

# Dynamic Timetable Generator

<sup>1</sup>Sujitha D, <sup>2</sup>Amrutha Ch, <sup>3</sup>Sandhya B, <sup>4</sup>Praveena B, <sup>5</sup>Varshitha Ch, <sup>6</sup>Swathi Ch

<sup>1</sup>Assistant Professor, Department of Computer Science and Engineering,

Narayana Engineering College, Nellore, Andhra Pradesh, India

<sup>2,3,4,5,6</sup>Students, Department of Computer Science and Engineering,

Narayana Engineering College, Nellore, Andhra Pradesh, India

---

**Abstract:** A college timetable is a temporal arrangement of a set of lectures and class rooms in which all given constraints are satisfied. Creating such timetables manually is complex and time-consuming process. By automating this process with computer assisted timetable generator can save a lot of precious time of administrators who are involved in creating and managing course timetables. Hence, we have developed practical approach for building lecture -course timetabling system, which can be customized to fit to any colleges timetabling problem. This project introduces a practical timetabling algorithm capable of taking care of both strong and weak constraints effectively, used in an automated timetabling system.

**Keywords:** Genetic Algorithm, Automated Timetable, Soft and Hard Constraints.

---

## I. INTRODUCTION

A Dynamic Timetable Generator system is an innovative solution designed to create schedules that adapt to changing conditions, preferences, and constraints in real-time or near real-time. Unlike traditional timetable generators, which produce static schedules that are fixed once generated, dynamic timetable generators offer flexibility, adaptability, and optimization capabilities that cater to the dynamic nature of modern organizations and institutions.

Every semester, colleges are required to create timetables, which used to be an extremely time-consuming task. The "Timetable Generator" project is created with HTML and CSS for the front end and Python and MySQL 3 for the back end. This method for creating timetables connects with numerous modules and processing. The most crucial area for college efficiency is frequently automation and control.

The use of information technology for quicker and simpler forms of communication is widespread. Following information was added by the administrator for Add the student, the staff, the subject, enter the timetable, and update the timetable. Staff and students can see the details of the timetable. Python is being used to implement this system for the timetable generator. Since HTML & CSS make up our front-end, the online application has a far more effective and secure appearance. A project for a very helpful for Students to read the timetable details in this website is the timetable generator system.

This undertaking Python and SQLite have been used in the development of the online Dynamic timetable generator. A program written in Python called Dynamic Timetable Generator is used to create timetables automatically. The timetable is currently manually handled. It will facilitate automatic management of all Periods and make timetables available to instructors. Additionally, it will control the schedule when a teacher is tardy or early. For the purpose of creating a schedule effectively, the maximum and minimum workloads for each faculty member will be set for each day, week, and month. In the current system, the issue arises when a teacher is absent and is unable to notify the school or does so too late, making the manual assignment of a substitute teacher an extremely challenging task.

There is a scenario when the department head wants to make some adjustments to the lectures while the schedule tracker is made manually. Because it is impossible for one teacher to remember every task that was completed in the past, the likelihood that the teachers' periods or assignments would overlap will rise in this case. The manual upkeep of item databases and scheduling tracker processing takes time and is in some ways inaccurate. Consequently, the new system is required to address these issues.

## II. Related Work

In the past, there have been researches carried out by different groups, belonging to both, industry and academia that bear similar resemblance to or are based on a topic similar to what we are doing.

### A. Automatic Timetable Generator, 2010

D. Nguyen, K. Nguyen, K. Trieu, and N. Tran (2010), have automated the scheduling issue at universities using the Tabu search technique. In the search space that the Tabu Search Algorithm uses, viable results are seen. The search space is the set of feasible solutions to the issue. Author uses "Tabu," or basic building blocks. Tabu allows you to stop cycling and move away from non-improving motions and local optimums. The primary goal of tabu search is to avoid becoming caught at nearby maxima. For the same reason, this search allows non-improving moves when it becomes locked in local optima. Benefit of Tabu Search process prevents using memories to cycle back to the prior findings, allowing for additional development, But evaluating resources is expensive and formulating the problem is hard which is the drawback of this approach.

### B. Dynamic Timetable Generator, 2013

N. M. Hussin and A. Azlan (2013), have put into practise the graph colouring heuristic method for building process scheduling difficulty stages. Schedule challenge is addressed as a graphical representation issue. Events are arranged in chronological order using certain domain heuristics, and then they are assigned to precise time slots, ensuring that no rules are broken. Using the graph colouring method, scheduling problem is reduced to its most fundamental components. In this graph technique, a node represents a subject and an edge represents disputes. The most important component is the building stage, which results in a resident of ideal solutions. The following phase is the improvement phase, where the best possible answer is produced. This approach does not schedule soft restrictions and takes a very lengthy time to solve a problem.

### C. A Novel Approach for Automatic Timetable Generation, 2021

The proposed Automatic Timetable Generator is carried out by Ayub Sayyed, Kritika Sharma, Kush Mandal, Dipali Bhole.

