International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 01 | Jan - 2025 SJIF Rating: 8.448 ISSN: 2582-3930

Automated Timetable Generation With Real-Time Updates

B Uma
Dept. of Computer Science and Engg.
Malnad College of Engineering Hassan, India
bu@mcehassan.ac.in

Appu H P
Dept. of Computer Science and Engg.
Malnad College of Engineering Hassan, India
appuhp9844@gmail.com

Bhuvan M
Dept. of Computer Science and Engg.
Malnad College of Engineering Hassan, India
bhuvanmuralidhara1@gmail.com

Aathish Shetty
Dept. of Computer Science and Engg.
Malnad College of Engineering Hassan, India
aathishshetty65@gmail.com

Bindu Prasad G S
Dept. of Computer Science and Engg.
Malnad College of Engineering Hassan, India
binduprasad728@gmail.com

***Abstract*—This paper presents an automatic generation of a real-time timetabling system, which overcomes the limitations of traditional systems by avoiding traditional conflicts, in efficiency, and inflexibility. The constraints used in optimizing the generation process are advanced: multiple constraints handling, including availability of teachers, classroom availability, balance in the scheduling, and conflicts. The current system incorporates some innovations, such as the capability to update real-time changes, thus making it easier to handle dynamic changes - for instance, swapping classes between teachers, reports of teacher absence, and special adjustment requests. These updates are instantly synchronized across all devices with minimal disruption to ensure flexibility in day-to-day academic operations. It is built for high scalability and efficiency, with the use of Firebase for real-time data updates, Flutter for a responsive and user-friendly cross-platform experience, and Docker to support scalable and reliable deployments. Moreover, the platform supports enhanced administrative control, real-time notifications, and detailed analytics for improved decision-making. This significantly improves the efficiency, accuracy, and adaptability of the scheduling process as the system has automation in the scheduling process that reduces manual effort, minimizes errors, and provides greater flexibility compared to the traditional static timetables. This paper is based on a detailed design of the system with a technological basis, providing rich benefits to the administrators, faculty, and students.**

I. Introduction

A. Background

Timetabling is one of the most critical and challenging tasks in educational institutions. Traditionally, academic schedules are prepared manually. Administrators spend a lot of time and effort in balancing various constraints such as faculty availability, classroom capacity, course requirements, and student needs. It is common to have errors, inefficiencies, and conflicts, such as overlapping schedules or underutilized resources.

Manual timetables have no flexibility when dealing with changes; for example, faculty absence or unavailability of classrooms may seriously disrupt institutional operations, and communication of updates to stakeholders may significantly be delayed due to the lengthy revisions.

Advances in computation and algorithmic optimization over the last two years have made this possible. Computational models and constraints solving are deployed to generate timetables automatically, which eventually leads to schedules that are faster and more reliable than the ones hand-written with few errors and less paper work. Further, real time updates ensure immediate changes and remain at high levels of accuracy and reliability.

This project aims to design an efficient, user-friendly system that addresses the current inefficiencies in manual timetabling by automating key inputs into the database. The system will provide real-time updates to ensure seamless and timely schedule modifications. Ultimately, it seeks to revolutionize the management of academic schedules, making the process more efficient, accurate, and adaptable.

Computational Complexity:The problem of generating an academic timetable is classified as NP-hard because scheduling under multiple constraints is combinatorial in nature and grows exponentially with an increase in variables. Heuristic and optimization techniques are used to obtain practical solutions within a reasonable timeframe.

B. Problem Statement

The process of scheduling academic timetables is a very time-consuming process and is error-prone. It involves various factors like faculty availability and subject-specific requirements, which make the process complicated. Traditional methods are not flexible in case of sudden changes such as faculty absence, leading to conflicts, inefficiencies, and disruptions. The process is slow in making adjustments and miscommunication arises without real-time updates. Therefore, an automated solution is needed to address these challenges, ensuring conflict-free scheduling and the ability to make seamless real-time adjustments.

