# OPTI - SCHEDULING

Sumith Yadav. N
*B. Tech*
*School of Engineering*
Computer Science-(AI&ML)
Malla Reddy University, India

SuryaKiran.K
*B. Tech*
*School of Engineering*
Computer Science-(AI&ML)
Malla Reddy University, India

Suryam. M
*B. Tech*
*School of Engineering*
Computer Science-(AI&ML)
Malla Reddy University, India

Swathi Reddy. R
*B. Tech*
*School of Engineering*
Computer Science-(AI&ML)
Malla Reddy University, India

Swejana. G
*B. Tech*
*School of Engineering*
Computer Science-(AI&ML)
Malla Reddy University, India

Swetha. G
*B. Tech*
*School of Engineering*
Computer Science-(AI&ML)
Malla Reddy University, India

Guide: Prof A. Siva Kumar
Assistant Professor
School of Engineering
Computer Science-(AI&ML)
Malla Reddy University,India

***1. Abstract:*** *In modern era time management is one of the most important in everyone's life. A timetable can be defined as an arrangement of schedules in various time slots for effective and efficient time management. Timetable generation is a difficult task but it is very important in educational institutions. This paper presents to generating a smart timetable using a genetic algorithm(GA). It utilizes genetic algorithm for the generation of a weekly timetable depending on various factors like time, subjects and different faculties. Moreover in many of the colleges and schools timetable is prepared manually which takes more time and effort, by using genetic algorithm we able to reduce time and effort which is more accurate and free of human errors.*

***Keywords:*** Python, Machine Learning(Genetic Algorithm), Django

## 2. INTRODUCTION

In recent years, the application of genetic algorithms has emerged as a powerful and innovative approach for timetable generation. Inspired by the principles of natural selection and evolution, genetic algorithms mimic the process of genetic recombination and mutation to iteratively improve solutions over successive generations. Traditional methods often face challenges in handling the intricacies of scheduling, leading to suboptimal solutions.

The maximum and minimal workload for a faculty for an afternoon, week and month could be exact for the effective generation of timetables . By the usage of this software program, users can exercise for leave through imparting go away needed date. In this system we are using Machine learning and AI for optimization of the system. It is suitable to give extra Burden to another person effectively.

## 3. LITERATURE REVIEW

This paper explores the application of genetic algorithms for the optimization of timetable scheduling in educational institutions. It discusses the challenges of timetable generation, emphasizing the complexity arising from various constraints. The genetic algorithm is proposed as a suitable solution due to its ability to handle optimization problems with multiple constraints. The study includes a detailed analysis of the algorithm's performance in comparison with traditional methods.

The study provides insights into the effectiveness of genetic algorithms in finding near-optimal solutions for large-scale timetable scheduling problems .The study

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 12 | December - 2023 SJIF Rating: 8.176 ISSN: 2582-3930

evaluates the algorithm's performance using real-world data and discusses the results, demonstrating the potential of genetic algorithms in handling complex scheduling problems. Scaling the system to handle a large number of academic departments, courses, or campuses may affect the efficiency of the timetable generation. The system's adaptability may be limited in cases where there are frequent changes in scheduling requirements that were not initially considered during the algorithm design.

### *3.1 Existing System:*

The current timetable generation system relies on traditional methods and heuristic approaches to address the complexities inherent in scheduling within educational institutions. Unlike the genetic algorithm-based system, this existing solution may follow a deterministic or rule-based approach, attempting to optimize schedules without leveraging evolutionary principles. The system likely involves a set of predefined rules and constraints to allocate courses, instructors, and resources. Data pre-processing steps would focus on cleaning and organizing input data, ensuring consistency and accuracy.

However, its reliance on fixed rules and algorithms may limit its ability to handle intricate scheduling scenarios and adapt to changes in real-time. The absence of data augmentation techniques, such as randomized initialization and dynamic mutation rates, might lead to a higher susceptibility to converging to suboptimal solutions. Despite these limitations, the existing system provides a benchmark for comparison with more advanced genetic algorithm-based approaches, shedding light on the potential advantages and challenges associated with each methodology.

