IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 8, March 2025
Impact Factor: 7.67

Timetable Scheduling – Promise Algorithm

Sai Krishna Durvasula[1], David Joshua Raj G[2], Santosh Adhvaidh Chemmani[3], Sai Prakash Gaddangi[4], Mr. K. V. Subba Raju[5]
Students, Department of Computer Science & Engineering[1-4]
Associate Professor, Department of Computer Science & Engineering[5]
Maharaj Vijayaram Gajapati Raj College of Engineering (Autonomous), Vizianagaram, India
saikrishnadurvasula@gmail.com, mailtojoshua22@gmail.com, adhvaidh02@gmail.com
gaddangisaiprakash123@gmail.com, kvsubbaraju@mvgrce.edu.in

Abstract*:* *Timetable generation in educational institutions is a complex process that requires careful coordination of faculty availability, lab schedules, and student preferences. Manual scheduling is not only time-consuming but also prone to errors, leading to conflicts such as overlapping classes and faculty assignment clashes. This project introduces an automated timetable generation system that utilizes a custom-developed algorithm to optimize scheduling while considering various constraints, including faculty availability and institutional requirements. The system enhances accuracy, minimizes manual effort, and allows for real-time adjustments, ensuring operational efficiency. With its scalability and adaptability, the proposed solution provides a well-balanced and optimized schedule suitable for institutions of all sizes.*

Keywords: Institutional policies, faculty availability, lab schedules, student elective preferences, dynamic updates, optimization algorithm, database management, scalable structure, Promise Algorithm, data security and privacy, User Acceptance Testing.

I. INTRODUCTION

Creating an optimized timetable is a complex yet essential task for educational institutions, involving the scheduling of courses, faculty, and resources while managing various constraints. Manual scheduling methods are time-consuming, error-prone, and inefficient, often leading to conflicts such as overlapping classes and faculty assignment clashes. Automated timetable generation systems use algorithms, database management, and optimization techniques to create conflict-free schedules, improving accuracy and efficiency. These systems enable real-time updates, scalability, and adaptability to unforeseen changes, making them crucial for modern educational institutions.

A. Understanding the Complexity of Timetable Generation

Timetable generation involves managing multiple departments, faculty, and laboratories while preventing scheduling conflicts. Traditional manual methods often result in inefficiencies like overlapping faculty schedules and simultaneous student courses, negatively affecting the academic experience. With the increasing number of students and courses, a dynamic, automated system is necessary to handle constraints such as faculty teaching hours, lab availability, and departmental needs while optimizing resource utilization.

B. Driving Forces Behind Automation

The primary motivation for automating timetable generation is to eliminate inefficiencies in manual scheduling. Manual processes are prone to human errors, faculty workload imbalances, and resource underutilization. Sudden changes, such as faculty absences or room reallocations, make manual modifications highly complex. Automation reduces administrative effort, enhances accuracy, and enables real-time adjustments, allowing institutions to respond swiftly to changes. As educational institutions grow, automation provides a scalable solution to timetable management.

C. Key Objectives

- Automated Timetable Generation – Develop an algorithm to generate timetables considering constraints like faculty and section availability.
- Conflict-Free Scheduling – Ensure no overlapping class times or faculty double assignments.
- Dynamic and Real-Time Adjustments – Enable automatic updates to schedules based on changes without disrupting the overall timetable.
- User-Friendly Interface – Provide a GUI for easy constraint input, timetable visualization, and manual adjustments.
- Optimization and Resource Management – Maximize faculty and lab utilization while preventing overbooking.
- Scalability and Flexibility – Design the system to handle an increasing number of students, faculty, and courses.

D. Defining the Scope of the System

- Departmental Scheduling – Generate department-specific timetables based on unique constraints.
- Faculty Scheduling – Ensure fair workload distribution and accommodate faculty preferences.
- Lab and Resource Allocation – Avoid double-booking of labs and ensure sequential scheduling.
- Dynamic Updates – Enable real-time timetable modifications
- User Interface and Reporting – Develop an intuitive interface for timetable management and reporting.
- Cross-Department Coordination – Manage shared resources efficiently.
- Scalability – Ensure adaptability for growing institutions and semester-based scheduling.

