International Journal of Scientific Research and Engineering Development-– Volume 9 Issue 2, Mar-Apr 2026
Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

CHRONO CLASS (Time Table Generator)

Dr.K.Karuppasamy
Head of the Department
Dept of Computer Science & Engineering
RVS College of Engineering & Technology,
Coimbatore, India.
karuppusamyrvs@gmail.com

R.Thenmalar
Assistant Professor
Dept of Computer Science & Engineering
RVS College of Engineering & Technology,
Coimbatore, India.
thenmalarcbe@gmail.com

I .Diviyadeepa
712822104007
Dept of Computer Science & Engineering
RVS College of Engineering & Technology,
Coimbatore, India.
diviyadeepa2004@gmail.com

S.Dharaneesh
712822104006
Dept of Computer Science & Engineering
RVS College of Engineering & Technology,
Coimbatore, India.
dharaneeshdharaneesh97@gmail.com

C.Jeevanashri
712822104702
Dept of Computer Science & Engineering
RVS College of Engineering & Technology,
Coimbatore, India.
jj879954@gmail.com

V.Chandrasekar
712822104004
Dept of Computer Science & Engineering
RVS College of Engineering & Technology,
Coimbatore, India.
Vschandrasekar2004@gmail.com

Abstract-- The Chrono Class project focuses on automating the timetable generation process using metaheuristic optimization techniques. The system collects input data such as teachers, subjects, rooms, and time slots, and processes them using OptaPlanner, which applies algorithms like Genetic Algorithm and Simulated Annealing. The optimized timetable is stored in a relational database and exposed via REST APIs. This approach significantly reduces conflicts, improves resource utilization, and minimizes manual effort. to automate the timetable generation process .to eliminate teacher, room, and time conflicts. To optimize time and resource utilization to apply metaheuristic algorithms for scheduling. To provide a scalable REST-based system. To reduce manual workload and human errors.

1. Introduction

Timetable generation is one of the most important administrative activities in educational institutions, as it directly influences the smooth functioning of academic schedules. A timetable must efficiently allocate subjects, faculty members, classrooms, and time slots while ensuring that all academic constraints are satisfied. In many institutions, this process is still carried out manually, making it time-consuming, labor-intensive, and highly prone to human errors. As the number of courses, faculty, and students increases, the complexity of timetable preparation grows significantly, making manual methods inefficient and difficult to manage.

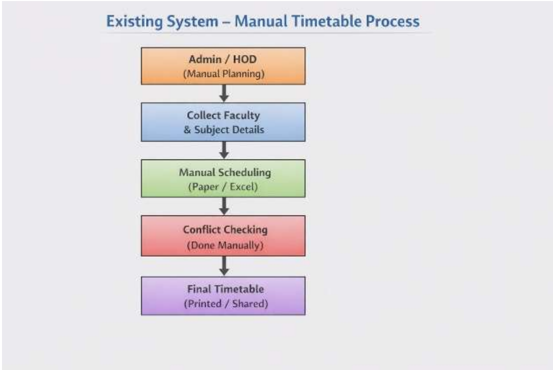
Manual timetable scheduling often results in several challenges such as overlapping classes, improper room allocation, uneven workload distribution among faculty members, and conflicts in faculty availability. Even a small modification, such as a faculty leave or classroom change, requires the timetable to be revised manually, which consumes considerable time and effort. These issues not only increase the workload of administrators but also affect the academic experience of both students and faculty members. Therefore, there is a strong need for an automated and intelligent system that can generate accurate and conflict-free timetables.

Chrono Class is an AI-based automated timetable generation system designed to address these challenges by using optimization techniques. The system aims to simplify and automate the scheduling process while ensuring optimal utilization of institutional resources. By leveraging modern technologies and intelligent algorithms, Chrono Class generates efficient timetables that satisfy both hard constraints (such as faculty availability and room capacity) and soft constraints (such as workload balance and preferred time slots).

The core of the Chrono Class system is powered by **OptaPlanner**, an open-source optimization engine that applies metaheuristic algorithms such as **Genetic Algorithm** and **Simulated Annealing**. These algorithms explore multiple scheduling possibilities and select the most optimal solution based on predefined constraints. This approach allows the system to handle complex scheduling problems effectively, even in large educational institutions with multiple departments and courses.