### *3.2 Limitations:*

Computational Intensity: The computational intensity of genetic algorithms, particularly in the evaluation of fitness functions, can be demanding. This might hinder their efficiency when dealing with large datasets or complex constraints.

Difficulty in Handling Dynamic Changes: Genetic algorithms may struggle to adapt quickly to dynamic changes in scheduling requirements. Real-world scheduling scenarios often involve unexpected alterations, and GAs might require further enhancements to handle dynamic environments seamlessly.

Dependency on Operator Selection: The performance of genetic algorithms is highly dependent on the choice of crossover and mutation operators. Inappropriately selected operators may lead to premature convergence or slow convergence, affecting the quality of generated timetables.

### *3.3 Proposed System*

Our project will help to optimize the timetable and will override any time period. It is efficient. Intend to display the Timetable as per the faculty is been allotted for the respective subject in according to the information given. Implement visualization techniques to interpret and present the generated timetables. This system uses Python Libraries and also machine learning for optimization.

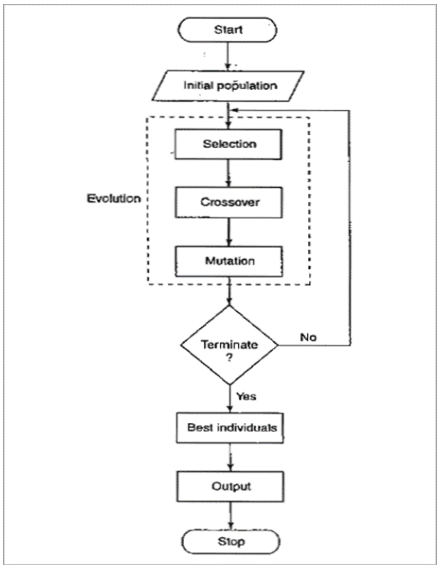


Figure 1: Working of Genetic Algorithm

## 4. PROBLEM STATEMENT

In the context of timetable generation using machine learning, the problem centers around the need for an intelligent and adaptable system that efficiently schedules courses, instructors, and resources within educational institutions. The conventional approaches often struggle to handle the intricacies of diverse constraints and preferences, leading to suboptimal scheduling outcomes. The aim of the machine learning component is to enhance the efficiency and effectiveness of timetable generation through the utilization of genetic algorithms.

### *4.1 Description of data:*

The data used in this project encompasses various aspects of educational scheduling using:

1) *Course Information:* Details about courses offered, including course codes, titles, and required resources.

2) *Choosing Department:* Details about which department must be chosen for certain course.

3) *Instructor Preferences:* Preferences and availability of instructors, including preferred time slots, teaching load constraints, and specific courses they are willing to teach.

4) *Room Availability:* Information regarding the availability and capacity of rooms or venues for conducting classes.

5) *Student Constraints:* Constraints related to student preferences, such as preferred time slots, course preferences, and any specific constraints affecting individual students or groups.

6) *Home:* Once the information is filled go to home page and click on generate timetable the time table will be generated.

## 5. METHODOLOGY

The smart timetable system uses machine learning algorithms. This is a genetic algorithm used to solve complex problems with more variables and possible outcomes or solutions. Bad solutions are replaced by good solutions. To make a timetable system generic that is efficient by the use of Django. It will focus on optimization of resources using ML. It also shows the view of timetable as we want by using Database (SQL).

### *5.1 Genetic Algorithm*

This algorithm was invented by John Holland. He wrote a book on genetic algorithms entitled Adaptation to Natural and Man-Made Systems. Genetic algorithms are based on evolutionary algorithms. This algorithm uses the principle of natural collection to create and develop the optimal solution as a result. A genetic algorithm is also a heuristic search with natural evolutionary features such as mutation, inheritance, crossover, and selection to generate solutions that optimize a problem to get the right solution. Genetic algorithms are often used in developing planning systems.

