Automatic Class And Exam Timetable Generator Using Genetic Algorithm

Mrs.R.Radhika[1], M.Jayalakshmi[2], T.Babyshalini[3], M.Santhiya[4], A.Manjupriya[5]
[1]Assistant Professor, Dept of Computer Science and Engineering
[2, 3, 4, 5]Dept of Computer Science and Engineering
[1, 2, 3, 4, 5] Adhi College of Engineering and Technology, Kanchipuram-631 605

Abstract- *The project aims to create an optimal timetable using a genetic algorithm. The program takes user input regarding the number of subjects, the number of classes in a day, number of staffs, details about the staffs and the students and the total number of working days in a week. It generates an initial population of timetables and uses a fitness function to evaluate each chromosome's fitness. The crossover and mutation functions are used to create new chromosomes, and the least fit chromosome is removed from the population. The algorithm continues for a specified number of generations or until it finds an optimal solution. The optimal timetable is then printed, and an Excel file is generated with the timetable. The program provides a solution to the timetable problem that optimizes the use of resources and time. To display the generated timetable on a web page, Flask can be used as a micro web framework in python.*

Keywords- Genetic Algorithm, Scheduling, Optimal Solution, Automated Timetable, Flask Framework.

I. INTRODUCTION

A class timetable is a schedule that outlines the different classes or subjects that students have to attend on specific days and at specific times. It is an essential tool for managing school or college activities and ensures that students attend all classes and cover the necessary curriculum. Class timetables can also help teachers and school administrators to manage resources such as classrooms and teachers effectively. The process of creating a class timetable can be a challenging task, especially for larger institutions with many subjects and students. Hence, the use of computer programs and algorithms to generate optimal timetables has become increasingly popular in recent years. This approach helps to save time and resources while ensuring that the timetable is optimized for the institution's specific needs. The proposed software is used to overcome the entire problem which they are faced and making complete automation of manual system to computerized system. Generating timetables for educational institutions is a challenging task that requires the consideration of multiple constraints and requirements. These constraints and requirements include the availability of teachers and classrooms, avoiding conflicts in scheduling, meeting the necessary constraints of courses, and so on. An automatic timetable generator using a genetic algorithm is proposed to solve this problem. This project presents the design and implementation of the automatic timetable generator using a genetic algorithm.

Exam timetabling is a very important process and one of the common scheduling problems in any educational institution, which can be described as the allocation of resources for tasks under predefined constraints so that it maximizes the possibility of allocation or minimizes the violation of constraints. A key factor in running an educational center or basically an academic environment is the need for a well-planned, well throughout and clash-free timetable. Back in the days when technology was not in wide use, the timetable was manually created by the academic institution. Therefore, the need to adopt an electronic system as opposed to the manual process cannot be overemphasized. Several other administrative sectors of most institutions have been automated, but lecture scheduling is still done manually due to the problems involved. The project timetable concerns all activities with regard to producing a schedule that must be subjective to different constraints. Timetable concerns activities with regards to producing a schedule that must be subjective to the different constraints.

II. RELATED WORK

The literature describes various studies devoted to automated timetable scheduling. For example, When scheduling is modeled, it must cover both hard and soft constraints, i.e. a set of various variables and values - such as halls, subjects, staff, and students - must be assigned in order to satisfy them. Thereby we can conclude the objective of the above method is to minimize the violation of soft constraints and satisfy the hard constraints. In the paper[1],The proposed algorithm can be further adapted to suit as per the requirements of different institutes and universities. It considers a wide range of constraints divided into soft and hard, to generate faculty, classroom/laboratory and student timetables. The complexity of the proposed algorithm is n3. In

[2], The paper presents a categorisation of the methodologies conducted in recent years based on chronology, category and application. [3]We propose two pattern-based formulations and a solution algorithm that simultaneously exploits column generation and a team of metaheuristics to build and improve solutions. [4]Graph Coloring Algorithm (GCA) takes 11 percent of the time, Genetic and Graph Coloring (GCA) takes 25 percent, Heuristic and Iterated Local Search Algorithms (ILSA) take 44 percent of time, and Heuristic Algorithm takes 20 percent. [5]This means developers can use imperative programming for business logic and declarative programming for user interface (UI) development. [6]The fitness score relates to the quantity of crashes the timetable has experienced. The above survey of various researchers of different algorithms of Automated Time-Table Generator trying to develop a software which helps to generate Timetable for an institution automatically. When we look at the existing system, we understand that the schedule generation is done manually. Manual adjustment of the schedule during the absence of any of the faculty, and this is a big challenge for the Automatic schedule generator, which manages the schedule automatically in case of absence of any of the faculty. As mentioned earlier, generating the schedule should take into account the maximum and minimum workload that the college has. Moreover, it is a time-consuming process.

