're using as constraints to solve there are principally two types of constraints, soft and hard constraints. Soft constraints are those if we violate them in scheduling, that is still valid, but hard constraints are those which can't be violated, if we violate them the schedule is no longer valid. doable results are results those which don't violate hard constraints and as well try to satisfy soft constraints. We need to choose the most applicable or the constraints which doesn't violate soft constraints to lesser extent one from doable results. In this design hard- constraints have been taken care rigorously and it has been assured that soft- constraints are as well followed as much as possible. To induce a most effective and easy to use lecture schedule for our institute. This time schedule inputs large data input as schedule. The ideal of the automated schedule generated design was to be developed a tool that enables department to stoutly induce schedule for pupil to pierce directly from the schedule table data which enables the department to present changes to their schedule incontinently tiding public. It enables the department to plan well in advance to design the pupil schedule. It'll also enable the operation to give timely accurate schedule data.

2. RELATED WORK

The existing system formulated scheduling the class to teacher timetabling problem without causing time clashes for the students and the teachers, as well as the resources clashes[1]. Over the past few years, a wide array of techniques have been proposed for solving the course timetabling problem and its' variants. Several systems have been developed for automating the timetable generation [2, 3]. One of the popular techniques is based on graph coloring algorithms [3, 4]. The main disadvantage of this approach is the difficulty of including application specific constraints into the problem formulation. Other methods include artificial neural networks [5], simulated annealing [6], and tabu search [7]. Constraint Based Programming [8] has become an interesting approach for solving timetabling problems recently. This approach combines the declarativity of logic programming with the efficiency of constraint solving. Recently, several researchers have applied evolutionary computation techniques to timetabling problems [9-18]. The problem was formulated using fuzzy logic in [9], and heuristics were used to improve the generalization capabilities. Ref. [10] represented the timetabling problem as a graph, and then manipulated it as a Graph-Coloring Problem, considering the presence of solved cliques in the mating parents. This approach has resulted in considerably smaller execution time. The methods described in [11] transform the large search space into one in which the proportion of feasible solutions is greatly increased. These approaches have established the effectiveness of genetic-based approaches for finding feasible solutions to this highly constrained problem. Hybrid approaches have also been proposed to further improve the quality of the solutions. Each implementation, however, is specific to the academic institution for which it is developed, and addresses a different problem with diverse constraints. Although, the timetabling problem is distinctive for each institution, there exist a set of entities and constraints which are common to every possible instantiation of the timetabling problem. In this paper, we present a model of this common core in terms of a knowledge-augmented evolutionary algorithm approach which may be extended to cover the needs of a specific academic unit. The algorithm incorporates two distinct objectives that concern precisely the minimization of the violations of both types of constraints, hard and soft. The problem is modeled as a bi-objective problem to construct feasible assignments of course modules to lecture rooms on specified timeslots.

3. SYSTEM ARCHITECTURE

Timetabling is one of the common scheduling problems, which can be described as the allocation of resources for tasks under preened constraints so that it maximizes the possibility of allocation or minimizes the violation of constraints. One cannot deny the human errors which will introduce conflicts such as overlapping of resources, double booking of time-slots, etc. So here our proposed system is very effective and simple in this architecture the user or admin or any institute faculty first registers creates login and password and then starts giving input to the system. The systems Algorithm explained in detail further. It shows the element of the system. It also shows the interactions between these elements[1]. The models govern the composition and the constraint of these models. When the system faces a complex problem, the system breaks it down into parts that become easier to solve with simple solutions. Then, by combining all these small solutions, we can find the solution to our complex problem. So, our added feature is it can combine two class as labs using constraints with genetic algorithm.



