ISSN (Online): 2583-648X

IJIRID

International Journal of Ingenious Research, Invention and Development

An International, High Impact Factor, Double-Blind Peer-Reviewed, Open-Access, Multidisciplinary Online Journal

Volume 3 | Issue 5 | October 2024

www.ijirid.in

A Computational Approach Towards Timetable Generation

Vaishnav Kudale[1], Parth Jangam[2], Chaitanya Khair[3], Rahul Jadhav[4], Prof. Amar Chadchankar[5]

[1,2,3,4]*Student, Zeal College of Engineering & Research, Pune (M.S.), India*

[5]*Assistant Professor, Zeal College of Engineering & Research, Pune (M.S.), India*

Abstract: Making a timetable by hand is exceedingly challenging in today's literate society. Timetables must be made specifically for each branch and each year. Making the schedules by hand becomes incredibly time-consuming, stressful, and labor-intensive. When a staff member needs to be replaced or is on leave, this procedure can occasionally become complicated. It would be practical if an algorithm generated a schedule, which would save a great deal of time and lessen the workload and stress on the employee. Time is saved by using software, which may also be used to develop schedules for complicated circumstances. Additionally, it will prevent human error such as subject conflicts and open slots.

Keywords: *Automated Timetable, Genetic Algorithm, Resource Scheduling Optimization, Component, Constraints.*

I. INTRODUCTION

It can be difficult to create timetables that work for both staff and students in educational institutions, which frequently results in mistakes and disagreements. Compiling timetables by hand takes a lot of time, is prone to errors, and produces uneven workloads. A technological option to automate and expedite this procedure is provided by the Automatic Time Table Generator. It minimizes scheduling conflicts and maximizes resource allocation through the use of data management and conflicts algorithms. For administrators, teachers, and students, the system offers an easy-to-use platform that guarantees effective scheduling and resource use. The system's characteristics, design, and the significance of conflict algorithms in enhancing scheduling effectiveness and reducing disputes are all highlighted in this paper. The Automatic Time Table Generator stands out as a crucial tool for improving educational experiences as technology continues to play a significant role in education. [1]
Lecture scheduling, which can be defined as the distribution of resources for tasks under predetermined constraints in a way that maximizes the possibility of allocation or minimizes the violation of constraints, is a crucial procedure and one of the most prevalent scheduling issues in any educational institution. A well-planned, well-executed, and conflict-free schedule is essential to the operation of an educational facility, or more broadly, an academic setting. The academic institution prepared the schedule by hand in the days before widespread use of computers. [3].
The Python-based AI timetable generator replaces manual scheduling techniques by automating the process. It allows teachers to view timetables on their phones and controls class periods. Additionally, the system modifies schedules to account for early departures, tardiness, and instructor absences. By defining the maximum and lowest number of teaching hours per day, week, or month, it optimizes the distribution of workload. By choosing a substitute teacher, dates, and length, users can request

ISSN (Online): 2583-648X

IJIRID

International Journal of Ingenious Research, Invention and Development

An International, High Impact Factor, Double-Blind Peer-Reviewed, Open-Access, Multidisciplinary Online Journal

Volume 3 | Issue 5 | October 2024

www.ijirid.in

leaves. To prevent conflicts, the system verifies the substitute's availability. It also gives faculty members in a department priority time slots, which reduces travel time between classes and boosts productivity. For administrators and instructors alike, this technology improves accessibility and expedites scheduling. [2].

Setting and controlling the timeline manually several times will be made simpler with the project's suggested system. Through the use of machine learning, the system will receive inputs such as faculty members, semester-specific subjects, and workload. A potential schedule for the working days of the week will be produced by depending on these inputs. Therefore, our suggested system will lessen the laborious task of manually creating timetables, which will help to overcome the limits that were arising in the system. [6].

