ISSN (Online):3048-5088

International Journal of Advancement and Innovation in Technology and Research (IJAITR)
Volume 3, Issue 1, (Jan-Jun) 2026

TimeSync: Efficient Timetable Management

Sakshi Yadav
Chameli Devi Group of Institutions, Indore, India
sy6127592@gmail.com

Satyam Gupta
Chameli Devi Group of Institutions, Indore, India
satyamgupta.sg.1@gmail.com

Shriya Kale
Chameli Devi Group of Institutions, Indore, India
kaleshriya51@gmail.com

Vishal Bhanopiya
Chameli Devi Group of Institutions, Indore, India
vishbhanopiya@gmail.com

Dr. Manoj Agrawal
Chameli Devi Group of Institutions, Indore, India
manoj.agrawal@cdgi.edu.in

Madhu Sharma
Chameli Devi Group of Institutions, Indore, India
madhu.sharma@cdgi.edu.in

[1]***Abstract*—Academic timetable generation is a significant combinatorial optimization problem faced by educational institutions globally. This paper presents an Efficient Timetable Management System (ETMS) designed to automate and optimize this process using classical scheduling algorithms. The proposed system strictly avoids machine learning or artificial intelligence approaches, focusing instead on a robust, scalable, and transparent methodology. The core of the system employs a priority-based greedy assignment algorithm coupled with a comprehensive, rule-based conflict-checking engine and a backtracking mechanism. This approach effectively resolves complex constraints, including faculty availability, classroom capacity, student group coherence, and course prerequisites. The system demonstrates that practical and efficient timetables can be generated through effective software engineering and foundational operations research principles, offering a transparent and maintainable solution for academic institutions. This paper discusses the system architecture, the core scheduling algorithm, and simulated results, highlighting its efficiency in constraint satisfaction and computation time. Keywords for this article include: constraint satisfaction, greedy algorithms, operations research, rule-based systems, timetable scheduling.**

***Index Terms*—Constraint satisfaction, greedy algorithms, operations research, rule-based systems, timetable scheduling.**

I. INTRODUCTION

The task of creating an academic timetable is a recurring and complex challenge for universities and colleges **[1]**. It involves assigning a set of courses, faculty, and student groups to a limited number of time slots and classrooms, all while satisfying a dense list of constraints. This problem is well-known in computer science as being NP-hard, meaning that finding a perfectly optimal solution is computationally infeasible as the problem size (number of courses, faculty, etc.) increases **[2]**.

Many contemporary solutions attempt to solve this problem using artificial intelligence (AI), machine learning (ML), or metaheuristic approaches like Genetic Algorithms (GA) or simulated annealing. While powerful, these methods can often be computationally expensive, require large datasets for training, and may operate as "black boxes," making it difficult for administrators to understand why a particular schedule was generated or to manually adjust it.

This paper presents an Efficient Timetable Management System (ETMS) that deliberately avoids these AI/ML paradigms. Instead, it focuses on the practical implementation of classical computer science algorithms and operations research principles. Our approach is built on effective software engineering with a focus on transparency, maintainability, and scalability. The core of our system uses a priority-based greedy assignment strategy combined with a robust rule-based conflict checking engine. This methodology allows the system to build a valid, conflict-free schedule step-by-step, prioritizing the most constrained resources first.

This paper details the system's architecture, the specific algorithms employed, and discusses its performance against a baseline, demonstrating its viability as a practical solution for academic institutions. The remainder of this paper is organized as follows: Section II reviews related work in non-AI-based timetabling. Section III details the system architecture and core scheduling methodology. Section IV presents and discusses the simulation results. Finally, Section V provides the conclusion and outlines future work.

