Smartslot: Evolutionary Timetable Design

Dipali Bhalerao	Rekha Shelake	Anuja Magar	Sayali Dumbare	Sucheta Lonare
Lecturer	Lecturer	Lecturer	Lecturer	Lecturer
Department of	Department of	Department of	Department of	Department of
Computer Engineering	Computer Engineering	Computer Engineering	Computer Engineering	Computer Engineering
Jaihind Polytechnic,	Jaihind Polytechnic,	Jaihind Polytechnic,	Jaihind Polytechnic,	Jaihind Polytechnic,
Kuran	Kuran	Kuran	Kuran	Kuran

Abstract- In this study, we propose a system to develop an efficient model for timetable scheduling utilizing a genetic algorithm. The goal of the research is to construct a model leveraging genetic algorithms to effectively address complex combinatorial optimization problems. The algorithm was evaluated on both small and large problem instances. The performance of the algorithm was notably improved through modifications to the fundamental genetic operators, which prevented the generation of new conflicts within individuals. The scheduling solution outlined in this paper is adaptive in nature

Index Terms - genetic algorithms, scheduling

I. INTRODUCTION

This article explores the implementation of a genetic algorithm to solve the timetable scheduling problem. Scheduling timetables is a common issue for educational institutions. The primary aim of the algorithm is to reduce conflicts in the timetable. An additional approach was employed to optimize the search space through encoding techniques.

The paper focuses on the application of genetic algorithms for managing timetables in universities and colleges. The project aims to introduce the concept of genetic algorithms and demonstrate their use in addressing scheduling challenges. The objectives are to explain the principles and functionality of genetic algorithms, analyse the common issues in timetable management, and create a prototype for an automated scheduling system. This prototype employs a genetic algorithm instead of traditional scheduling methods.

While the current scheduling system at Coventry University is deemed adequate, this study does not propose a replacement solution.

II. REVIEW OF LITERATURE

The current system defines the class or teacher timetabling problem by assuming that each lecture involves a single group of students, one teacher, and any number of time slots that can be freely selected. Over time, this problem has been extensively studied under various conditions. Initially, the focus was primarily on schools, where the timetabling challenges are comparatively straightforward due to the simplicity of class structures. In such cases, traditional methods, like linear or integer programming techniques, were effectively applied.\ The increasing focus on cases involving higher secondary schools and universities, which feature more intricate class structures, has significantly added to the complexity of the problem.

III. REQUIREMENT ANALYSIS

1. Functional Requirements Analysis

What it includes: Identifies what the system should do, such as features and behaviours.

Examples for SMARTSLOT:
- Generate conflict-free timetables for institutions.
- Allow input of constraints (e.g., room availability, teacher schedules).
- Adapt to different levels of complexity (schools vs. universities).
- Provide manual adjustment options for admins.

2. Non-Functional Requirements Analysis

What it includes: Defines system qualities like performance, usability, and reliability.

Examples for SMARTSLOT:
- Timetable generation must occur within a specified time limit.
- The system should be scalable to handle large institutions.
- User interface should be intuitive for non-technical users.
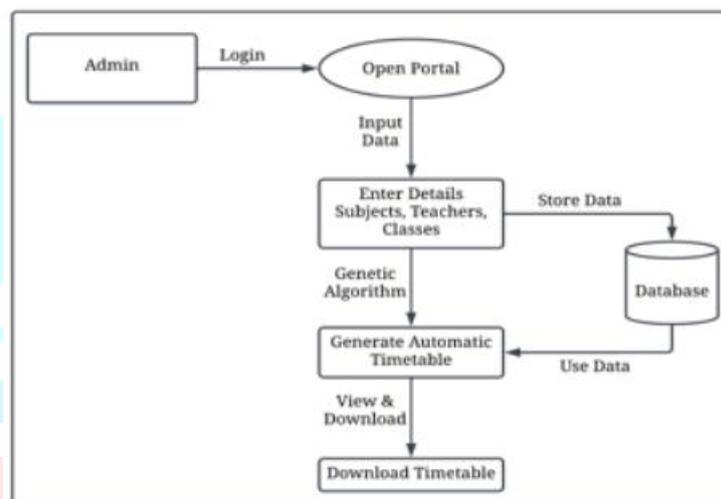
IV. PROPOSED SYSTEM ARCHITECTURE

