IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 13, Issue 2, April-May 2025
ISSN: 2320 – 8791 (Impact Factor: 2.317)
www.ijreat.org

Web Based Timetable Generation System Leveraging Genetic Algorithm

Dr. R. Kavitha[1], K.Aleenshylaja[2], R.Rajapriyanka[3], K.Madeshwaran[4]

[1]Professor, Department of Computer Science and Engineering,
Parisutham Institute of Technology and Science, PITS
Thanjavur, Tamil Nadu - 613 006, India
pits.hod.cse@gmail.com

[2]UG Student, Department of Computer Science and Engineering,
Parisutham Institute of Technology and Science, PITS
Thanjavur, Tamil Nadu - 613 006, India
shylinji15@gmail.com

[3]UG Student, Department of Computer Science and Engineering,
Parisutham Institute of Technology and Science, PITS
Thanjavur, Tamil Nadu - 613 006, India
rajapriyanka1101@gmail.com

[4]UG Student, Department of Computer Science and Engineering,
Parisutham Institute of Technology and Science, PITS
Thanjavur, Tamil Nadu - 613 006, India
madeshwaran1101@gmail.com

Abstract

In today's Educational Landscape, managing academic administration in educational institutions remains a complex challenge due to inefficiencies in scheduling and leave management. Existing systems often lack real-time coordination, leading to workload imbalances, delays and miscommunication. This study presents an optimized scheduling system utilizing a Genetic Algorithm, an evolutionary computing technique that iteratively refines solutions for optimal results. The proposed system integrates Spring Boot, a Java-based framework that simplifies backend development with features like auto-configuration and efficient dependency management. By leveraging a Genetic Algorithm for timetable generation, the system enhances workload distribution, improves resource utilization and ensures seamless coordination. The results demonstrate that this intelligent framework significantly improves operational efficiency of scheduling processes in the educational systems. Thus, the Genetic Algorithm-based approach offers a robust solution for automating academic administration in educational institutions.

Keywords-- *Genetic Algorithm, Academic Scheduling, Spring Boot, Educational Administration*

I. INTRODUCTION

In recent years, academic institutions have faced significant challenges in timetable generation due to complex scheduling requirements, unbalanced workload distribution, and frequent conflicts in faculty assignments. Traditional scheduling methods often struggle to accommodate institutional constraints, leading to inefficiencies and resource mismanagement. However, the implementation of a Genetic Algorithm-based scheduling system has emerged as a transformative solution, optimizing timetable generation by automating faculty assignments and ensuring an even workload distribution. The proposed system leverages genetic algorithm techniques to iteratively refine scheduling solutions based on faculty availability, subject requirements, and institutional policies. By analyzing existing scheduling patterns, the system intelligently optimizes timetables to minimize conflicts while maintaining structured academic workflows. The algorithm evaluates multiple timetable configurations, selecting the most efficient arrangement that adheres to predefined constraints such as classroom allocation and teaching hours. By automating the timetable generation process, the system reduces human errors, eliminates scheduling conflicts, and enhances institutional efficiency. Administrators gain valuable insights into resource allocation, enabling data-driven decision-making in academic planning. The integration of Spring Boot technology ensures a scalable and robust platform, facilitating real-time coordination between faculty and administrators. With its intelligent scheduling capabilities, the system streamlines faculty assignments, optimizes institutional workflows and enhances overall academic administration. By eliminating manual interventions and ensuring an efficient timetable structure, the Genetic Algorithm-based scheduling system significantly improves resource utilization and institutional outcomes.

II. LITERATURE SURVEY

Osyczka and Krenich [1] introduced a constraint tournament selection method within genetic algorithms to address multicriteria optimization problems. This approach effectively handles constraints by integrating them into the selection process, enhancing the algorithm's ability to find optimal solutions in complex design scenarios

Dipmala Salunke et al. [2] conducted a survey on Role-Based Access Control (RBAC), highlighting its role in secure authentication and authorization. The study explores permission management, session control and database-independent access layers for enhanced security.

Parkavi A. [3] proposed an automated timetable generation system using genetic and heuristic algorithms to optimize scheduling. The system considers faculty availability, course constraints and workload balancing to create conflict-free timetables. By automating the process, it reduces manual effort and scheduling errors, improving efficiency in academic institutions.

Kavya Guntupally et al. [4] developed a Spring Boot-based REST API to enhance data quality report generation for large scientific datasets. This system improves the efficiency and accuracy of data processing workflows within research data centers.

