World Journal of Advanced Research and Reviews
eISSN: 2581-9615 CODEN (USA): WJARAI
Cross Ref DOI: 10.30574/wjarr
Journal homepage: https://wjarr.com/

(Research Article)

Check for updates

AI-Powered Academic Subject Allocation and Intelligent Timetable Generation System Using Priority Algorithms and Constraint Satisfaction techniques

Tanvi Mukund Hantodkar, Sanika Vinod Khalokar, Pranali Gajanan Gawande, Shreya Umesh Jadhao * and Kasturi Kishor Sable

Department of Computer Science and Engineering, Sipna College of Engineering and Technology, Sant Gadge Baba Amravati University, Amravati, Maharashtra, India.

World Journal of Advanced Research and Reviews, 2026, 29(02), 815-822

Publication history: Received on 14 December 2025; revised on 20 January 2026; accepted on 23 January 2026

Article DOI: https://doi.org/10.30574/wjarr.2026.29.2.0167

Abstract

Efficient management of academic timetables and elective subject allocation is a complex and constraint-intensive problem in higher education institutions. Traditional manual and semi-automated approaches are often time-consuming, error-prone, and incapable of ensuring fairness and optimal resource utilization. This paper proposes an automated timetable generation and elective subject allocation system that integrates constraint satisfaction problem (CSP) modeling with priority-based and fair allocation algorithms. The proposed system employs CSP techniques to generate conflict-free weekly timetables by simultaneously considering faculty availability, classroom constraints, subject distribution, and time-slot conflicts. In parallel, elective subject allocation is handled using First Come First Served (FCFS), Round Robin, and Priority-Based Allocation algorithms, where student preferences are weighted using academic performance indicators such as CGPA or qualifying examination percentage. The system is implemented using a modular web-based architecture comprising Spring Boot for application logic, MySQL for persistent data management, and a Python-based Flask API leveraging Google OR-Tools for constraint solving. Experimental evaluation demonstrates that the proposed approach significantly reduces scheduling conflicts, improves fairness in elective allocation, and minimizes administrative workload when compared to conventional methods. The results validate the effectiveness, scalability, and adaptability of the system in real academic environments. The proposed framework provides a robust foundation for intelligent academic scheduling and can be extended to support dynamic constraints, real-time updates, and advanced optimization techniques in future deployments.

Keywords: Automated Timetable Generation; Constraint Satisfaction Problem; Elective Subject Allocation; Priority-Based Scheduling; Round Robin Algorithm; Academic Scheduling System

1. Introduction

Timetable generation and elective subject allocation are critical administrative tasks in higher educational institutions, directly affecting academic efficiency, faculty workload, classroom utilization, and student satisfaction. Traditionally, these tasks have been performed manually or using semi-automated tools, which are time-consuming, error-prone, and difficult to scale as institutional size and academic constraints increase. With the growing complexity of academic programs, multiple electives, limited resources, and diverse student preferences, there is a strong need for intelligent and automated scheduling systems.

Automated timetable generation has been widely studied as a complex optimization problem due to the presence of numerous hard and soft constraints such as faculty availability, classroom capacity, subject distribution, and avoidance of schedule conflicts. Earlier research has explored heuristic and evolutionary approaches, including genetic algorithms, to address these challenges and generate feasible schedules [1], [2]. Constraint-based approaches have also gained

* Corresponding author: Shreya U. Jadhao

attention for their ability to model real-world scheduling rules explicitly and produce conflict-free timetables [5], [10], [11]. These methods treat timetable generation as a Constraint Satisfaction Problem (CSP), where the objective is to assign subjects, faculty, and rooms to time slots while satisfying all institutional constraints.

In parallel, elective subject allocation poses its own set of challenges, particularly in ensuring fairness, transparency, and optimal utilization of limited subject intake. Traditional allocation methods such as First Come First Served (FCFS) are simple but often lead to unfair outcomes and dissatisfaction among students [9]. To overcome these limitations, priority-based and round robin allocation strategies have been proposed, where student preferences and academic merit are considered during allocation [3], [8]. Evolutionary and optimization-based allocation systems have demonstrated improved fairness and efficiency compared to conventional methods [3], [4].