II. OBJECTIVES

1. The goal of developing an effective system for autonomous timetable production is to maximize resource utilization and reduce disputes. The approach guarantees that classes, teachers, and rooms are assigned in a way that avoids overlapping bookings or unassigned resources by emphasizing conflict-free scheduling. This implies that the system can monitor instructor schedules and classroom availability, allocating resources in a way that prevents conflicts. An efficient schedule allows organizations to make the most of their resources, which lowers expenses and cuts down on classroom idle time. Furthermore, automatic timetable development reduces the amount of time spent on manual scheduling, which greatly lessens the administrative burden and facilitates prompt reaction to modifications.

2. The system enables modification according to user-defined limits and preferences to improve the scheduling experience. Conditions that users can establish include instructor availability, classroom size limitations, and specific time windows for specific sessions. This adaptability guarantees that the produced schedule satisfies the particular requirements of every institution, honoring user choices while preserving effectiveness. In order to keep the final schedule in line with important institutional objectives, the system can also give some restrictions more weight than others, such as designated days for core classes. All things considered, user-defined limits give automatic scheduling a flexible and customized feel.

3. The system also provides flexibility to handle last-minute adjustments and particular requirements in order to satisfy dynamic needs. For instance, the system can effectively modify the timetable if a teacher is unavailable or a class requires a larger space. This flexibility increases the timeline's resilience by preventing significant setbacks from disturbances. The system can proactively reduce scheduling problems by employing algorithms to identify such conflicts before they arise. A responsive, adaptable design facilitates smooth and dependable schedule management by allowing organizations to make changes in real time without interfering with the overall schedule.

ISSN (Online): 2583-648X

IJIRID

International Journal of Ingenious Research, Invention and Development

An International, High Impact Factor, Double-Blind Peer-Reviewed, Open-Access, Multidisciplinary Online Journal

Volume 3 | Issue 5 | October 2024

www.ijirid.in

4. Along with flexibility and conflict resolution, the system's ability to learn from past data gradually enhances scheduling. The system can determine trends, such the busiest hours for teachers or the periods when particular classes are most used, by examining past schedules. Then, it can modify the next schedule appropriately. In addition to streamlining the scheduling process, this data-driven strategy assists organizations in forecasting their future resource requirements, enabling them to efficiently manage their budget, staff, and space. All things considered, an automatic timetable generator facilitates a seamless and effective workflow, lowers administrative work, and customizes schedules to satisfy present and future requirements.

III. LITERATURE REVIEW

Table 1: Literature Survey Table

Sr No.	Title	Year	Objective	Methodologies	Advantages	Future Scope
1.	Automating Timetable Generation with Conflict Resolution Algorithms in Web-Based Systems for Educational Institutions [1]	2023	Automate timetable generation using conflict-resolution algorithms to optimize scheduling	Implement heuristic algorithms in a web-based platform for dynamic timetable generation	Reduces manual effort, conflicts, and improves resource allocation	Enhance UI/UX, real-time updates, predictive analytics, and system integration
2.	Survey Paper on Automatic Timetable Generator [3]	2023	Implement an algorithm to save time and reduce manual effort.	Conducted a study of various previous works.	Highlights potential drawbacks of other methodologies.	Develop an algorithm to reduce time complexity and dealing with analysed limitations
3.	Timetable Generator for Educational Institute Using Genetic Algorithm [2]	2023	Make the timetable generation process quick, more precise, and error-free, while also managing schedules efficiently.	Reduce difficulties of generating a schedule using genetic algorithm.	Faster and Accurate Schedule generation with reduced error rates.	Emphasizes on using genetic algorithm in order to create quick as well as precise timetables.
4.	An evolutionary algorithm hyper-heuristic for producing feasible timetables for the curriculum based university course timetabling problem [5]	2010	Present the result of initial attempt of timetable generation using evolutionary algorithm hyper heuristic.	The Evolutionary Algorithm hyper heuristic searches a heuristic space of combinations of lower level construction rather than a solution space.	Generalizes heuristic combinations over a set of problems instead of only developing a feasible schedule for one or more of the problems.	Examine the relationship among frequency, construction of heuristics and characteristics of problem leading to both feasible and better quality timetable.
5.	Automatic Timetable Generation using	2015	Focus on reducing time and error rates	Implement Genetic algorithm to generate timetable.	Reduces manual difficulties such as	Generate timetables that have high level of efficiency and