The proposed system we propose to make an automated system with computer assisted timetable generator. The system will take various inputs like number of subjects, teachers, maximal lectures a teacher can conduct, priority of subject and topics to be covered in a week or a lecture, depending upon these inputs it will generate possible timetables for working days of the week, making optimal use of all resources in a way that will best suit the constraints.

### D. A Greedy-based Algorithm in Optimizing Student's Recommended Timetable Generator with Semester Planner, 2022.

The proposed Automatic Timetable Generator is carried out by Khyrina Airin Fariza Abu Samah, Siti Qamalia Thusree Ahmad Firdaus Ahmad Fadzil, Lala Septem Riza, Shafaf Ibrahim, Noraini Hasan.

In this project, a recommended semester planner using optimization of the greedy-based algorithm was used quickly to find an approximate solution in the optimization prickly.

## III. IMPLEMENTATION

Building the dynamic timetable generator involves several steps. First, we'll choose a programming language like Python and relevant libraries for data manipulation, implementing the evolutionary algorithm, and potentially building a user interface.

Next, we'll design a way to store data on classes, teachers, rooms, and user preferences. This might involve creating software objects to represent this information. Data import/export functionalities and data validation checks will also be crucial.

The system will allow users to define hard constraints (like no teacher overlap) and soft constraints (minimizing back-to-back classes). These constraints will be translated into code to be considered by the evolutionary algorithm.

The core functionality lies in the chosen evolutionary algorithm, such as a Genetic Algorithm. We'll need to define how a timetable is encoded within the algorithm (e.g., assigning classes to specific timeslots and rooms) and create a fitness function to evaluate each generated timetable. This function will consider constraint violations, resource utilization, and user preferences. Finally, genetic operators like selection and mutation will be implemented to create new generations of improved timetables.

**Admin login:**

As soon as opening the application, admin has to login with their login credentials and enter the subject details, faculty details and student details. After providing the details model develops the timetable according to number of classes, number of subjects and number faculty available for each subject. This generated timetable can be accessed by admin, students and faculty. To access this timetable students and faculty has to login with their respective credentials.

**Storing subject details:**

After entering the subject details, faculty details and student details, all the data is stored in a database and it can be accessed by admin only. No other person like faculty or student can access the data. Even if there is any change in the data, it has to be done by admin only.

**Timetable generation:**

All the details of students, faculty and subject details are stored in the database. Even the generated timetable is also stored in the database. There will be many timetables generated for multiple classes but every student or faculty will not get each and every timetable on their screen.

**Accessing timetable:**

As there are multiple timetables generated for an institute, there will be confusion and any of the faculty or student can get the timetable which does not belongs to their class or section. To overcome this, while giving the student details, their section or class name is also provided. So, the generated timetable is automatically shown to the students who belongs to the respective section.

#### **IV. RESULTS AND OUTPUTS**

- After running the file, a web page displayed in browser it asks for the login credentials of the admin.
- After successful login, a navigation bar is placed at the top which shows options to add subject details, faculty details, timetable details etc... Click on add subject details and below tab.
- After adding subject details, then add faculty details such as course name, department name, faculty name, qualification etc...
- After adding faculty details, now assign the faculty to the subjects that are added previously.
- After assigning faculty details to subjects, add timetable details such as department name, year, semester and section for which timetable is being generated and click proceed.
- After adding all the required details, click on proceed and timetable is generated as shown below.

The screenshot shows a web browser displaying the 'DYNAMIC TIMETABLE GENERATOR' interface. The course details are: Course :: B.Tech, Department :: CSE, Year :: 1, Semester :: 1, Section :: A. The timetable is a grid with days of the week as rows and time periods as columns. The subjects and activities are as follows:

Day/Period	9:00 to 9:50	9:50 to 10:40	10:40 to 11:00	11:00 to 11:50	11:50 to 12:40	12:40 to 1:40	1:40 to 2:30	2:30 to 3:20	3:20 to 3:30	3:30 to 4:20	4:20 to 5:10
Monday	English	Chemistry Lab					Problem Solving and Programming	English Lab	BREAK	Engineering Graphics Lab	Problem Solving and Programming lab
Tuesday	Chemistry Lab	English	BREAK	Problem Solving and Programming lab	Engineering Graphics Lab		Problem Solving and Programming	English Lab	BREAK	Chemistry	Algebra and Calculus
Wednesday	English Lab	Problem Solving and Programming lab	BREAK	Engineering Graphics Lab	Chemistry Lab	LUNCH	Algebra and Calculus	English Lab			
Thursday	Chemistry Lab	Engineering Graphics Lab					Engineering Graphics Lab	Problem Solving and Programming lab	BREAK	Algebra and Calculus	Problem Solving and Programming
Friday	English Lab	Problem Solving and Programming lab					Algebra and Calculus	Problem Solving and Programming lab	BREAK	Problem Solving and Programming	Chemistry Lab
Saturday	Chemistry Lab	Engineering Graphics Lab	BREAK	Chemistry	English Lab	HALF DAY					
Sunday	HOLIDAY										

Below the timetable, there is a table listing subjects and faculty names:

Subject Name	Faculty Name
Chemistry Lab	Chaitanya
Algebra and Calculus	Aayeesha
English Lab	VinayKumar
English	VIRAO
Chemistry	Vignesh
Problem Solving and Programming lab	Suneetha
Engineering Graphics Lab	ViraBrazad

Fig 1: Generated Timetable

- For faculty login, user has to select for faculty login and enter the login credentials.
- After successful login, faculty can add topic names, time and date by which they have to complete the topic.