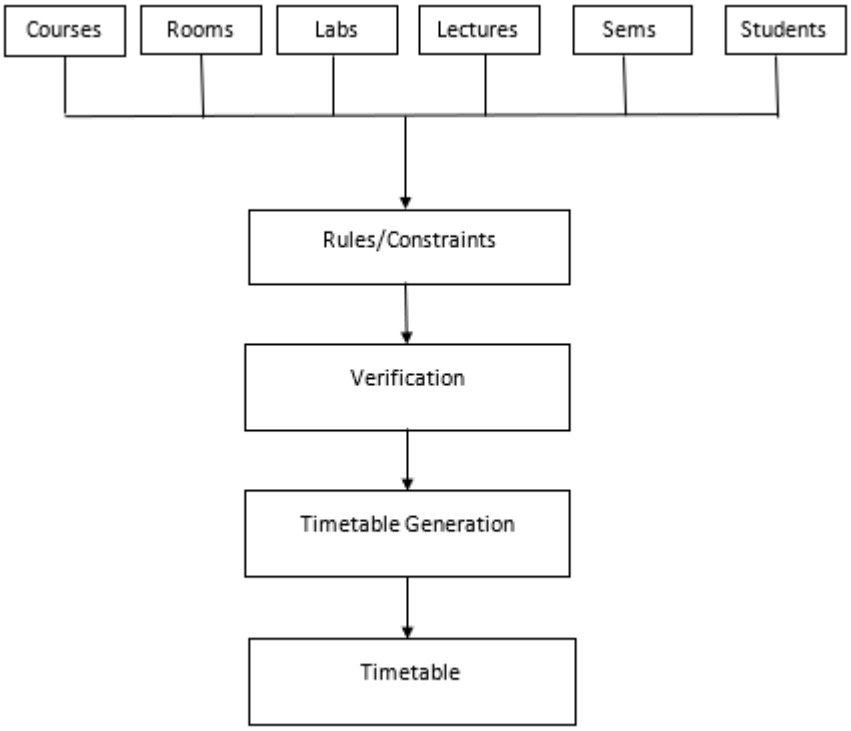
1.1. Block diagram of generator

4. ALGORITHMS

The genetic algorithm starts first with an initial generation of chromosomes is created randomly and their fitness value is analyzed. New Generations are created after this. For every generation, it performs following basic operations:

A) First, it preserves few fittest chromosomes from the previous generation as it is.
B) Then it randomly selects a pair of chromosomes from the previous generations.
C) Perform crossover depending on the crossover rate which is high usually.
D) Perform mutation on the more fit chromosome so obtained depending on the mutation rate which is kept small usually.

Now it analyzes the fitness of the new generation of chromosomes and order them according to fitness values. It repeats creating new generations unless chromosomes of desired fitness value are obtained.



1.2 Diagram of Algorithm

5. CONCLUSIONS

The process of Time Table generating has been fully automated with this software. This web app can now be used by multiple colleges, universities and schools which can rely on it for their Time Table scheduling which earlier had to be done by hand. Moreover, each of the solutions are much better than a manually prepared solution which is in currently use. As we describe how this evolutionary algorithm can change employed to be efficient in time table generating for any institutes or universities.

6. REFERENCES

[1]. Mrunal Gaikwad, Adwait Gaikwad, Mukesh Chaudhary, Dhanshree Sawarkar, Mehul Bhargava; Auto Timetable Generator; e-ISSN: 2582-5208 vol: 04, May 2022

[2]. Farzana Jawale, Renuka Sabale, Dhanashri Kulkarni; Automated TimeTable Generator System; e-ISSN: 2582-5208; vol-03; June 2021.

[3]. Mrunmayee V. Rane, Vikram M. Apte, Vishakha N. Nerkar, Mani Roja Edinburgh, K.Y. Rajput; Automated Timetabling System for University Course; Mar 5-7 2021.

[4]. Shraddha Thakare, Tejal Nikram; Automated Timetable Generation using Genetic Algorithm; vol.9 july 2020.

[5]. Achini Kumari Herath; Genetic Algorithn for university course timetabling problem (2017) Electronic Theses and Dissertations. 443. https://egrove.olemiss.edu/etd/443.

[6]. Downslund K. A., "Simulated annealing solutions for multi-objective scheduling and timetabling", In Modern Heuristic Search Methods. Wiley. Chichester, England, 1996, pp. 155-166.

[7]. Hertz A., "Tabu search for large scale timetabling problems", European Journal of Operations Research, Vol. 54, 1991, pp. 39-47.

[8]. Abbas, A.M.; Tsung, E.P.K., " Constraint-based timetabling-a case study", ACS/IEEE International Conference on Computer Systems and Applications, 2001, Page(s): 67 -72.

[9]. Calogero Di Stefano; Andrea G. B. Tettumunzi, "An Evolutionary Algorithm for Solving the School Time-Tabling Problem", Lecture Notes on Computer Science, LNCS 2037, p. 452.

[10]. Terashimu-Marin, H.; Ross, P.; Valenzuela-Rendon, M., "Cliquebased crossover for solving the timetabling problem with GAS", Proceedings of the 1999 Congress on Evolutionary Computation, CEC 99, pp. 1999-1206 vol. 2.

[11]. Puechter, B.; Cumming, A,; Luchian, H.; Petriuc, M., " Two solutions to the general timetable problem using evolutionary methods", Proceedings of the First IEEE Conference on Computational Intelligence, World Congress on Computational Intelligence, 1994, Page(s): 300 -305 vol. 1.