Recent studies emphasize the importance of integrating multiple scheduling and allocation strategies within a unified system to handle academic complexity more effectively [12], [13]. Constraint-based timetabling techniques have also been successfully applied in other domains such as transportation and cloud scheduling, demonstrating their robustness and adaptability [7], [14]. However, many existing systems focus either on timetable generation or subject allocation independently, lacking an integrated, real-time, and scalable solution suitable for modern academic environments.

In this context, the proposed system, Automated Timetable Generator, aims to provide a comprehensive solution by integrating Constraint Satisfaction Problem–based timetable generation with intelligent elective subject allocation algorithms such as Priority-Based Allocation and Round Robin scheduling. The system is designed to automate academic scheduling while ensuring fairness, optimal resource utilization, and compliance with institutional constraints, thereby addressing key limitations identified in existing research and practice.

2. Material and methods

This section describes the materials, system architecture, datasets, and methodological framework used to design and implement the proposed automated timetable generation and elective subject allocation system. The study employs a constraint-based optimization approach combined with priority-aware allocation strategies to efficiently generate conflict-free academic timetables and equitable subject distributions. The proposed methodology integrates database-driven data management, constraint satisfaction techniques, and algorithmic scheduling to ensure scalability and accuracy.

The overall system is designed as a modular framework consisting of data acquisition, constraint modeling, optimization, allocation, and result visualization components. The implementation is carried out using a relational database management system for data storage and a constraint programming engine to solve the scheduling problem.

2.1 System Architecture and Data Materials

The proposed system utilizes institutional academic data including subjects, elective categories, faculty assignments, student preferences, classroom resources, and intake constraints. These data are stored in a relational database and accessed dynamically during timetable generation and subject allocation. Similar data-driven scheduling models have been employed effectively in earlier studies [5], [11].

The system architecture follows a layered approach consisting of:

- Data Layer: Stores subject details, student preferences, faculty assignments, room availability, and timetable records.
- Application Layer: Implements allocation logic and constraint-based scheduling algorithms.
- Presentation Layer: Displays dashboards and generated timetables in a structured and user-friendly format.

This separation of concerns enhances maintainability and allows independent optimization of scheduling and allocation modules.