Chrono Class follows a modular and scalable architecture built using **Java**, **Spring Boot**, **REST APIs**, and a **relational database**. Institutional data such as subjects, faculty details, classrooms, and time slots are collected and stored securely in the database. The optimization engine processes this data to generate a conflict-free timetable, which can be accessed by different users such as administrators, heads of departments, faculty members, and students based on role-based permissions.

By automating the timetable generation process, Chrono Class significantly reduces manual workload, minimizes scheduling conflicts, and improves overall academic planning efficiency. The system provides flexibility for future enhancements and can easily adapt to changes in academic requirements. As a result, Chrono Class serves as a reliable, efficient, and intelligent solution for modern educational institutions seeking to improve their scheduling processes.



Existing System time table process

2. RELATED WORKS

* 2007 – Wang, Zhang & Shin: Introduced real-time monitoring methods to detect abnormal system behavior in complex environments.

* 2013 – Zargar, Joshi & Tipper: Reviewed optimization techniques for handling large-scale complex computational problems.

Timetable scheduling has been widely studied as a complex optimization problem due to the presence of multiple constraints such as faculty availability, classroom allocation, subject distribution, and time slot limitations. Over the years, researchers have proposed various computational and optimization techniques to automate and improve the timetable generation process.

Wang, Zhang, and Shin (2007) introduced real-time monitoring and optimization techniques to handle complex system behaviors in large-scale environments. Their work highlighted the importance of automated decision-making systems in reducing manual intervention and improving efficiency. Although their study was not directly focused on educational timetabling, the principles of optimization and constraint handling influenced later scheduling systems.

Zargar, Joshi, and Tipper (2013) conducted a detailed review of optimization methods used to solve large-scale computational problems. They discussed how heuristic and metaheuristic algorithms are effective in handling problems with multiple constraints and large search spaces. This research laid the foundation for applying such algorithms to timetable scheduling, where finding an optimal solution through brute-force methods is impractical.

Johnson and Kumar (2019) demonstrated that intelligent resource management systems significantly improve operational efficiency. Their study emphasized the role of automated systems in balancing workloads and optimizing resource usage, which is directly applicable to academic timetabling where faculty and classroom resources must be efficiently utilized.

Chen et al. (2021) presented a comprehensive survey on university course timetabling problems and analyzed various solution techniques, including Genetic Algorithms and Simulated Annealing. Their findings showed that metaheuristic approaches outperform traditional rule-based and manual methods, especially when dealing with complex and large datasets. This study strongly supports the use of optimization engines such as OptaPlanner in automated timetable systems.

Tan et al. (2021) focused on optimization methodologies used in school timetabling problems and highlighted the need for scalable, reusable, and flexible scheduling solutions. They pointed out that many existing systems lack adaptability when academic requirements change, reinforcing the importance of modular and extensible system architectures.

Lee and Kim (2022) recommended cloud-based and modular architectures to improve scalability and performance in academic management systems. Their research emphasized the integration of modern technologies and APIs to ensure seamless communication between system components, which aligns with the architectural design of the Chrono Class system.

From the literature, it is evident that metaheuristic optimization techniques combined with modern software architectures provide an effective solution to timetable scheduling problems. However, many existing systems are either too complex to deploy or lack flexibility for real-world academic environments. Chrono Class addresses these limitations by integrating proven optimization algorithms with a user-friendly, scalable, and automated system tailored specifically for educational institutions.

3.THE PROPOSED CONSTRUCTION

Chrono Class is an AI-based system that automatically creates conflict-free timetables using optimization algorithms. It reduces manual work, improves resource usage, and provides fast and efficient scheduling for educational institutions.

The proposed Chrono Class system is designed to automate the timetable generation process **using AI-**based optimization techniques. The methodology focuses on efficiently allocating faculty members, subjects, classrooms, and time slots while satisfying all academic constraints. The system follows a structured and modular approach to ensure accuracy, flexibility, and scalability.