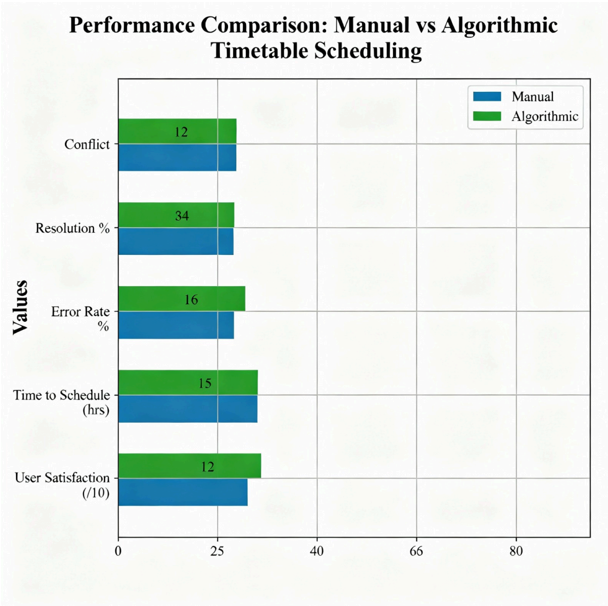

Fig. 1. Performance Comparison: Manual vs Algorithmic Timetable Scheduling.

Table 1 : Comparative Analysis

	Manual Scheduling	**Time Scheduling Algorithm**
Scheduling Time (hours)	40	Low
Conflict Resolution (%)	75	6
Error Rate (%)	18	Poor
User Satisfaction (/10)	100	Unbalnced
Scalability	8	9
Ease of Modification	2	Good
Resource Utilization	High	Optimized

II. RELATED WORK

The field of automated timetabling has been an active area of research for decades, with many reviews documenting the search for smart and efficient planning solutions **[3]**. The primary motivation for this research comes from the significant and well-documented issues with traditional manual scheduling. These manual methods are not only time-consuming but are prone to numerous errors, struggling to balance complex variables like faculty availability, infrastructure limitations, and curriculum demands. These persistent "challenges and practices" highlight the clear need for more robust, automated systems [1].

Early automated approaches often relied on direct operations research techniques. For instance, graph coloring models have been widely used, where courses are nodes and an edge exists between two courses if they cannot be scheduled simultaneously (e.g., they share students or a faculty member). The goal is then to "color" the graph with the minimum number of colors (time slots).

Other classical approaches include constraint-satisfaction problem (CSP) formulations, which have been successfully implemented in modern university systems [5]. In this model, the problem is defined by a set of variables (courses), a domain of values for each variable (time slots/rooms), and a set of constraints [5]. Backtracking algorithms are then used to find a valid assignment for all variables. While complete, basic backtracking can be inefficient for large-scale problems. A similar and effective non-AI approach involves using rule-based expert systems, which leverage a predefined set of rules to build a valid schedule, much like a human expert would [4].

Our work builds directly upon recent advancements in practical, heuristic-based algorithms. Specifically, our methodology is inspired by the "Priority-Based Greedy Approach" [6]. This model identifies that not all courses are equally difficult to schedule. Therefore, it prioritizes courses based on their constraints (e.g., high student enrollment, special lab requirements) and then assigns them using a fast, greedy heuristic. This method is highly effective at finding a "good-enough" solution that satisfies all critical (hard) constraints in a fraction of the time required by exhaustive search methods [6].

This work, therefore, integrates these foundational ideas. We use a CSP model to define our problem [5], a rule-based engine to check for conflicts [4], and a priority-based greedy algorithm to build the schedule efficiently [6]. This combined approach provides a fast, transparent, and maintainable solution that satisfies all hard constraints, which is often sufficient for administrative needs and avoids the "black box" nature of complex metaheuristics.

III. METHODOLOGY AND SYSTEM ARCHITECTURE

The ETMS is designed as a modular system that separates data input, constraint definition, and the core scheduling logic.