II. LITERATURE SURVEY

A. Manual Timetable Scheduling

Timetable generation is a critical optimization challenge in educational institutions, requiring efficient allocation of lectures, labs, and faculty while satisfying multiple constraints. Traditional manual scheduling methods, though offering control, are inefficient, error-prone, and difficult to scale, necessitating the adoption of automated solutions [1].

B. Automated Approaches to Timetable Generation

1. Genetic Algorithms (GAs) and Heuristic Methods

Genetic Algorithms (GAs) optimize schedules using evolutionary principles but may face slow convergence on large datasets [2]. Heuristic techniques, including greedy algorithms and graph colouring, generate schedules efficiently but struggle with adaptability to real-time changes [2].

2. Constraint Satisfaction Problem (CSP) Techniques

CSP methods use systematic search strategies like backtracking and arc consistency to resolve conflicts, effectively handling complex constraints [2]. However, they require significant computational power, impacting scalability.

3. Web-Based and AI-Driven Solutions

Recent AI-powered timetable systems integrate multiple optimization methods to automate scheduling. Web-based platforms enhance usability by integrating databases and real-time updates [3]. Open-source projects, such as the Automated Timetable Generator, demonstrate real-world applications, featuring database integration and user-friendly interfaces [4].

C. Challenges in Existing Systems

- Limited real-time adaptability, making last-minute adjustments difficult [3].

- Inadequate comprehensive constraint handling, as most approaches focus on specific constraints rather than holistic optimization [2].
- Complexity in user interfaces, restricting usability for administrators [1].
- Scalability concerns, with many systems struggling to manage large student populations and course structures efficiently [4].

III. PROBLEM STATEMENT

Manual timetable generation in educational institutions is a highly complex and time-consuming task. It involves multiple stakeholders, including faculty, students, and administrators, and must take into account a wide range of constraints such as lab availability, course prerequisites, faculty schedules, and student preferences. Traditional manual methods are inefficient and prone to errors, leading to scheduling conflicts, underutilized resources, and reduced overall efficiency.

The key problem being addressed here is the inefficiency and inaccuracy of manually generated timetables in college settings. The aim is to automate the timetable generation process, ensuring that all constraints are met while minimizing scheduling conflicts and maximizing the use of available resources.

By automating the process, we will have a significant impact on the institution's efficiency, enabling it to generate accurate, conflict-free timetables quickly and with minimal human intervention. Additionally, the system will be adaptable to real-time changes, allowing for smooth and efficient adjustments as needed.

IV. REQURIEMENT GATHERING

A. Functional Requirements

- Automated Timetable Generation: The system must generate conflict-free schedules for multiple departments, considering faculty availability, lab assignments, and course prerequisites.
- Constraint Handling: It must accommodate faculty availability, lab allotments, prerequisite sequencing, and timing differences between lectures and labs.
- Real-Time Adaptability: The system should dynamically adjust schedules in response to faculty unavailability or other changes without significant manual intervention.
- Customizable Scheduling: Users should have the ability to manually override or adjust schedules based on last-minute requirements.

B. Non-Functional Requirements

- Usability: The system should have an intuitive interface, enabling administrative staff to easily generate and modify timetables.
- Scalability: It must support growing institutions by handling increasing departments, courses, and faculty efficiently.
- Performance: The system should generate schedules quickly, even with complex constraints, ensuring minimal processing delays.
- Security: Role-based access control should be implemented to protect sensitive data, ensuring only authorized personnel can access and modify schedules.