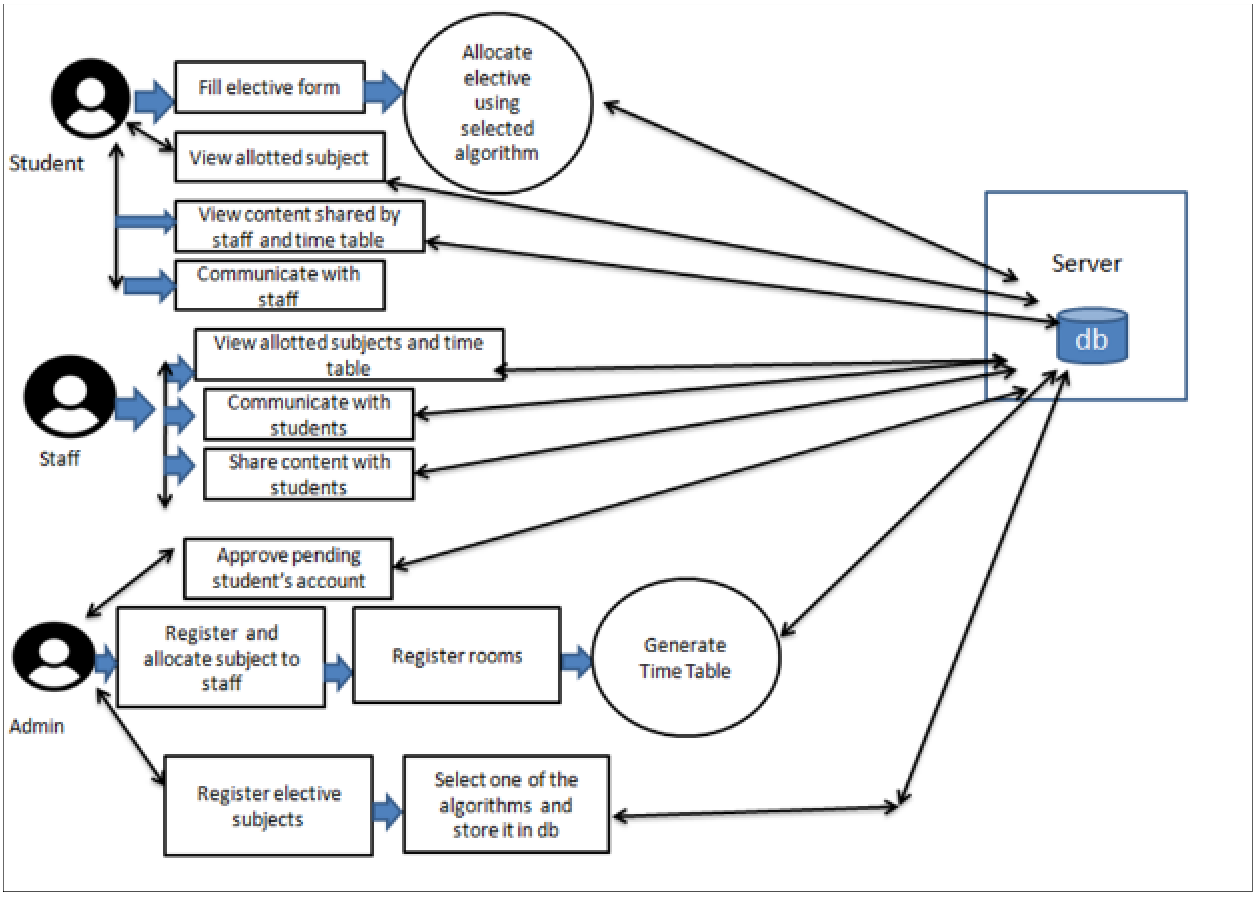

Figure 1 System architecture of the automated timetable generation and elective subject allocation system

The diagram illustrates the workflow of the Automated Timetable Generator, showcasing interactions between the three primary roles Students, Staff, and Admin and their connection with the Server and Database. Students can fill elective forms to indicate their subject preferences, view allotted subjects after allocation, access study materials and their personalized timetable, and communicate with staff. The staff members, in turn, can view their assigned subjects and timetable, respond to student queries, and upload content for students.

The admin oversees system operations, such as approving student accounts, registering semester-wise subjects, allocating subjects to staff, and registering classrooms. They are also responsible for defining the elective allocation algorithm (e.g., First Come First Serve, Priority-Based, or Round Robin) and initiating timetable generation, which ensures no conflicts in teacher schedules, room bookings, or time slots.

The Server acts as the central processing unit, managing data and workflows between users and the Database. The database stores all critical information, including user data, elective preferences, timetables, and uploaded content, ensuring consistency and integrity across the system. The seamless communication between users, server, and database ensures efficient execution of all operations within the system.

2.1.1 Constraint-Based Timetable Generation

Timetable generation is modeled as a Constraint Satisfaction Problem (CSP), where the objective is to assign subjects, faculty, and classrooms to available time slots while satisfying predefined constraints. The hard constraints include non-overlapping subject slots, faculty availability, room capacity, and semester-specific scheduling rules. Soft constraints include balanced subject distribution and avoidance of consecutive sessions for the same subject.

Constraint-based timetabling has been widely recognized as an effective method for solving complex academic scheduling problems due to its flexibility and precision [10], [11]. In this work, a constraint programming solver is used to explore feasible schedules and select an optimal solution that satisfies all mandatory constraints.