1. Automated Timetabling System for University Course The proposed algorithm can be further adapted to suit as per the requirements of different institutes and universities. It considers a wide range of constraints divided into soft and hard, to generate faculty, classroom/laboratory and student timetables. The complexity of the proposed algorithm is n3. An algorithm has been proved effective for replacing an oldfashioned, time-consuming timetabling system with an enhanced, flexible and automated automated system. The proposed algorithm saves hours of your precious time which can be utilized in other areas of your life.

2. A survey of the state-of-the-art of optimisation methodologies in school timetabling problems The paper presents a categorization of methodologies performed in recent years based on chronology, category and application. The optimization model captures the behavior of the system in all possible scenarios.

3. Pattern-based models and a cooperative parallel metaheuristic for high school timetabling problems We propose two pattern-based formulations and a solution algorithm that simultaneously uses column generation and a metaheuristics team to generate and improve solutions. More modelling power and More readily implementable answers.

4. The Intelligent Scheduler: A Comparison of Genetic Algorithms, Graph Coloring, Heuristic, and Iterated Local Search Algorithms. In this method, timetable problem is dealt as graph problem and then they order the events using domain specific heuristics and then assign the events sequentially into valid time slots, so no constraints are violated for each time slot. Graph Coloring Algorithm (GCA) takes 11% of the time.

5. Development of an Efficient Timetable System using AngularJs and Bootstrap3 This means developers can use imperative programming for business logic and declarative programming for user interface (UI) development. A user friendly GUI is provided with the help of Angular material and Bootstrap. Single page webpage is generated with the help of AngularJS.

III. GENETIC ALGORITHM

A genetic algorithm is a heuristic optimization technique inspired by the process of natural selection. It operates on a population of candidate solutions and evolves them over generations using genetic operators such as crossover, mutation, and selection. The fitness of each solution is evaluated based on a set of objective functions, and the best solutions are selected to produce offspring for next generation. The genetic algorithm module can be implemented as follows:

1- Initialization:

- Initialization refers to the process of creating an initial population of chromosomes.
- In this phase, a set of chromosomes is generated randomly, and each chromosome represents a potential timetable. The number of chromosomes in the initial population can be determined by the user and is typically set based on the size of the problem.
- Each chromosome is represented as a string of genes, where each gene corresponds to a specific class in the timetable. The value of each gene represents the class that is scheduled for that particular time slot.
- The initialization phase is critical to the success of the genetic algorithm as the quality of the initial population can significantly impact the algorithm's

ability to find an optimal solution. Therefore, it is essential to create an initial population that is diverse and representative of the problem space.

2- Fitness:

- The fitness function is used to evaluate the quality of each solution (chromosome) in the population. In the context of class time table preparation, the fitness function assigns a fitness score to each timetable based on how well it satisfies the constraints and requirements of the problem.
- For example, the fitness function may take into account constraints such as,
- Ensuring that no teacher or student is assigned to more than one class at the same time.
- Ensuring that each class has a qualified teacher assigned to it.
- Ensuring that each teacher has a maximum workload limit.
- Ensuring that the number of consecutive free periods is within an acceptable range.
- The fitness function may also incorporate preferences or requirements such as:
- Prioritizing specific subjects or classes that have a higher importance or urgency.
- Prioritizing specific teachers or students based on their availability or needs.
- Maximizing the use of available resources (such as classrooms) to minimize conflicts and overlaps.
- The fitness function assigns a fitness score to each chromosome based on how well it satisfies the constraints and preferences. The higher the fitness score, the better the solution. The fitness function is used by the genetic algorithm to select the most fit chromosomes for crossover and
- mutation, which leads to the creation of new, potentially better solutions in the next generation.

3- Selection:

- Selection is the process of choosing individuals from the population to become parents for the next generation. In genetic algorithms, the selection process is based on fitness, where individuals with higher fitness values are more likely to be selected.
- Random module is used to perform selection of parents for crossover in a genetic algorithm.
- The higher the fitness value of a chromosome, the greater the probability of it being selected as a parent for crossover.