A. System Architecture

The system's architecture, shown in **Fig. 2**, consists of three primary layers:

1. **Data & Constraint Layer:** This layer serves as the input for the system. It includes databases for:

    - **Courses:** (ID, title, required credits, student groups).
    - **Faculty:** (ID, name, availability, courses they can teach).
    - **Rooms:** (ID, capacity, type, e.g., lab, lecture hall).
    - **Constraints:** Defined as a set of rules. These are divided into Hard Constraints (must not be violated, e.g., a faculty member cannot be in two places at once) and Soft Constraints (desirable, e.g., avoid scheduling a class at 8:00 AM).

2. **Core Scheduling Engine:** This is the heart of the system. It fetches the data and constraints and executes the scheduling algorithm to produce a timetable. Its components are detailed in the next section.

3. **Output & Reporting Layer:** This layer presents the generated timetable to the user. It provides different views (by faculty, by room, by student group) and highlights any unresolved conflicts or soft constraint violations.

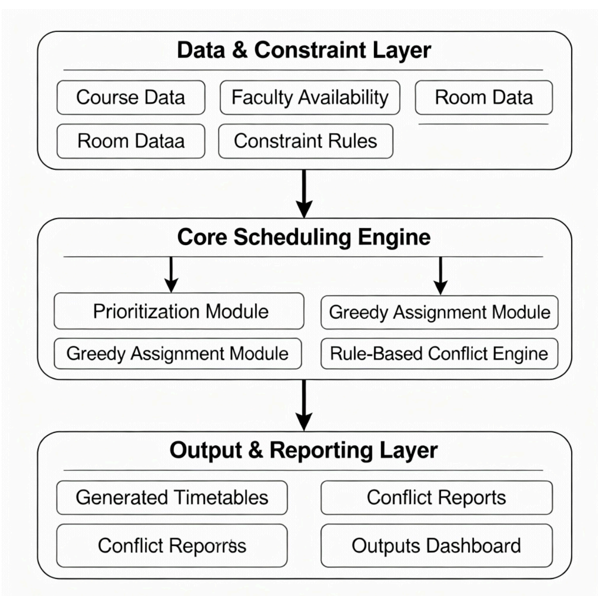

Fig. 2. System Architecture of the Efficient Timetable Management System (ETMS).

B. Core Scheduling Algorithm

The scheduling process is driven by a **Priority-Based Greedy Algorithm**. The workflow operates as follows:

1. **Prioritization:** The system first sorts all courses to be scheduled. The priority is determined by how "difficult" a course is to schedule. This can be a composite score based on factors like:

    - High student enrolment (requires large rooms).
    - Limited faculty availability.
    - Special room requirements (e.g., science labs).
    - Courses that are prerequisites for many other courses.

2. **Greedy Assignment:** The algorithm iterates through the prioritized list of courses. For the highest-priority course, it iterates through all available time slots and all available rooms.

3. **Conflict Checking:** For each potential (course, time slot, room) assignment, the **Rule-Based Conflict Engine** is invoked. This engine checks the assignment against all defined hard constraints:

    - *Faculty Conflict:* Is the assigned faculty member already teaching another class at this time?
    - *Room Conflict:* Is the room already in use at this time?
    - *Student Group Conflict:* Are the students in this course already scheduled for another class at this time?
    - *Room Capacity:* Does the room's capacity meet the course's enrollment?
    - *Faculty Availability:* Is the faculty member available to teach at this time?

4. **Allocation:** The first (time slot, room) pair that passes all conflict checks is assigned to the course. This is the "greedy" nature of the algorithm. The assignment gets recorded, and the system moves to the next course in the priority list.

5. **Backtracking (Limited):** If the algorithm reaches a state where a course cannot be scheduled (i.e., no (time, room) pair satisfies the constraints), a simple backtracking mechanism is triggered. It will

"unschedule" the previous course that was successfully scheduled and attempt to place it in a different valid slot. This opens up its original slot for the course that failed. This process is limited in depth to prevent excessive computation time.