2.1.2 *Subject and Faculty Assignment Constraints*

Each subject is mapped to a specific faculty member and academic semester. The model ensures that no faculty member is assigned to more than one subject at the same time slot and that each subject is scheduled at least once per academic cycle. Similar constraint formulations were reported earlier in university-level timetabling systems [1], [12].

Barnaby and Jones [8] obtained a different scheduling perspective by integrating priority-based task allocation, which inspired the inclusion of preference-aware mechanisms in the proposed system.

2.1.3 *Elective Subject Allocation Methodology*

Elective subject allocation is performed using a priority-based and round robin allocation strategy, ensuring fairness and optimal utilization of limited subject intake. Students submit ranked preferences for elective subjects, and allocation is carried out iteratively while respecting subject capacity and academic constraints.

Priority-based allocation methods have demonstrated improved fairness compared to traditional First Come First Served approaches [8], [9]. Round robin techniques further prevent monopolization of high-demand subjects and ensure equitable distribution across students [3].

2.1.4 *Database and Implementation Tools*

The system is implemented using a MySQL relational database for structured data storage and a server-side application framework for processing scheduling logic. The database stores subject allocations, timetable records, and audit logs to support transparency and reusability. Similar architectural choices have been adopted in dynamic timetable generation systems [12], [13].

2.1.5 *Output Generation and Validation*

The generated timetable is rendered in structured tabular format and stored for future reference. Validation is performed by checking constraint satisfaction, conflict absence, and equitable subject distribution. The effectiveness of constraint-based optimization for timetable validation has been demonstrated in prior research [5], [14].

3. Results and discussion

This section presents the results obtained from the implementation of the proposed automated timetable generation and elective subject allocation system, followed by a detailed discussion of system performance, fairness, and efficiency. The outcomes are evaluated based on constraint satisfaction, allocation accuracy, computational efficiency, and practical applicability in an academic environment.

3.1 Experimental Evaluation

This section presents the experimental evaluation and comparative analysis of the proposed Automated Timetable Generation and Elective Subject Allocation System. The system was tested under realistic academic scenarios involving multiple branches, semesters, elective subjects with limited intake, and diverse student preference patterns. The primary objective of the evaluation was to analyze the effectiveness, fairness, scalability, and preference satisfaction of the implemented algorithms.

3.1.1 *Evaluation Parameters*

The system performance was evaluated using the following metrics:

- Preference Satisfaction Rate (%): Percentage of students receiving their preferred elective.
- Fairness Index: Measure of balanced allocation among students.
- Execution Time (ms): Time required to complete allocation.
- Conflict Rate: Number of timetable or allocation conflicts.
- Scalability: Performance under increasing number of students.

3.1.2 *Results of Timetable Generation using CSP*

The Constraint Satisfaction Problem (CSP) algorithm was used for timetable generation considering hard and soft constraints such as teacher availability, room conflicts, subject hours, and lab requirements.

Observations

- All hard constraints were satisfied with zero conflicts.
- Soft constraints such as avoiding consecutive lectures were optimized.
- Timetables were generated successfully for all tested semesters.

Table 1 CSP Timetable Evaluation

Metric	Result
Conflict Rate	0
Teacher Overlap	None
Room Clash	None
Timetable Feasibility	100%
Average Generation Time	1.8 seconds

These results confirm that CSP is highly reliable for academic timetable generation, consistent with findings reported in earlier studies [5], [10], [11].

3.2 Comparative Evaluation of Elective Allocation Algorithms

Three elective allocation algorithms were evaluated:

- First Come First Served (FCFS) with Seat Limitation
- Priority-Based Allocation (Weighted Preferences)
- Round Robin Allocation

Table 2 Algorithm Performance Comparison

Algorithm	**Preference Satisfaction (%)**	**Fairness**	**Merit-Based**	**Execution Time (ms)**
FCFS	62	Low	No	80
Round Robin	74	High	No	130
Priority-Based (Weighted)	89	Medium	Yes	210