C. Research Objective

1) ***Automation Of Timetable Generation:*** Design an automated system for replacing the present manual process of generating timetables, hence saving time and effort in the scheduling process.

2) ***Conflict-Free Scheduling:*** Ensure the system resolves the scheduling conflicts taking into account factors such as the availability of the faculty, capacity of classrooms, subject specific needs, and requirements of students.

3) ***Integration of Real-Time Updates:*** Allow for dynamic changes in the system like rescheduling, cancellations or Faculty changes. All these changes must be captured across the schedule in a flash with least possible disruption.

4) ***Avoiding Human Error:*** Algorithmically solve the sophisticated constraints and dependencies that are imposed, so that the possibility of Human scheduling errors are avoided.

5) ***Optimal Resource Utilization:*** All resources- classrooms, laboratories, and faculties- should be utilized with an optimal degree, without under utilization or double booking.

D. Scope Of The Project

This project will limit itself to the scope of automating the generation of timetables for a semester of an undergraduate engineering program. In particular, it focuses on three sections within a semester, scheduled to use two available classrooms. Thus, this scope allows the project to focus more on the challenge of timetable generation in a limited and manageable setting.

It has the provision for ensuring that a schedule generated takes into account all such constraints as faculty availability, requirements for subjects, and classroom capacities. The system will allow real-time update so that such changes like cancellation, rescheduling, or even change of faculty without much discomfiture in the overall procedure.

Given this particular context, the project is more of a proof of concept in that it testing and validation for the efficiency and reliability of a proposed solution. Although the scope at present may be narrow, it guarantees that this system can easily be expanded into the handling of more complex scenarios such as handling multiple semesters, larger student cohorts, and even additional resources.

E. Significance Of The Project

This is an important project in that it attempts to address one of the most recurrent administrative problems that are faced by educational institutions through effective management of academic timetables. The traditional manual generation of timetables is not only time-consuming but also error-prone and beset with conflicts. The project helps streamline operations, save crucial time, and utilize more resources to heighten overall institutional productivity.

It contributes in the following ways: It can make timetables conflict-free. The system considers faculty availability, classroom capacities, and subject-specific requirements to ensure that scheduling would be very smooth; there would be no overlapping classes. It improves the day-to-day working of institutes since there would be lesser disturbances because of scheduling conflicts.

Changes such as unforeseen faculty leave or room unavailable often occur and are experienced educational institutions. That the system evolves dynamically allows real-time updates which ensures that a change is not only reflected without delay but reduced impact on affected students and teaching staff.

This project, on the other hand, minimizes human interventional efforts and, thus human error by vast proportions. Also, there is an increased precision of timetables. In this aspect also, staff members' workload associated with some of the other work of absolute necessity is diminished.

Overall, this program brings critical value to educational institutions by offering a flexible, scalable solution for timetable management. It benefits stakeholders such that administrators have an efficient tool with which they can schedule the academics, faculty face minimal disruptions, and most importantly, there is a more organized academic environment for the students.

II. Literature Review

A. Overview Of Existing Research

Tarek El-Sakka investigates this in his paper, treating the University Course Timetable Problem as a multidimensional constraint satisfaction problem. He makes a thrust on the fact that due to many constraints involved, automating the production of timetables is a hard and time-consuming task, in particular NP-hard. A CLP/OPL model is presented to effectively solve the problem in the Community College of the University of Sharjah. Real data testing by El-Sakka shows the efficiency of the model in optimizing resource allocation with satisfaction of scheduling constraints, contributing valuable insights to automated timetabling solutions.

Viliam Ď uris˘, in his paper, deals with the Constraint Satisfaction Problem (CSP) concerning the Timetable Problem (TP). The paper discusses the framework for the automation of the TP and deals with fundamental concepts and algorithms associated with CSPs. A tree-based search algorithm to solve the TP is discussed in the paper. The paper also develops a computer program which tests the validity of the constraints and demonstrates its functionality. This work helps in developing ways to automate the timetabling method and overcoming computational complexity problems linked to CSPs.