	Genetic Algorithm. [6]		while satisfying all soft constraints.		time consumption and errors.	precision. Improve the process of timetable generation with the help of technology.
6.	Optischedule Algorithm for Automatic Time Table Generator [4]	2024	Streamline scheduling process and optimize resource allocation resulting in better operational efficiency and organizational performance.	Combination of heuristic search techniques and metaheuristic optimization methods to iteratively improve generated timetable.	Ability to develop high quality timetables while considering complex constraints leads to minimized conflicts, flexibility and balanced workload distribution	Broader applications to other disasters like floods and hurricanes, incorporating real-time data, and improving model accuracy with additional data sources.

IV. MOTIVATION

By automating the process, the suggested approach makes the laborious effort of manually producing timetables easier. It creates an ideal plan for the working days of the week based on inputs like workload, semester-specific subjects, and faculty members. This makes it simpler to manage intricate scheduling requirements by doing away with the necessity for tedious, manual schedule alterations. The system will guarantee balanced workloads and prevent scheduling conflicts by intelligently evaluating the input data, greatly cutting down on the time and effort needed. In the end, this approach overcomes the drawbacks of creating timetables by hand and offers a dependable and practical way to better manage academic schedules.

V. PROPOSED SYSTEM DESIGN

A system architecture designed especially for creating timetables with a genetic algorithm is shown in the first image. It begins by gathering the required information, such as course numbers and faculty profiles, and then it establishes limitations, such as the weekly lecture count for each subject. After that, the algorithm creates a schedule and verifies that it complies with all requirements. The procedure is over if the limitations are satisfied. If not, it assesses any conflicts or mistakes and keeps creating new schedules until all limitations are met. The final schedule satisfies all requirements thanks to this iterative method.

Process:

1. Start: The process begins.
2. Taking necessary data as input: Collect essential information such as faculty details, course numbers, and other relevant data needed to create the timetable.
3. Setting constraints: Define the constraints, like the number of lectures per week for each subject, to ensure the timetable meets all requirements.

4. Generating a timetable using a genetic algorithm: Create an initial timetable using a genetic algorithm, which mimics natural selection to find optimized solutions.
5. Does it satisfy the constraints?: Check if the generated timetable meets all the defined constraints.
   Yes: If the constraints are satisfied, proceed to the end.
   No: If the constraints are not satisfied, evaluate the clashes or errors.
6. Evaluate clashes or errors: Identify and assess any conflicts or issues in the timetable.
7. Loop back: If there are errors, the process loops back to generating a timetable using the genetic algorithm until all constraints are satisfied.
8. End: The process concludes with a feasible timetable.