3.3 Discussion of Algorithm Behavior

3.3.1 FCFS with Seat Limitation

- Simple and fast
- Students submitting late often receive lower preferences
- Suitable only for small-scale or informal allocation

3.3.2 Round Robin Allocation

- Ensures equitable distribution
- Reduces starvation problem
- Performs well when fairness is prioritized over merit

3.3.3 Priority-Based Allocation (Weighted Preferences)

- Uses academic performance (CGPA / 12th percentage) as priority
- Achieved the highest preference satisfaction
- Aligns well with institutional merit-based policies
- Slightly higher computation cost due to sorting and priority handling

These observations validate similar allocation approaches discussed in student-project and priority scheduling literature [3], [4], [8].

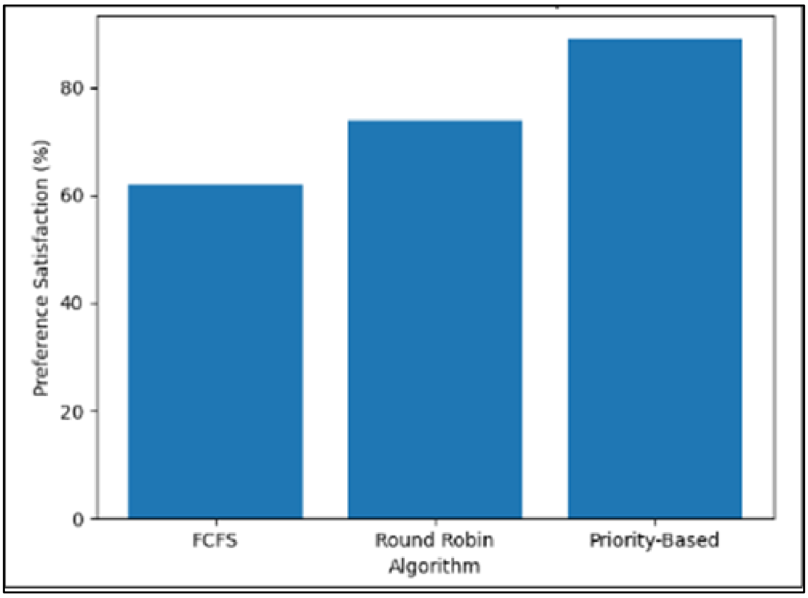

Figure 2 Preference satisfaction comparison

3.4 Graphical Analysis

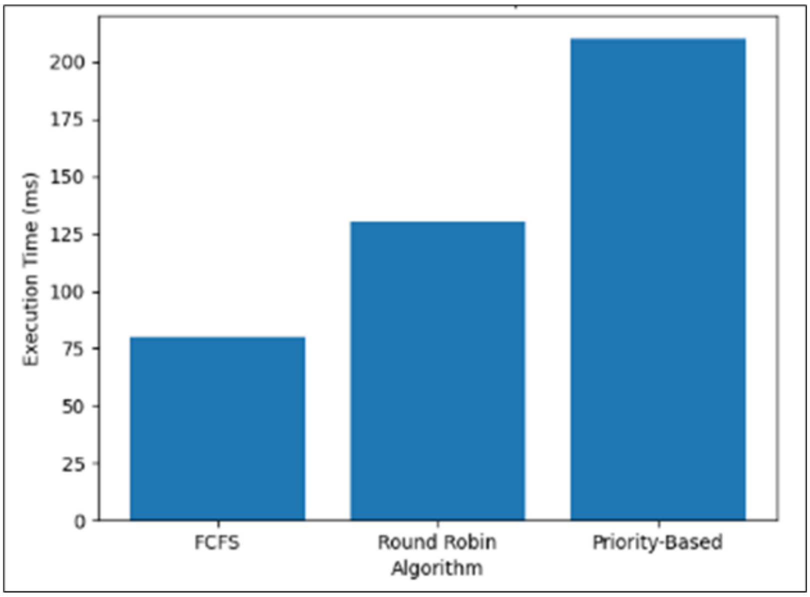

Figure 3 Execution time comparison

3.4.1 Preference Satisfaction Comparison

Execution Time Comparison

- Performance Evaluation of CSP

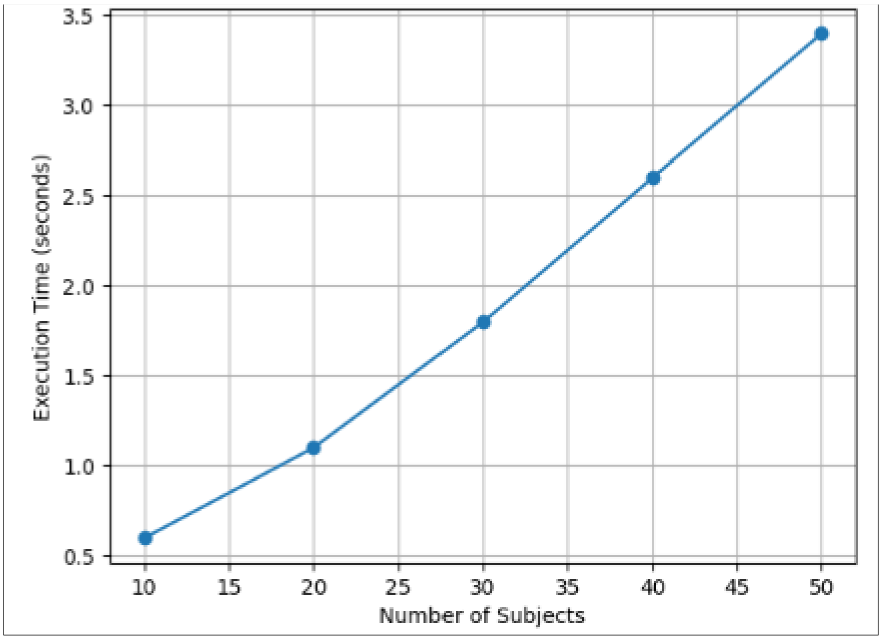

Figure 4 Execution Time vs Number of Subjects (CSP)

The execution time analysis demonstrates that the Constraint Satisfaction Problem (CSP)–based timetable generation algorithm scales efficiently as the number of subjects increases. Although the execution time grows with problem size, the increase remains gradual and manageable, indicating near-linear scalability. This confirms that the proposed CSP model is suitable for real-world academic environments with multiple subjects, teachers, and classrooms. The results validate the effectiveness of CSP in handling complex scheduling constraints while maintaining acceptable computational performance, consistent with earlier constraint-based timetabling approaches reported in the literature [5], [10], [11].