Nima Khodadadi in this paper, the Dynamic Arithmetic Optimization Algorithm, DAOA is proposed for truss structures that are optimized subject to natural frequency constraints. This algorithm has improved the former Arithmetic Optimization Algorithm (AOA) with the dynamic features added to it. These features enhanced exploration and exploitation during optimization, and this work demonstrates that DAOA can indeed be a great tool for robust structural design optimization in terms of minimizing the truss weight and keeping the frequencies within the constraints. Khodadadi's work has substantially contributed to this field as the complex dynamic behavior in engineering applications is being approached.

Nicola Coviello researches on the Train Timetabling Problem (TTP) and develops the ATMO: an optimization tool designed to automatically create timetables while optimizing the set of different conflicting objectives of travel time, energy consumption, and timetable stability. Here, she applies MOACO in conjunction with MILP to generate the Pareto optimal solutions. This is applied in the context of real-world railway data, making it efficient in strategic railway planning. This research offers a new solution to automated railway timetabling that improves decision-making in the assessment of capacity and infrastructure.

B. Key Findings In The Research

Some researchers consider that the Timetable Problem can be better solved with the help of sophisticated optimization techniques. Tarek El-Sakka and Viliam Ď urisˇ formulated the problem as a Constraint Satisfaction Problem, applied it to automated timetabling. The first author applied CLP/OPL, whereas the second-used a tree search algorithm to check the validity of constraints. Nicola Coviello applies MOACO and MILP to TTP, obviously showing efficiency in the optimization of multiple conflicting objectives. Nima Khodadadi introduces a Dynamic Arithmetic Optimization Algorithm for structural optimization, explaining how dynamic techniques are employed in solving more complex constraint-based problems. In short, CSP remains the best course timetable generation approach since it allows dealing with complex constraints.

III. Methodology

A. Design Of System Architecture

The C4 Model is the base framework used in the design of the system architecture, which is widely accepted in the field for the visualization and documentation of software architecture. It represents a system's structure clearly, hierarchically divided into four levels: Context, Container, Component, and Code. In this project, the focus will be on the Container Level Design, detailing the primary building blocks of the system, their responsibilities, and how they interact.

*1) **Container Level Design**:* There are four major containers that are the base of Automated Timetable Generation System at the container-level architecture; these are Desktop Application, Mobile Application, Backend as well as a Database. Each of the containers plays a unique role in the system and allows it to work efficiently.

a) Desktop Application: The Desktop App will be the interface for admins. It provides an interface that will allow them to manage the process of generation, conflict resolution, and make updates in real time. The container communicates with the Backend Container directly to transmit data, which can include scheduling parameters and updates.

b) Mobile Application: The Mobile App is for both teachers and students, which would be provided with a user-friendly interface to check their schedules and receive real-time updates. The app would be useful for the teachers to show preferences or to indicate unavailability while the students would have their personal timetables. The Mobile App will interact with the Backend Container. All changes are immediately reflected to all users from the Backend Container.

c) Backend Container: The Backend Container is the central processing unit of the system. It manages the core functionalities, such as generating a timetable, conflict resolution, and processing real-time updates. The container also handles communication between the Desktop and Mobile App containers. In addition, it interacts with the Database Container for storing and retrieving data so that consistency and accuracy are ensured across the system.

d) Database Container: The Database Container is responsible for storing all system data, including user information, schedules, preferences, and institutional constraints. This container ensures that data is organized, secure, and readily accessible to the Backend Container. The use of a NoSQL database, such as Firebase Firestore, allows for dynamic data handling and real-time synchronization.