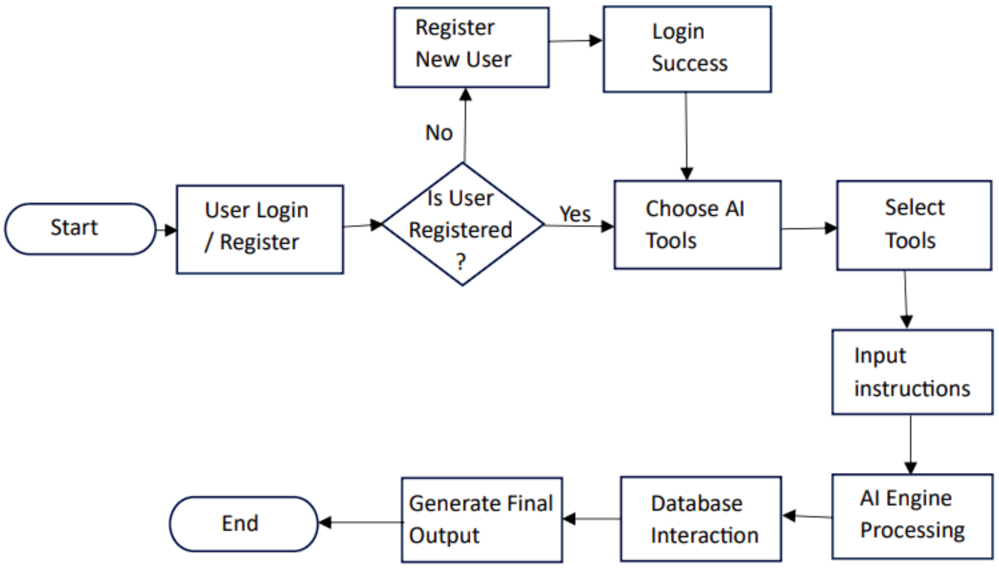

Figure 1: Proposed System Architecture for Timetable Generator

A flowchart for a genetic algorithm that creates timetables is shown in the first image. A population of possible solutions is initialized first, and then the fitness of each solution is assessed. The algorithm next determines if the halting condition which could be a maximum number of generations or a good solution is satisfied. If not, it moves on to the crossover step, where new offspring are created by combining pairs of solutions, and then to the mutation step, where diversity is maintained by introducing random alterations. The final solution is then output after this process is repeated until the halting condition is met.

Process:

1. Start: The algorithm begins its process.
2. Initialization: Initial populations of potential timetable solutions are created randomly.

3. Evaluation: Each timetable is assessed based on a fitness function, which measures how well it meets the scheduling constraints and objectives.
4. Stop Condition Check: The algorithm checks if the stopping criteria are met (e.g., a satisfactory solution is found or a maximum number of generations is reached).
   No: If the stop condition is not met, the algorithm proceeds to the next steps.
   Yes: If the stop condition is met, the algorithm moves to the output step.
5. Crossover: Pairs of timetables are combined to produce new offspring timetables, mixing their characteristics to explore new solutions.
6. Mutation: Random changes are introduced to some offspring timetables to maintain genetic diversity and avoid local optima.
7. Output: The final timetable solution is presented.

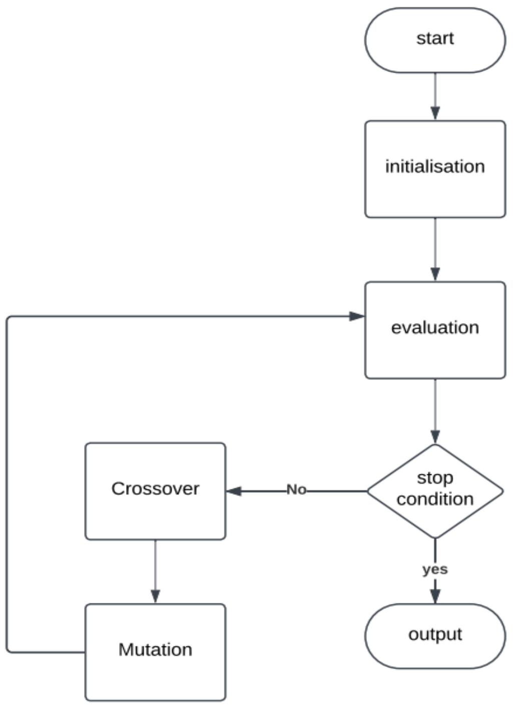

Figure 2: Proposed Flowchart for Genetic Algorithm Implementation

VI. LIMITATIONS

1. An automatic timetable generator has many benefits, but after it is put into use, certain drawbacks occur. Managing extremely complicated and overlapping limitations is a major difficulty, particularly in large schools with a wide range of course offerings. The scheduling process is more complicated the more restrictions an institution imposes, such as certain instructor time slots, room sizes, or student preferences. Sometimes the system could have trouble creating a timetable that is completely free of conflicts, or it might need to loosen some restrictions. This intricacy may result in less-than-ideal solutions where certain needs are partially

ISSN (Online): 2583-648X

IJIRID

International Journal of Ingenious Research, Invention and Development

An International, High Impact Factor, Double-Blind Peer-Reviewed, Open-Access, Multidisciplinary Online Journal

Volume 3 | Issue 5 | October 2024

www.ijirid.in

satisfied but may not be acceptable to all users. The efficiency of the schedule is limited when these limitations are balanced, especially when they contradict, and may necessitate manual modifications.