V. PROPOSED WORK

A. Solution Approach

The Automated Timetable Generation System will use a custom-developed "Promise Algorithm" to generate conflict-free schedules while considering faculty availability, lab constraints, and subject dependencies. The system will feature an intuitive user interface, enabling administrators to efficiently create and modify timetables. It will also support real-time adaptability, automatically reallocating classes when necessary.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 8, March 2025
Impact Factor: 7.67

B. Process Flow

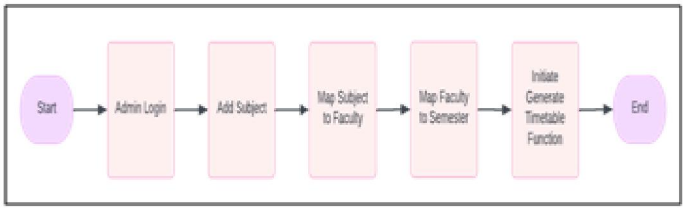

Fig. 1 Flow Diagram

- Admin Login: Administrators log in to manage faculty, subjects, and semester mappings before generating a timetable. The system assigns time slots automatically, ensuring conflict-free scheduling.
- User Login (Faculty & Students): Faculty can view their teaching schedules, and students can access their assigned timetables. Real-time updates ensure accuracy.
- Web Framework Flow: Role-based authentication ensures that admins manage data, while faculty and students have controlled access to relevant schedules.

C. Enhanced Entity Relationship Diagram

The EER diagram outlines the relationships between key entities, including faculty, subjects, semesters, and schedules, forming the database schema for the project.

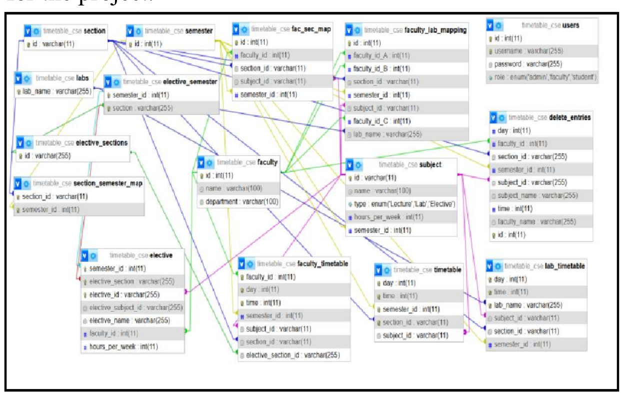

Fig. 2 EER Diagram

VI. SYSTEM ARCHITECTURE

A. Promise Algorithm

The Promise Algorithm is a systematic approach to automated timetable generation, ensuring efficiency, fairness, and conflict-free scheduling. It follows a prioritized sequence to handle different academic sessions effectively.

- Lab Period Allocation: Lab periods are scheduled first due to limited availability and faculty specialization. The algorithm checks faculty, class, and lab availability to assign slots efficiently without conflicts.
- Elective Period Scheduling: Electives are grouped based on faculty assignments and student enrolments, then distributed across available days to prevent scheduling conflicts for faculty handling multiple electives.
- Lecture Period Assignment: Lecture slots are assigned last, ensuring an even subject distribution throughout the week while balancing faculty workload and student schedules.
- The algorithm is adaptive, dynamically adjusting schedules to resolve conflicts. If unresolved, conflicts are logged for manual adjustments, optimizing accuracy and resource utilization

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 8, March 2025
Impact Factor: 7.67