1. Data Collection and Preparation

The first step in the methodology involves collecting essential institutional data required for timetable generation. This includes information about faculty members, subjects, classrooms, working days, time slots, and faculty availability. The collected data is validated to ensure completeness and correctness. Proper data preparation is crucial, as the quality of input data directly affects the accuracy of the generated timetable.

2. Constraint Definition

Once the data is prepared, the system defines a set of constraints that guide the scheduling process. These constraints are categorized into two types:

- Hard Constraints, which must be strictly satisfied, such as preventing faculty or room conflicts and ensuring correct subject allocation.

- Soft Constraints, which aim to improve timetable quality, such as balanced faculty workload and preferred time slots.

Defining these constraints allows the system to generate feasible and high-quality schedules.

3. Optimization Using Metaheuristic Algorithms

The core functionality of Chrono Class is handled by OptaPlanner, which uses metaheuristic optimization techniques such as Genetic Algorithm and Simulated Annealing. These algorithms explore multiple timetable combinations and iteratively improve solutions by minimizing conflicts and constraint violations. This approach enables the system to efficiently search large solution spaces and produce optimal timetables within a reasonable time.

4. Timetable Generation

Based on the optimization process, the system generates a conflict-free timetable that satisfies all defined constraints. The generated timetable is reviewed automatically to ensure compliance with institutional rules. If necessary, adjustments can be made by modifying constraints or input data, and the timetable can be regenerated with minimal effort.

5. Database Storage

The finalized timetable is stored in a relational database such as MySQL or PostgreSQL. This ensures data persistence, consistency, and easy retrieval. Storing timetables in a database also allows version management and future updates.

6. REST API Integration and Access

The system uses REST APIs to provide secure access to timetable data. Different user roles such as Admin, Head of Department, Faculty, and Students can access the timetable based on their permissions. This ensures controlled access and smooth integration with other academic systems.

7. Feedback and Improvement

Chrono Class supports future improvements by allowing feedback-based modifications. Changes in faculty availability or institutional requirements can be easily incorporated, and the timetable can be regenerated efficiently. This makes the system adaptable to dynamic academic environments.

3.1. Data Collection

Institutional data such as teachers, subjects, classrooms, and time slots are gathered and prepared for scheduling.

3.2. Optimization Engine

OptaPlanner applies Genetic Algorithm and Simulated Annealing to generate the best possible timetable.

3.3. Constraint Handling

System checks rules like teacher availability, room allocation, and time conflicts to ensure a feasible schedule.

3.4. Database Storage

The optimized timetable is stored in a relational database for secure and easy access.

3.5. REST API Integration

Timetable data is shared through REST APIs for system integration and user access.

3.6. Visualization & Feedback

Generated schedules can be viewed and improved using feedback for better future results

4.System Model

The **Chrono Class system model** represents the overall structure and interaction between users, system modules, and the optimization engine used for timetable generation.

4.1. User Layer

This layer consists of authorized users who interact with the system:

- **Admin**
- **HOD**
- **Faculty**
- **Students (view-only)**

Users access the system through a secure interface to manage or view timetable data.

4.2. Application Layer

This layer handles all core functionalities of the system.

a) User Management Module

b) Master Data Module

c) Scheduling Module

4.3. Optimization Layer

This is the core intelligence of the system.

Uses **OptaPlanner** with **metaheuristic algorithms** such as:

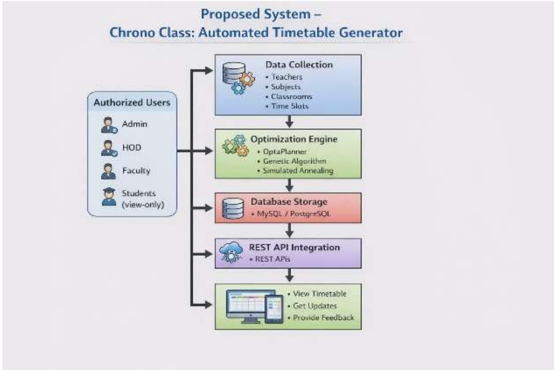
- Genetic Algorithm
- Simulated Annealing

4.4. Data Layer

- Optimized timetables and master data are stored in a **relational database** (MySQL/PostgreSQL).
- Ensures data consistency and persistence.
- Supports updates and modifications.