4- Crossover:

- Generating a class timetable using a genetic algorithm, crossover refers to the process of creating a new offspring solution by combining the genetic information of two parent solutions.
- In this particular implementation, the crossover function takes two parent solutions (timetables) as input, and creates a new child solution by randomly selecting a subject from each of the parent timetables for each class period, with equal probability.
- The resulting child timetable inherits genetic information from both parents, and may possess desirable traits from each.

5- Mutation:

- Mutation is a process that is applied to a single chromosome of the population with a certain probability.

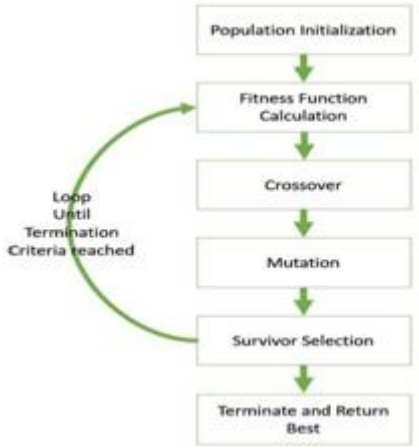

Fig 1- Algorithm Overview

- The mutation function generates a new chromosome by randomly selecting a day, class, and subject and replacing the original subject with a new one.
- This allows the algorithm to explore different solutions in the search space and avoid getting stuck in local optima.

IV. MODULES

The class timetable generator consists of three main modules:

- Data Input Module
- Genetic Algorithm Module, and
- Timetable Output Module.

[1] DATA INPUT MODULE

The Data Input Module collects information such as the number of teachers, classrooms, and courses, as well as their availability and constraints. This module obtains data from spreadsheets or databases.

GENETIC ALGORITHM MODULE

The Genetic Algorithm Module uses a genetic algorithm to create an optimal timetable that satisfies all constraints and requirements. This module initializes a population of chromosomes representing potential timetables and then improves them iteratively through genetic operators such as crossover and mutation. The fitness function evaluates the quality of each solution in the population and selects the most fit chromosomes for the next generation. The selection process is based on fitness, where individuals with higher fitness values are more likely to be selected. The crossover process creates a new child solution by combining genetic information from two parent solutions. The mutation process is applied to a single chromosome with a certain probability and generates a new chromosome by randomly replacing a subject with a new one.

TIMETABLE OUTPUT MODULE

The Timetable Output Module presents the final timetable in a human-readable format, such as an Excel spreadsheet. The Flask framework can be used to display the generated timetable on a web page, using the GET and POST methods.