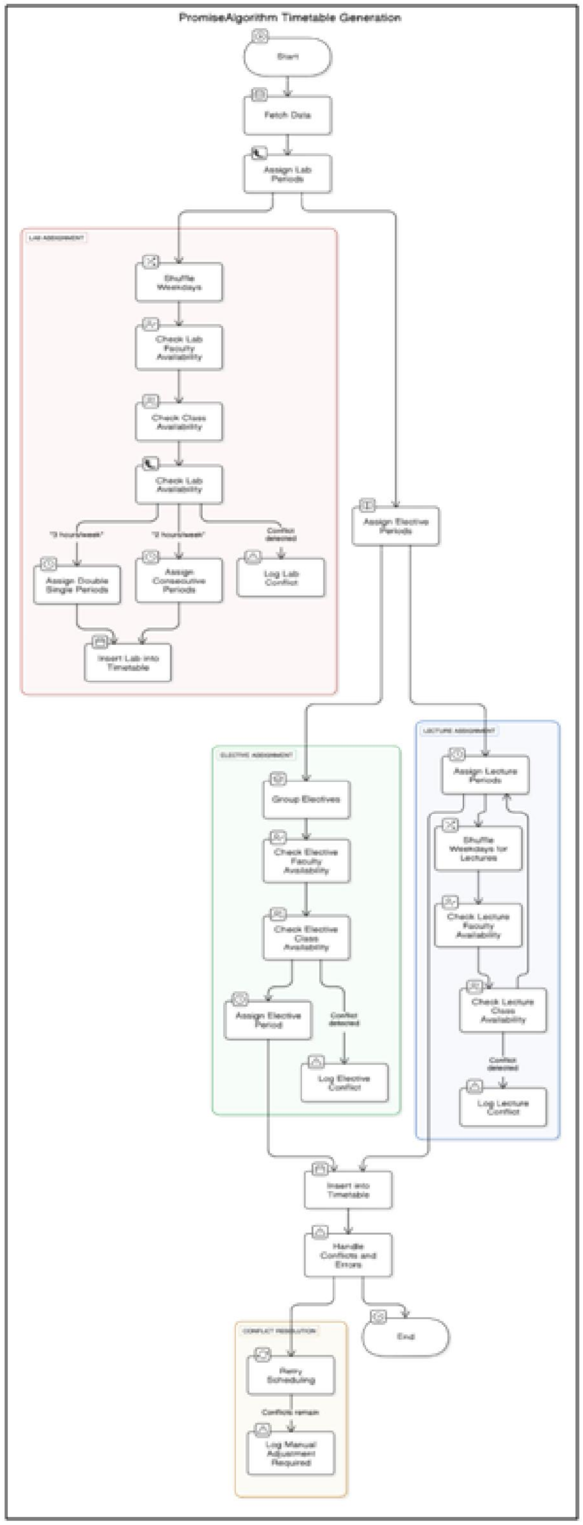

Fig. 3 Flowchart of Promise Algorithm

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 8, March 2025
Impact Factor: 7.67

VII. RESULTS

The system successfully achieved its primary objectives by generating error-free timetables without scheduling conflicts, dynamically adjusting schedules in real-time to accommodate last-minute changes, and providing an intuitive user interface for administrators, faculty, and students. Key observations from user feedback highlighted the system's rapid updates and ease of use. While performance remained stable under normal workloads, further optimization was required for extreme stress testing scenarios.