*2) **Interaction Flow**:* The Desktop Application container interacts with the Backend Container to produce administrative activities such as generating timetables and updating data. The Mobile Application also interacts with the Backend Container to ensure that there is real-time access and updates for teachers and students. The Backend Container acts as the intermediary where data fetched from the Desktop and Mobile Apps is processed, and then data can either be fetched or stored in the Database Container. This structured interaction ensures a smooth flow of information across the system, thereby allowing for efficient and reliable management of timetables.

B. Constraint Optimization Technique

Google OR-Tools is a good, efficient open-source optimization library that is remarkably well suited for the solution of timetable generation problems with constraint optimization-particularly its CP-SAT solver. Timetable generation is a complex constraint satisfaction problem with both hard constraints, such as teacher availability and non-overlapping classes, as well as soft constraints, such as minimizing idle hours or balancing workloads.

The timetable generation problem is a Constraint Satisfaction Problem (CSP) that involves:

- **Variables**: Let $X_{t,s,c}$ represent the assignment of a teacher t to section s in time slot c (e.g., Monday 8:30-9:30).
- **Domain**: $X_{t,s,c} \in \{0, 1\}$, where 1 means the teacher t is assigned to section s at time slot c, and 0 otherwise.
- **Constraints**:
  1) **Teacher Availability**: A teacher t can only be assigned if they are available during the time slot c.

$$X_{t,s,c} = 0 \quad \text{if teacher } t \text{ is unavailable at } c.$$

  2) **No Overlap**: A teacher t cannot teach more than one section at the same time:

$$\sum_{s} X_{t,s,c} \leq 1 \quad \forall t, c.$$

3) **Classroom Availability**: Only one section can occupy the classroom at a given time:

$$\sum_{t,s} X_{t,s,c} \leq 1 \quad \forall c.$$

4) **Class Count**: Each course *k* must meet its required number of classes C_k per week:

$$\sum_{c} X_{t,s,c} = C_k \quad \forall k.$$

5) **Breaks**: Breaks such as tea break (10:30-11:00) and lunch break (1:00-2:00) must be unoccupied:

$$X_{t,s,c} = 0 \quad \text{for } c \text{ during break periods.}$$

6) **Saturday Timing**: Classes on Saturday must end by 1:00 PM:

$$X_{t,s,c} = 0 \quad \text{for } c \text{ after 1:00 PM on Saturday.}$$

Constraint optimization techniques allow us to define these variables, domains, and constraints, ensuring the generation of a valid and optimized timetable that meets all requirements.

C. Real-Time Synchronization

The system uses Firebase, a cloud-based backend platform that supports real-time data updates, to achieve real-time synchronization. Firebase's real-time database allows for instantaneous data exchange between the various system components, including the Desktop Application, Mobile Application, Backend, and Database containers. Any changes made in the system, such as timetable adjustments, cancellations, or rescheduling, are propagated instantly to all connected devices.

The update of the timetable by the administrator will be sent to the Backend Container, where it will be processed and validated. The Backend will then communicate with the Database Container to save the changes. Consequently, Firebase's real-time synchronization allows these updates to be instantly pushed to the Mobile Application, where they can be viewed by the teachers as well as the students in the changed schedules.

D. Tools and Technologies

1) Firebase: Firebase serves as the essential back-end structure providing real-time database management, user authentication, cloud storage and hosting services, in which every connected device provides instantaneous updates into a user's timetables, thus achieving free flowing communication.

2) Flutter: The open-source UI development framework, Flutter, combines Dart programming for the front-end. Since Flutter supports cross-platform functionality, it allows running on web and mobile devices with the same codebase, thereby maintaining compatibility and saving time at the development end. It utilizes the Dart language to enhance its capabilities through fast performance and user interface development that is responsive to intuitive needs, using widgets as customizable units.

3) VS Code: The coding and development processes are supported by Visual Studio Code, a lightweight yet powerful integrated development environment. Its extensive features, including debugging tools, language-specific extensions, and version control integration, streamline the development workflow and enhance productivity.