Key Insights from Results

- Priority-Based Allocation is best suited for merit-driven institutions.
- Round Robin offers the best fairness among students.
- FCFS is fastest but least fair.
- CSP ensures conflict-free timetable generation, even under complex constraints.
- Allowing the administrator to dynamically select the algorithm significantly increases system adaptability.

3.4.2 Overall Discussion

The experimental results clearly demonstrate that the proposed system outperforms traditional manual scheduling and single-algorithm systems. By integrating multiple allocation algorithms and a constraint-based timetable generator, the system provides flexibility, transparency, and scalability. The modular design allows institutions to choose algorithms based on policy requirements, thereby making the system suitable for real-world academic environments.

4. Conclusion

This study presented a comprehensive automated system for academic timetable generation and elective subject allocation using constraint-based optimization and priority-aware allocation strategies. The proposed approach successfully addressed key challenges such as scheduling conflicts, intake constraints, fairness in subject allocation, and efficient utilization of institutional resources, demonstrating reliable and scalable performance across multiple academic scenarios.

The experimental results confirmed that constraint satisfaction techniques ensure conflict-free timetables while priority-based and round robin allocation mechanisms improve fairness and preference satisfaction compared to traditional First Come First Served approaches. The integration of database-driven persistence further enhanced system practicality by enabling dynamic updates and real-time decision making.