Krishna Shingala [5] proposed using JSON Web Tokens (JWT) for secure client authentication in MQTT-based IoT systems, reducing reliance on traditional TLS methods. JWT provides lightweight, efficient and scalable authentication while ensuring secure communication between devices. This approach improves performance in constrained environments and aligns with modern cloud-based IoT security practices.

Salman Ahmed and Qamar Mahmood [6] proposed a JWT-based authentication scheme to enhance security in web applications. By eliminating traditional session-based methods, JWT improves scalability, efficiency and secure user verification. The study highlights JWT's role in reducing server load and enhancing performance in distributed systems.

Shraddha Thakare et al. [7] proposed an automated timetable generation system using genetic algorithms to optimize scheduling. Their approach minimizes conflicts by modifying genetic operators, ensuring efficient and clash-free timetable generation.

Sushmita C. Hubli and Dr. R. C. Jaiswal [8] presented a comprehensive overview of Spring Boot for efficient backend development. The paper explores core concepts, architectural principles and best practices, highlighting Spring Boot's ability to simplify the creation of robust, enterprise-grade backend applications.

Dexter Romaguera et al. [9] introduced a web-based course timetabling system utilizing an enhanced genetic algorithm to improve scheduling efficiency in educational institutions. The proposed approach integrates a heuristic mutation technique, which focuses on modifying infeasible genes to enhance the algorithm's exploration and exploitation capabilities. By applying this method to actual datasets from Caraga State University, the system demonstrated optimized classroom resource utilization while ensuring that both hard constraints, such as scheduling conflicts, and soft constraints were met. The study concluded that the enhanced genetic algorithm significantly improves timetable efficiency, reduces computational time and provides a more balanced workload distribution for faculty members.

Zidi Chen and Juan Gao [10] proposed a hybrid architecture-based educational management system that integrates both Client/Server (C/S) and Browser/Server (B/S) models to enhance efficiency in college administration. The study introduced a Genetic Algorithm-based course scheduling system to optimize timetable generation. By analyzing course goodness, scheduling uniformity and interval effectiveness, the system ensures optimal allocation of faculty and resources. The experimental results demonstrated that the genetic algorithm approach significantly. Secondly, the dataset is structured to represent diverse scheduling requirements. Factors such as laboratory periods and elective courses are incorporated to ensure comprehensive scheduling. effectively enhances educational administration systems in colleges and universities.

Aviraj Latpate et al. [11] introduced an AI-based Automatic College Timetable Generator utilizing ReactJS and Firebase to streamline scheduling in educational institutions. The system aims to minimize manual effort and scheduling conflicts by incorporating optimization techniques, ensuring efficient resource allocation. The proposed method integrates constraint-based scheduling, considering faculty availability, student preferences and room utilization. The user-friendly interface allows administrators to generate and modify schedules dynamically. The research highlights the significance of automation in academic planning, demonstrating how AI-driven solutions can improve efficiency, accuracy and adaptability in timetable generation.

III. PROPOSED MODEL

In response to the challenges faced in academic scheduling, the proposed model leverages the power of genetic algorithms (GA) to automate and optimize the generation of timetables. By utilizing an evolutionary approach, it aims to improve efficiency, fairness and adaptability in academic scheduling while minimizing conflicts and ensuring adherence to constraints.

A. DATA ACQUISITION

Acquiring a high-quality and representative dataset is a fundamental step in developing an effective genetic algorithm for timetable generation. The dataset encompasses faculty availability, course requirements, room allocations and institutional constraints to ensure the model's practicality and real-world applicability.

The data acquisition process involves multiple steps and considerations. Firstly, gathering information from institutional databases, faculty preferences and course structures is essential for defining scheduling constraints. This includes details such as:

- Course names, subjects and credit hours
- Faculty members assigned to courses
- Availability and unavailability of faculty

B. GENETIC ALGORITHM FOR TIMETABLE GENERATION

Genetic algorithms mimic the principles of natural selection to find optimal solutions for complex problems. The timetable generation process follows these steps:

F(T) = ∑ (Bonuses from soft constraints) – ∑ (Penalties from hard constraints violations)

Fig. 1 Genetic Algorithm workflow.

1. Initial Population Generation

An initial population of random feasible timetables are generated, ensuring that constraints such as faculty availability and batch assignments are initially respected.

2. Fitness Function

The fitness function evaluates each timetable based on: conflict minimization (faculty and course overlaps), adherence to predefined constraints (lab periods, preferred slots), balanced distribution of classes throughout the week and satisfaction of faculty and institutional requirements

$$F(T) = \sum \text{(Bonuses from Soft Constraints)} - \sum \text{(Penalties from hard Constraints Violations)} \quad (1)$$

3. Selection

A selection mechanism, such as tournament selection is applied to choose the best-performing timetables for reproduction.