4) Git and Github: This version control system relies on Git in order to maintain versions, monitor changes in the codebase of the project, as well as team collaboration among team members. This hosting server has a repository from GitHub, and it makes code sharing and cooperation easy throughout the development cycle.

5) Adobe XD: The design and experience of the application are made in Adobe XD. With this tool, the team is able to create wireframes and prototypes that can assist in achieving a user-friendly interface as well as smooth interaction by stakeholders. This software bridges the gap between design and development and allows for an overall quality application.

6) Docker: This tool containers the application and thus ensures consistency in performance irrespective of the environment. This technology streamlines the process of deploying and testing, thus making the system scalable and secure. Furthermore, a NoSQL database such as Firebase Firestore has been used for the storage of application data. The ability to store unstructured or semi-structured data will help in real-time synchronization and scalability, which are critical to the dynamic capabilities of the system.

IV. Conclusion

The present project has brought out the shortcomings of conventional methods and underlined the need for a more robust, automated solution through a comprehensive survey of existing systems and techniques.

By implementing the modern tools and technologies of Firebase for real-time synchronization, Flutter for a cross-platform application, and a NoSQL database for handling dynamic data, the system has successfully automated the complex task of timetable creation and management. The C4-based architecture ensures modularity, reliability, and proper communication between the Desktop Application, Mobile Application, Backend, and Database.

This system is very innovative in that it allows the updates to be on real time, meaning any changes on the timetable will flash across all devices connected. This feature helps to meet the important needs of flexibility and responsiveness in order to allow for administrators, teachers, and students to effectively switch their academic operations to address unforeseen disruptions. The system further limits the occurrence of human errors, optimizes resource utilization, and significantly reduces time and effort in planning.

This project demonstrates not only the possibility of automating the generation of timetables for a defined scale-three sections and two classrooms in a semester-but also serves as a foundation for scaling the solution up to larger and more complex scenarios. The use of user-friendly interfaces, robust backend processing, and real-time updates ensures that the system is well-equipped to meet the needs of its stakeholders.

References

[1] S. Dev Gupta, B. Genc¸, and B. O'Sullivan, "Explanation in Constraint Satisfaction: A Survey," in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (JCAI 2021)*, Virtual Event, Montreal, Canada, Aug. 2021, pp. 4400–4407.

[2] N. Khodadadi, V. Snasel, and S. Mirjalili, "Dynamic Arithmetic Op- timization Algorithm for Truss Optimization Under Natural Frequency Constraints," *IEEE Access*, vol. 10, pp. 1–12, Jan. 2022.

[3] V. Dˇ urisˇ, "Algorithmic Verification of Constraint Satisfaction Method on Timetable Problem," *Mathematics and Statistics*, vol. 8, no. 6, pp. 728–739, 2020.

[4] T. El-Sakka, "University Course Timetable Using Constraint Satisfac- tion and Optimization," *International Journal of Computing Academic Research (IJCAR)*, vol. 4, no. 3, pp. 83–95, June 2015.

[5] A. Chowdhary, P. Kakde, S. Dhoke, S. Ingle, R. Rushiya, and D. Gawande, "Timetable Generation System," *International Journal of Computer Science and Mobile Computing*, vol. 3, no. 2, pp. 410–414, February 2014.

[6] N. Coviello, G. Medeossi, T. Nygreen, P. Pellegrini, and J. Rodriguez, "Automatic generation of timetable with the ATMO tool," *Technical Report*, 18/11/2021. ++

[7] DK. R. Rashmi and Dr. A. M. B, "Automated University Timetable Generation using Prediction Algorithm," *International Research Journal of Engineering and Technology (IRJET)*, vol. 8, no. 6, pp. 2345, June 2021, e-ISSN: 2395-0056, p-ISSN: 2395-0072.

[8] L. Y. Ying and H. K. Mammi, "Timetable Scheduling System using Genetic Algorithm (tsuGA)," *International Journal of Innovative Com- puting*, vol. 11, no. 2, pp. 67–72, 2021.