4.5. API Layer

- Exposes system functionalities using **REST APIs**.
- Enables communication between frontend and backend.
- Allows integration with other academic systems.



Proposed working System of Automated time table generator

5. ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to **Dr. K. Karuppasamy**, Head of the Department of Computer Science and Engineering, RVS College of Engineering and Technology, for his continuous guidance, encouragement, and valuable suggestions throughout the development of this project. His support and motivation played a crucial role in the successful completion of the work.

The authors also extend their heartfelt thanks to **Ms. R. Thenmalar**, Assistant Professor, Department of Computer Science and Engineering, for her constant support, insightful guidance, and constructive feedback during every phase of the project. Her expertise and timely advice greatly helped in improving the quality of this work.

We would like to thank all the faculty members of the Department of Computer Science and Engineering for their cooperation and support. We are also grateful to the management of RVS College of Engineering and Technology for providing the necessary facilities and resources to carry out this project successfully.

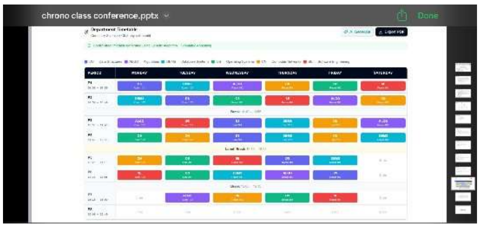
Finally, we would like to express our sincere appreciation to our friends and family members for their encouragement and moral support, which helped us stay motivated throughout the project duration.

Flexible time table

6.CONCLUSION

- This project delivers a flexible, optimized Java-based timetable system. It efficiently solves complex scheduling problems, drastically reducing manual effort and creating conflict-free schedules for diverse institutions.

- Chrono Class successfully automates the timetable scheduling process using AI-based optimization techniques. The system eliminates conflicts, improves efficiency, and significantly reduces manual effort. Its scalable architecture and intelligent design make it suitable for educational institutions and training centers.



Sample output of time table

7.REFERENCES

- [1] M. C. Chen, S. N. Sze, S. L. Goh, N. R. Sabar, and G. Kendall, “A survey of university course timetabling problem: Perspectives, trends and opportunities,” *IEEE Access*, vol. 9, pp. 106515–106529, 2021, doi: 10.1109/ACCESS.2021.3098972.
- [2] J. S. Tan, S. L. Goh, G. Kendall, and N. R. Sabar, “A survey of the state-of-the-art of optimisation methodologies in school timetabling problems,” *Expert Systems with Applications*, vol. 165, Art. no. 113943, Mar. 2021.
- [3] T. Song, M. Chen, Y. Xu, D. Wang, X. Song, and X. Tang, “Competition-guided multi-neighborhood local search algorithm for the university course timetabling problem,” *Applied Soft Computing*, vol. 110, Art. no. 107624, Oct. 2021.
- [4] Z. Zargar, A. Joshi, and D. Tipper, “A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks,” *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 2046–2069, 2013.
- [5] R. Wang, H. Zhang, and K. G. Shin, “Change-point detection with dynamic monitoring in complex systems,” *IEEE Transactions on Dependable and Secure Computing*, vol. 4, no. 4, pp. 259–272, Oct.–Dec. 2007.
- [6] S. Johnson and V. Kumar, “Intelligent resource management using optimization techniques,” *International Journal of Computer Applications*, vol. 178, no. 24, pp. 1–6, 2019.
- [7] J. Drools and OptaPlanner Community, “OptaPlanner User Guide,” Red Hat, 2023. [Online]. Available: https://www.optaplanner.org
- [8] L. Lee and H. Kim, “Design of scalable cloud-based academic management systems,” *International Journal of Information Management*, vol. 62, Art. no. 102439, 2022.
- [9] S. L. Goh, G. Kendall, and N. R. Sabar, “Improving timetable scheduling using metaheuristic algorithms,” *Applied Intelligence*, vol. 51, no. 4, pp. 2152–2170, 2021.
- [10] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd ed. Cambridge, MA, USA: MIT Press, 2009.