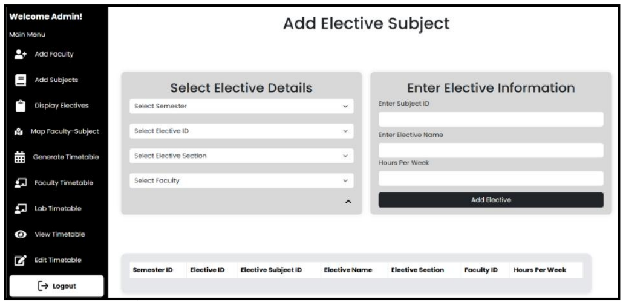

Fig. 4 Add Elective

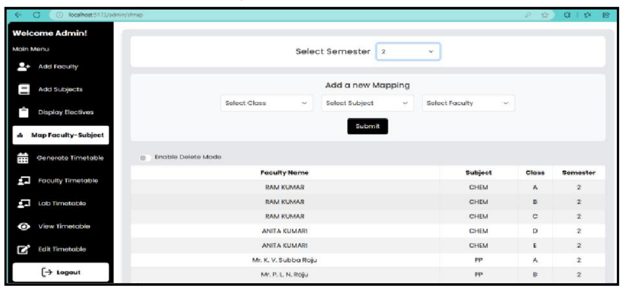

Fig. 5 Map Faculty - Subject

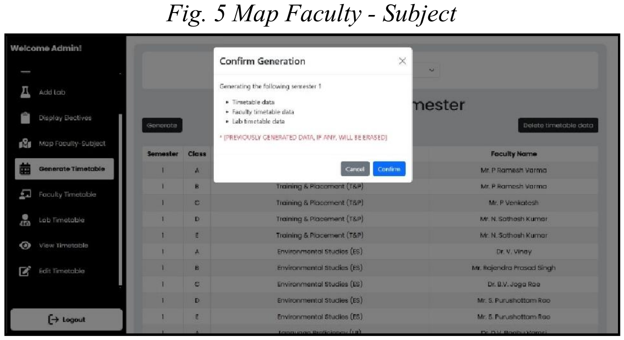

Fig. 6 Generate Timetable

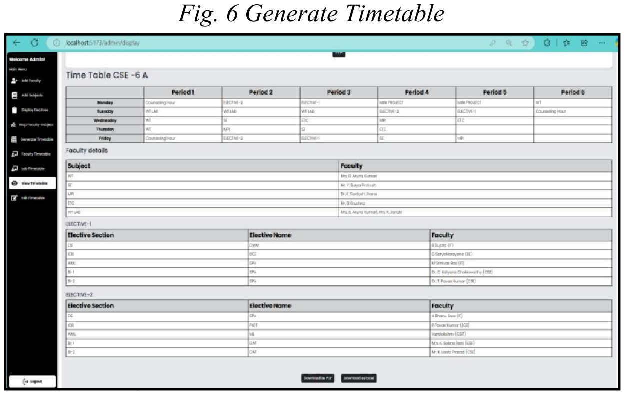

Fig. 7 Display Timetable

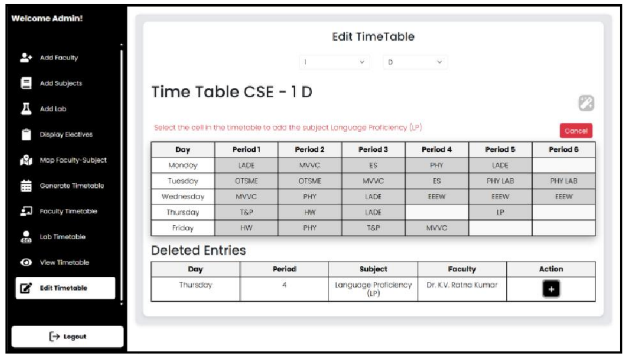

Fig. 8 Edit Timetable

VIII. CONCLUSION

The system successfully automates timetable generation, prevents scheduling conflicts, and adapts to real-time modifications. Performance testing validated its efficiency and accuracy, while user feedback highlighted its usability and adaptability. The modular design ensures flexibility for future enhancements.

A. Future improvements include:

- AI-Powered Scheduling: Leveraging AI for intelligent timetable optimization.
- Automated Notifications: Implementing automated notifications for faculty and students.
- Multi-Department Support: Expanding the system to support multiple departments.
- Enhanced Security: Strengthening data integrity and access control through advanced security measures.

By addressing these areas, the system can achieve greater scalability, efficiency, and user satisfaction, making it a robust solution for educational institutions of all sizes.

REFERENCES

[1]. Jadhav, A., et al., Comparative Analysis of Manual vs. Automated Timetable Generation, International Journal of Computer Science and Mobile Computing (IJCSMC), 2024. A Comparative Analysis of Manual and Automatic Timetabling Approaches for Resource Utilisation in Tertiary Higher Learning Institution [https://ijcsmc.com/docs/papers/December2024/V13I12202407.pdf]

[2]. Patil, S., et al., Optimization Techniques for Timetable Scheduling, International Journal of Creative Research Thoughts (IJCRT), 2023 IJCRT2310497.pdf [https://ijcrt.org/papers/IJCRT2310497.pdf]

[3]. Kumar, R., et al., A Web-Based Approach to Automated Timetable Generation, IEEE Xplore, 2024. [https://ieeexplore.ieee.org/document/10449048]

[4]. Gatkal, A., Automated Timetable Generator, GitHub Repository, 2024. [https://github.com/amitgatkal2530/Automated-Timetable-Generator-?tab=readme-ov-file]