4. Crossover and Mutation

Parent timetables are combined using crossover techniques (e.g., one-point or two-point crossover) to produce new offspring timetables that inherit properties from both parents. A small percentage of the timetables undergo mutations, where randomly selected genes are modified to introduce diversity and avoid local optima.

5. Iteration

The process iterates for multiple generations, continuously refining the timetables until an optimal or near-optimal solution is found based on the fitness function.

C. PERFORMANCE METRICS

Evaluating the performance of the genetic algorithm involves rigorous testing, analysis and validation using appropriate metrics and techniques. The model is assessed based on:

1. Constraint Satisfaction Rate

Measures how well the generated timetables adhere to hard constraints (e.g., no faculty or room conflicts) and soft constraints (e.g., preferred time slots).

2. Optimization Metrics

Fitness Score: Quantifies the quality of generated timetables based on predefined criteria.

Time Complexity: Evaluates the computational efficiency of the algorithm.

3. Comparative Analysis

The genetic algorithm-generated timetables are compared with manually created schedules to assess improvements in fairness, conflict resolution and overall efficiency.

IV. CONCLUSION

The proposed Web Based Educational Management System utilizing a Genetic Algorithm presents a transformative approach to academic scheduling. By leveraging evolutionary computing techniques, the system optimizes faculty assignments, reduces scheduling conflicts and ensures efficient workload distribution. The integration of spring boot provides a scalable and robust backend, facilitating seamless real-time coordination between professors and administrators. Future enhancements could focus on expanding the dataset with more diverse scheduling scenarios to improve generalization and adaptability. Further optimization of the genetic algorithm, including advanced mutation and selection strategies, can enhance convergence speed and overall performance. Exploring ensemble learning techniques and hybrid optimization methods may further improve timetable efficiency. Continuous evaluation and refinement of the system will ensure its long-term reliability and effectiveness in real-world educational environments. By embracing cutting-edge computational techniques and automation, this research contributes to the advancement of academic administration, fostering efficiency, transparency and improved institutional planning. Ultimately, this approach holds significant promise for educational institutions striving for optimized resource management and enhanced operational workflows.

REFRENCES

[1] Osyczka and Krenich, "A New Constraint Tournament Selection Method for Multicriteria Optimization Using Genetic Algorithm," Proceedings of the 2000 Congress on Evolutionary Computation (CEC 2000), 1 (2000) 501–508.

[2] Dipmala Salunke et al."A Survey Paper on Role-Based Access Control," International Journal of Advanced Research in Computer and Communication Engineering, vol. 2, no. 3, pp. 1340–1344, March 2013.

[3] A. Parkavi, "A Study on Automatic Timetable Generator," International Journal of Innovative Research and Growth, 7 (5) (2018) 479–482.

[4] Kavya Guntupally et al. "Spring Boot Based REST API to Improve Data Quality Report Generation for Big Scientific Data: ARM Data Center Example," 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 2018, pp. 2721–2725.

[5] Krishna Shingala, "JSON Web Token (JWT) Based Client Authentication in Message Queuing Telemetry Transport (MQTT)," International Journal of Engineering and Advanced Technology (IJEAT), 8 (4) (2019) 42–46.

[6] Salman Ahmed and Qamar Mahmood, "An Authentication-Based Scheme for Applications Using JSON Web Token," 2020 International Conference on Computing, Electronics & Communications Engineering (iCCECE), 2020, pp. 1–6.

[7] Shraddha Thakare et al. "Automated Timetable Generation Using Genetic Algorithm," International Journal of Scientific & Technology Research (IJSTR), 9 (8) (2020) 8192–8196.

[8] Sushmitha C. Hubli and Dr. R. C. Jaiswal, "Efficient Backend Development with Spring Boot: A Comprehensive Overview," International Journal for Research in Applied Science & Engineering Technology (IJRASET), vol. 11, no. 11, pp. 759–765, November 2023.

[9] Dexter Romaguera et al. "Development of a Web-based Course Timetabling System Based on an Enhanced Genetic Algorithm," Procedia Computer Science, vol. 234, pp. 1714–1721, 2024.

[10] Zidi Chen and Juan Gao, "Design of College Educational Management System Based on CS and BS Hybrid Architecture," Measurement: Sensors, vol. 33, 2024, 101201.

[11] Aviraj Latpate et al. "AI-Based Automatic Timetable Generator Using React," International Journal of Creative Research Thoughts (IJCRT), vol. 12, no. 4, April 2024, pp. 136–140.