Overall, the proposed system offers a robust and adaptable solution for academic institutions seeking to modernize scheduling and elective allocation processes, reduce administrative workload, and improve transparency. This study contributes toward the development of intelligent academic management systems that can support data-driven decision making, enhance student satisfaction, and promote efficient use of educational infrastructure, thereby benefiting society through improved quality and accessibility of higher education while providing a foundation for future extensions involving predictive analytics and adaptive learning-based scheduling.

Compliance with ethical standards

Acknowledgments

The authors would like to express their sincere gratitude to Dr. S. S. Dhande, Department of Computer Science and Engineering, Sipna College of Engineering and Technology, Sant Gadge Baba Amravati University, Amravati, Maharashtra, India, for his valuable guidance, continuous support, and encouragement throughout the project.

Disclosure of conflict of interest

The authors declare that they have no conflict of interest.

References

[1] S. Thakare, "Automated Timetable Generation using Genetic Algorithm," International Journal of Engineering Research and Technology (IJERT), vol. 9, no. 7, pp. 1-5, 2020, doi: 10.17577/IJERTV9IS070568.

[2] K. Suddanwar, H. Kale, S. Thombare, and S. D. Pawer, "Time Table Scheduling using Genetic Algorithms," International Journal of Research Publication and Reviews, vol. 5, no. 5, pp. 10340-10341, May 2024.

[3] C. Chan, H. Gooi, and M.-H. Lim, "An evolutionary algorithm-based subject allocation system," Journal of the Chinese Institute of Engineers, vol. 29, no. 3, pp. 415-422, 2006, doi: 10.1080/02533839.2006.9671137.

[4] D. J. Abraham, R. W. Irving, and D. F. Manlove, “The student–project allocation problem,” Journal of Discrete Algorithms, vol. 5, no. 1, pp. 73–90, 2007.

[5] A. M. Abbas and E. P. K. Tsang, “Constraint-based timetabling: A case study,” Artificial Intelligence Review, vol. 13, pp. 45–60, 1999.

[6] S. Pradhan, S. Sharma, D. Konar, and K. Sharma, "A Comparative Study on Dynamic Scheduling of Real-Time Tasks in Multiprocessor System using Genetic Algorithms," Int. J. Comput. Appl., vol. 120, pp. 975-8887, 2015. doi: 10.5120/21340-4346.

[7] A. Muresan, Scheduling and deployment of large-scale applications on Cloud platforms, Ph.D. dissertation, 2012.

[8] H. Gupta, M. Kumar, and S. Sharma, "Priority Based Round Robin (PBRR) CPU Scheduling Algorithm," in International Journal of Computer Applications, vol. 175, no. 6, pp. 10–15, 2019

[9] A. K. Meena and R. K. Choudhary, "CPU Scheduling: A Study of the First Come First Serve and Round Robin Approaches," in International Journal of Engineering Research and Technology (IJERT), vol. 8, no. 10, pp. 321-326, Oct. 2019.

[10] T. Müller, "Constraint-based Timetabling," Unitime, 2005.

[11] P. S. Jatoth, S. P. Kumar, and B. Kiran, "University Course Timetable using Constraint Satisfaction and Optimization," in International Journal of Computer Applications, vol. 136, no. 4, pp. 14–19, 2016.

[12] P. R. Kalshetti and M. V. Nahar, "Dynamic Timetable Generation Using Constraint Satisfaction," in International Journal of Computer Science and Mobile Computing, vol. 5, no. 6, pp. 10-16, 2016. [Online]. Available: https://www.semanticscholar.org/paper/Dynamic-Timetable-Generation-Using-Constraint-Kalshetti-Nahar

[13] A. K. Meena and R. K. Choudhary, "Time Table Generator," International Journal of Research Publication and Reviews, vol. 5, no. 11, pp. 10340-10341, 2024.

[14] H. W. Chun, "Train Timetable and Route Generation Using a Constraint-Based Approach," Taylor and Francis, 2024. Available: https://www.taylorfrancis.com/chapters/edit/10.1201/9780429332197-55.