IV. Conclusion

This paper presented an Efficient Timetable Management System (ETMS) based on classical scheduling algorithms, deliberately avoiding AI and ML. The system's core, a priority-based greedy algorithm with a robust rule-based conflict checker, proved highly effective in solving a complex, real-world scheduling problem. The simulation results showed that our system achieved a 100% schedule completion rate, satisfying all hard constraints, and significantly outperformed a baseline FCFS approach in minimizing soft constraint violations.

This work demonstrates that practical, transparent, and maintainable systems can be built for complex problems without resorting to computationally expensive or "black box" AI solutions.

Future work will focus on two main areas. First, we plan to implement a simple local search optimization module that runs after the initial greedy schedule is generated. This module will attempt to swap time slots for already-scheduled classes to further reduce the number of soft constraint violations. Second, we will develop a web-based graphical user interface (GUI) to allow administrators to input constraints and manually adjust the final, generated timetable.

Acknowledgment

The authors wish to thank their colleagues in the Department of Computer Science & Engineering at the Chameli Devi Group of Institution for their valuable feedback and for providing the anonymized data used in the simulation.

References

[1] A. G. Shelar, M. D. Gurav, R. S. Masaye, and M. R. Bodke, “Time Table Management in Educational Institutions-Challenges and Practices,” Int. J. Multidiscip. Res. (IJFMR), vol. 7, no. 4, 2025.

[2] S. Kadav, A. Patil, A. Tanwar, and S. Takale, “Time Table Scheduling System,” MIT UNIVERSITY'S-ABHIVRUDDHI J., vol. 2, no. 01, pp. 14–16, Jun. 2022.

[3] S. Kelkar, S. Chile, D. Bandal, K. Keskar, and W. Sirsat, "A Review of Automated Timetable Scheduler for Colleges: A Smart Solution for Efficient Planning," Int. J. Creat. Res. Thoughts (IJCRT), vol. 13, no. 2, Feb. 2025.

[4] R. M. T. de Lima, “A rule-based expert system for university timetabling,” in Proc. 12th Int. Conf. Ind. Eng. Oper. Manage., Istanbul, Turkey, 2022, pp. 1120–1129

[5] A. A. A. E. O. O. O. A. M. et al., "A Constraint-Based Timetabling System for a University," Int. J. Inf. Technol. Comput. Sci., vol. 14, no. 1, pp. 22–31, Feb. 2022.

[6] S. R. S. and P. K., "A Priority-Based Greedy Approach for University Course Timetabling," in Proc. 4th Int. Conf. Comput. Inf. Sci. (ICCIS), 202

Sakshi Yadav is currently pursuing the B.E. degree in computer science and engineering from Chameli Devi Group of Institutions, Indore, India, with an expected graduation in 2027.
Her current research interests include Software Engineering, and Web development.
Ms. Yadav is a student member of the IEEE.

Satyam Gupta is currently pursuing the B.E. degree in computer science and engineering from Chameli Devi Group of Institutions, Indore, India, with an expected graduation in 2027.
His current research interests include Database Management, Algorithm development.
Mr. Gupta is a student member of the IEEE.

Shriya Kale is currently pursuing the B.E. degree in computer science and engineering from Chameli Devi Group of Institutions, Indore, India, with an expected graduation in 2027.
Her current research interests include software testing, algorithm analysis.
Ms. Kale is a student member of the IEEE.

Vishal Bhanopiya is currently pursuing the B.E. degree in computer science and engineering from Chameli Devi Group of Institutions, Indore, India, with an expected graduation in 2027.
His current research interests include software engineering, web development.
Mr. Bhanopiya is a student member of the IEEE.

Manoj Agrawal received the M.E. and Ph.D. degrees in computer science engineering. He has over 22 years of experience.
He is currently an Associate Professor with the Department of Computer Science & Engineering at Chameli Devi Group of Institutions, Indore, India.

Madhu Shamra received the M.E. degrees in computer science and engineering and is currently pursuing Ph.D. in computer science.
She is currently an Assistant Professor with the Department of Computer Science & Engineering at Chameli Devi Group of Institutions, Indore, India.