The screenshot shows the 'DYNAMIC TIMETABLE GENERATOR' interface with the 'Add Today's Topic' form. The form includes the following fields and controls:

- Select Date:** A date picker set to mm/dd/yyyy.
- Select Time:** A time picker set to --:--:--.
- Enter Topic:** A text input field.
- Update:** An orange button to submit the form.

The page also shows navigation links: Home, Change Password, View Subject Details, Assigned Subjects, Add Daily Topic, View Topics, and Logout. The user is logged in as 'Aayeesha'.

Fig 2: Faculty Task List

## V. CONCLUSION

By removing some of the problem's dimensions and putting those dimensions into constraints, the original timetabling problem with its enormous number of binary variables has been shrunk to an appropriate size. The size of the individual was greatly decreased by combining numerous binary variables into a single gene value. The full-size problem (the problem of the entire FER schedule) can now be attempted to be solved using a genetic algorithm approach. Small size problems are resolved in tens of seconds when the scheduling problem is represented in this way, which reaches the acceptable method speed. Using intelligent operators has resulted in significant benefits. The intelligent algorithm converges significantly more quickly than the fundamental algorithm, and it serves as a suitable foundation for fully resolving the original problem. We have made significant progress, and we can say with confidence that it is highly fulfilling. We gained a solid understanding of the fundamentals of web construction as well as how evolutionary and genetic algorithms can be combined to get the best goal-finding performance in terms of time and space complexity. Regarding our comprehension of Python. Exceptional chance to understand the framework and the advantages it offers. We have made a wonderful find here. Our method of creating an automated timetable system is effective in resolving the lecture-course scheduling issue in institutions. We've also demonstrated how our timetabling system can be integrated into a rich web-based desktop environment. The graphical user interface used in this system provides an easy way in understanding how system works and also makes ease in providing the input.

## **VI. FUTURE SCOPE**

This project will be very beneficial to the university because managing numerous faculties and assigning courses to them simultaneously by hand is a very challenging task that this project will assist in managing effectively. This faculty timetable can be readily controlled while taking into account the maximum and lowest workload. The faculty data in the database can also be used to keep track of the faculty's expertise in specific fields. Attribute The accuracy of the project will allow for a more corrective approach to the creation of this schedule. This project will produce output that is mostly corrective and error-free. The project's potential future improvement is the creation of a master schedule for the departments and the entire college. Further adjustments can be made while maintaining the project's approach and methods to accomplish this improvement. Additionally, it can be utilized to assign a certain time slot that the instructor prefers. The university website may incorporate this timetable maker, making it more useful. The implementation of a timetable management system can make it simpler for the schools to assign a teacher to a class in the event of an absent teacher.

## REFERENCES

- [1] Bagul, M. R., Pushkar R. Patil, S. C., & Nagare, S. N. (October 2015). A Novel Approach for Automatic Timetable Generation . International Journal of Computer Applications
- [2] Chowdhary, A., Priyanka Kakde, S. D., & Rupal Rushiya, D. G. (February 2015). TIMETABLE GENERATION SYSTEM . International Journal of Computer Science and Mobile Computing
- [3] Deshkar, M., Mayur kale, M. B., & Ghom, A. (MARCH 2016). TIME TABLE AT A CLICK. international research journal of engineering and technology
- [4] Lahoti, Y., & Aaditya Puneekar, H. P. (2015). Automated Timetable Generator . International Journal of Science and Research
- [5] M, S., & Pranav Kiran Vaze, P. M. (MAY 2017). Automatic Time Table Generator. International Journal of Advanced Research in Computer Science and Software Engineering
- [6] Prof. Jyothi Patil, Shambhavi V, Sneha N T, Sweta Jadhav, Tahura Sadaf (2023) Automatic Timetable Generator. International Journal For Research in Applied Science and Engineering Technology.
- [7] A Study on Automatic Timetable Generator (2018), Akshay P, H M Arshad Ali Khan, Chandan S V, Parkavi A. International Journal of Science and Innovative Engineering and Technology.
- [8] Automatic Timetable Generator (2020), Shraddha Ambhore, Pooja Walke, Rohit Ghundrudkar, Akshay Alone, Anushree Khedkar. International Journal of Research in Engineering, Science and Management.
- [9] Automatic Timetable Generator (2023), Prof. Manisha. P. Navale, Aniket Marne, Omkar Pasalkar, Atharva Pawar, Shubham Yadav. International Research Journal of Modernization in Engineering Technology and Science.