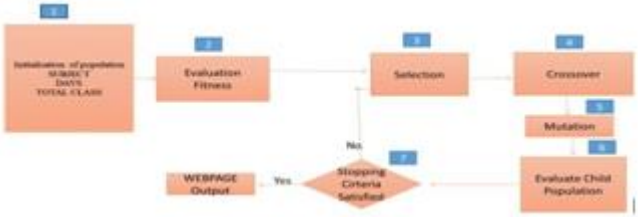

Fig 2- Architecture Overview

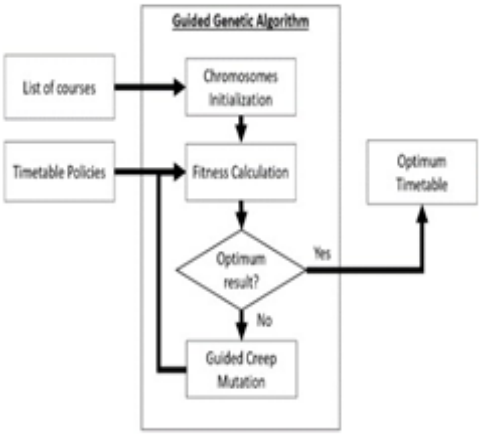

Fig 3- Flow Chart

V. CONCLUSION AND FURTHER WORK

In conclusion, the class and exam timetable generator using a genetic algorithm is a great example of how genetic algorithms can be applied to solve complex optimization problems. The program uses the genetic algorithm to generate the best possible class timetable given a set of constraints such as the number of subjects, number of classes in a day, and the total number of working days in a week. The program uses the fitness function to evaluate the quality of each generated timetable and selects the best ones to generate new timetables through crossover and mutation. The crossover function generates a new timetable by selecting genetic material from two parent timetables, while the mutation function introduces random changes to the current timetable. Thus, we can produce timetable for any number of courses and multiple semesters. This system will help to create dynamic pages so that for implementing such a system we can make use of the different tools are widely applicable and free to use also. This report addresses the procedure of the timetable generation which will be consistently easier with Automated System for Timetable Generation than using spreadsheet manually which might lead to constraints problem that are arduous to establish when timetable is generated manually. The Automated System for Timetable Generation is provided with necessary details of courses, classrooms and faculties which are stored in database (SQL server). The Automated System for Timetable Generation generates timetable based on these details in database with minimum time and satisfies all the constraints. Genetic Algorithm is one of the most effective ways of generating timetables although it does not give a 100% optimal timetable. Its degree of optimality depends on the constraints used and also the fitness function of the parameters. Using Genetic Algorithm is a little slower due to the steps it has to undergo. Generating random numbers, mutating and crossing over parameters before the final result is gotten. Notwithstanding, it still remains the best way to solve timetable issues.

REFERENCES

[1] Sara Ceschia... Andrea Schaerf, Educational timetabling: Problems, bench-marks, and state-of-the-art results (Survey), Jun 2021

[2] Mrunmayee V. Rane,Vikram M. Apte,,Vishakha N. NerkarMani Roja Edinburgh,K.Y. Rajput ”Automated Timetabling System for University Course” March 2021

[3] Joo SiangTan, Say LengGoh, GrahamKendall, Nasser R.Sabard “A survey of the state-of-the-art of optimisation methodologies in school timetabling problems” 1 March 2021

[4] J. Ansel, S. Aravind, "A Survey of Timetable Scheduling Techniques", March 2021
[5] P. V. Panchal, P. N. P. Nair, "A Review of Automated Timetable Generation Techniques and Tools for Higher Education Institutions", Jan 2021
[6] S. A. Al-Tarawneh, A. M. Al-Ani, " A Comprehensive Review of Timetable Scheduling Techniques", Nov 2021
[7] R. K. Singh, V. K. Gupta, " A Comprehensive Review of Timetable Generation Techniques for Educational Institutions" 2021
[8] R. M. Al-Kasasbeh, M. Al-Kasasbeh, " Automated Timetabling: A Review of Techniques and Applications" May 2020
[9] LandirSaviniec, Maristela O.Santos, Alysson M.Costa, Lana M.R. dosSantos "Pattern-based models and a cooperative parallel metaheuristic for high school timetabling problems"1 Feb 2020
[10] Tiny Wijerathna Ekanayake, Pavani Subasinghe, Shawn Ragel, Anjalie Gamage, Suchini Attanayaka," Intelligent Timetable Scheduler: A Comparison of Genetic Graph Coloring, Heuristic and Iterated Local Search Algorithms", December 5-6,2019.
[11] Sundresan A/L Perumal, Mujahid Tabassum, Norita MD Norwani, Ganthan A/L Narayana Samy, Sivanathan A/L Perumal," Development of an Efficient Timetable System using AngularJs and Bootstrap3" 2018
[12] Namrata Bodas , Jhanvi Shah, Yash Shah, Aishwarya Sontakke, Dnyaneshwar Dhangar. "Exam Cell Automation System and Timetable Generator" 2018
[13] A Literature Review on Timetable Generation Algorithms [Tanuja Padale, Harshada Mane] Imperial Journal of Interdisciplinary Research(IJIR) Vol-3, Issue-1, 2018 ISSN: 2454-1362.
[14] P.K.Mahajan , A.K.Verma, "Automated Timetable Generation Techniques", March 2018
[15] S.Singla and A.Juneja, " A Comprehensive Survey Of Timetable generation Techniques", Feb 2018
[16] Automated college timetable genetor [Adithya R Pai, Ashwitha S,Aksha Shetty, Prof.Geethalaxmi] International Journal of Scientific & Engineering Research Volume 9, Issue 4, April- 2018.
[17] M, S., & Pranav Kiran Vaze, P. M. (MAY 2017). Automatic Timetable Generator. International Journal of Advanced Research in Computer Science and Software Engineering.
[18] Priyanka Gore "Timetable Generation Using Ant Colony Optimization Algorithm." Vol. 5, Issue 3, March 2017
[19] Yash Lahoti, Aaditya Punekar, Hiten Patel, Vishal Bhimsariya, " Automated Timetable Generator", Jan 2017
[20] S. V. Deo, A. G. Kulkarni, " A Comprehensive Survey of Timetable Generation Techniques and Their Applications" , 2017