2. Another drawback is that the model might not adequately take unexpected occurrences or dynamic changes into account. Unexpected staff absences, maintenance-related room shortages, or abrupt shifts in student demand, for example, could throw off the meticulously planned schedule. Some models can respond to little changes, but frequent or large changes could necessitate a lot of manual labor. The model's dependability may be hampered by its inability to smoothly handle real-time events, which makes it difficult for institutions to continue operating efficiently. Institutions may require extra resources or assistance to manage catastrophes and other unforeseen events due to the inflexible structure of certain timetable generators.
3. Lastly, it can take a lot of resources to set up and maintain an automatic timetable generator. Accurate and current information on room sizes, instructor availability, course prerequisites, and other factors must be supplied by the institutions to the system. The generated schedule may be faulty if any of the data is out-of-date or erroneous, leading to conflicts that offset the efficiency that automation provides. Furthermore, ongoing supervision is necessary to maintain and update the model to account for institutional changes, such as new professors, courses, or classroom arrangements. Particularly for institutions with limited technical resources, this setup and maintenance effort may partially offset the time and cost benefits and may call for staff retraining or frequent technical support.

VII. EXPECTED RESULT AND CONCLUSION

Simplified scheduling, less disagreement, and better resource allocation are anticipated outcomes of putting in place an autonomous timetable generator. Institutions can expect a large decrease in conflicts between students and teachers, which will result in more efficient and successful operations. Students and staff benefit from less disruptions and more consistent scheduling because to the system's capacity to balance room capacities, instructor availability, and course demands. Staff members can concentrate on higher-value duties as a result of the administrative time saved on manual scheduling adjustments. Overall, the institution gains from more effective use of human and physical resources, fewer unused classrooms, and fewer scheduling errors that can impact student satisfaction and course delivery.

To sum up, an automatic timetable generator is a useful tool for contemporary educational establishments that improves scheduling flexibility and efficiency. Although there are certain drawbacks, such managing intricate limits and responding to changes in real time, they can be lessened with thorough preparation, frequent data updates, and the option of manual overrides as needed. In the end, the timetable generator improves overall operational effectiveness by automating a historically labor-intensive process and bringing institutional schedules closer to demands and available resources. A strong timetable generator is a first step in developing a flexible and responsive learning environment as educational institutions continue to use data-driven solutions.

REFERENCES

[1] V, Lalitha & K, Magesh & A, Selvanarayanan & G, Youkesh. (2023). Automating Timetable Generation with Conflict Resolution Algorithms in Web-Based Systems for Educational Institutions. 1-4. 10.1109/ICCEBS58601.2023.10449048.

[2] Bhatt, Chandradeep & Chaurasiya, Divyanshu & Chauhan, Rahul & Singh, Teekam & Sharma, Sanjay & Baloni, Dev. (2023). Timetable Generator for Educational Institute Using Genetic Algorithm. 19-24. 10.1109/ICTACS59847. 2023.10389824.

[3] Pounikar, Ankit & Bhandage, Hrushikesh & Dalvi, Nupur & Borade, Tanvi & Lokhande, S.. (2023). Survey Paper on Automatic Timetable Generator. International Journal of Advanced Research in Science, Communication and Technology. 620-623. 10.48175/IJARSCT-8016.

[4] Mamatha, T. & Aditya, G.V. & Meghana, S. & Anvitha, S. & Veekshitha, P.. (2024). Optischedule Algorithm for Automatic Time Table Generator. 1-7. 10.1109/ICDCECE60827.2024.10548582.

[5] Els, Rosanne & Pillay, N.. (2010). An evolutionary algorithm hyper-heuristic for producing feasible timetables for the curriculum based university course timetabling problem. Proceedings - 2010 2nd World Congress on Nature and Biologically Inspired Computing, NaBIC 2010. 460-466. 10.1109/NABIC.2010.5716340.

[6] Mittal, Dipesh & Doshi, Hiral & Sunasra, Mohammed & Nagpure, Renuka. (2015). Automatic Timetable Generation using Genetic Algorithm. IJARCCE. 245-248. 10.17148/IJARCCE.2015.4254.