Fig 1 Proposed Architecture

System Architecture

The architecture can be divided into several components that support the Genetic Algorithm for timetable generation:

- **User Interface (UI)**:
  - A user interface for inputting data about courses, instructors, rooms, and time slots.
  - Provides feedback on the optimization process and displays the generated timetable.
- **Database**:
  - A database stores information about resources (courses, rooms, instructors) and constraints (availability, room capacities, prerequisites).
  - The database is queried to retrieve information needed to initialize the genetic algorithm and store results.
- **Genetic Algorithm Engine**:
  - **Initialization**: The initial population of schedules is created by randomly assigning courses to available time slots and rooms.
  - **GA Operations**: The engine performs selection, crossover, and mutation to evolve better timetables over generations.
  - **Fitness Evaluation**: The engine evaluates each generated timetable using a fitness function that checks the validity and optimization score.
- **Scheduler/Timetable Generator**:
  - Once the genetic algorithm completes, the best solution (schedule) is selected and presented to the user.
  - Can include features such as conflict resolution (rescheduling), optimization suggestions, and manual overrides by administrators.

- **Web Service Layer (Optional)**:
    - o For scalable deployment, the architecture can include a web service layer for remote access to the timetable generation tool, allowing users to interact with the system over the internet.
- **Reports and Analytics**:
    - o Provides the ability to generate reports, such as resource utilization, instructor schedules, room usage, and optimization results.
    - o Allows administrators to analyze and adjust the timetable based on performance metrics (e.g., resource utilization rates).

Advantages of Proposed System

1. **Flexibility**: The system allows for dynamic changes and adjustments to the timetable, unlike manual methods that are rigid and time-consuming to modify.
2. **Easy Data Entry and Revision**: With an intuitive interface, users can easily enter and update data, streamlining the process of timetable creation and ensuring timely revisions without difficulty.
3. **Increased Productivity**: By automating time-consuming tasks, it frees up human resources to focus on other critical activities, leading to a more productive environment.
4. **Elimination of Paperwork**: By digitizing the process, the need for physical documents is significantly reduced, leading to paperless operations and a more environmentally friendly approach.

V. APPLICATION

1. University Timetable Management.
2. School Timetable Management.
3. Any organization timetable management

VI.CONCLUSION

The initial timetabling problem, which involved a large number of binary variables, has been simplified by reducing its size. This was achieved by eliminating some dimensions of the problem and incorporating them into the constraints. By grouping several binary variables into a single gene, the overall problem size was significantly reduced. As a result, it is now feasible to tackle the full-scale problem using a genetic algorithm approach.

In other words, the complexity of the problem has been reduced by consolidating binary variables, allowing the genetic algorithm to efficiently handle the entire scheduling task. This reduction in dimensionality makes it possible to apply a genetic algorithm to generate a more optimal timetable for the full problem.

The intelligent algorithm converges much more quickly than the basic algorithm, making it a more efficient approach for solving the timetabling problem. Its faster convergence allows for quicker progress toward an optimal solution. Additionally, it serves as a strong starting point for fully solving the full-scale problem, providing a good foundation upon which further refinements and optimizations can be made. This makes it particularly effective in addressing complex timetabling challenges within a reasonable timeframe.

VII.REFERENCES

1. Chowdhary A., Kande, P., Dhone, S., Ingle, S., Rushiya, R. And Gawande, D.,2014 Timetable Generation system. International Journal of Computer Science and Mobile Computing, 3(2)
2. Bhatti, U. and Hanif. M. 2010. Validity of Capital Assets Pricing Model.Evidence from KSE-Pakistan.European Journal of Economics, Finance and Administrative Science, 3 (20).
3. Leon Bambrick, "Lecture Timetabling Using Genetic Algorithms" Available:http://secretgeek.net/content/bambrilg.pdf
4. Burke, E.K. and Petrovic, S., 2002. Recent research directions in automated timetabling European Journal of Operational Research, 140(2), pp.266Y280.