When planning a schedule, there may be many solutions that do not violate the constraints. So, using a genetic algorithm gives us a good pool of solutions. Evolutionary features such as mutations and crossovers make this algorithm more efficient and less time consuming to search.

### *5.2 Model Architecture:*

1) *Initialization:* Generate an initial population of schedules, each representing a potential solution to the timetable generation problem.

2) *Fitness Assignment:* Define a fitness function to evaluate the quality of each schedule. The function considers constraints such as room availability, instructor preferences, and student constraints.

3) *Selection:* Select schedules from the population based on their fitness scores using tournament selection. Higher-ranked schedules have a higher chance of being selected.

**4)** *Reproduction:* Creating a new generation of individuals through processes such as crossover and mutation.

5) *Termination:* Determine whether the algorithm has converged to an optimal or satisfactory solution. Terminate if convergence criteria are met.

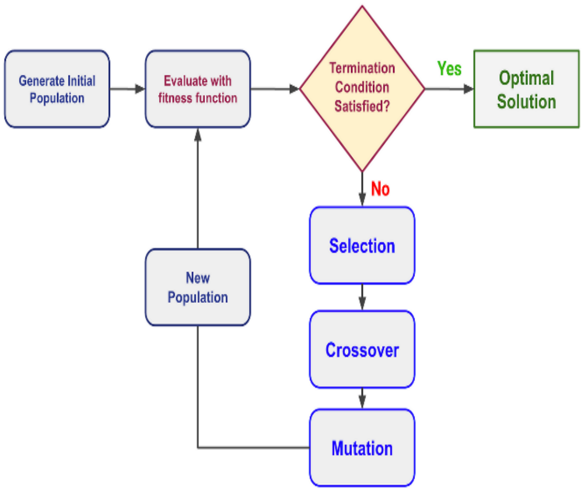


Figure 2: Model Architecture

### *5.3 Pre-Processing steps:*

- Clearly define the constraints and objectives of the timetable. Constraints may include room availability, teacher availability, and class restrictions.

- Each chromosome in the genetic algorithm population could represent a possible timetable.

- Implement visualization techniques to interpret and present the generated timetables.

- Design a fitness function that evaluates how well a timetable satisfies the given constraints and objectives.

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 12 | December - 2023 SJIF Rating: 8.176 ISSN: 2582-3930

*5.4 Data Augmentation:*

1) *Randomized Initialization:* Introduce randomness in the initial population generation to diversify the starting solutions. This prevents the genetic algorithm from converging prematurely to suboptimal solutions.

2) *Dynamic Mutation Rates:* Implement dynamic mutation rates based on the performance of schedules in previous iterations. If certain schedules consistently fail to meet constraints, increase their mutation rates to explore alternative solutions.

3) Instructor and Room Swapping: Implement swapping operations during crossover to exchange instructors or rooms between schedules.

## 6. EXPERIMENTAL RESULTS

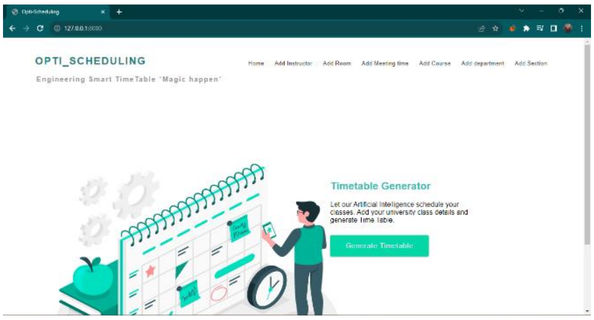


Figure 3: Home page

In home page we can see certain options given Add instructor, Add Course, Add Department, Add meeting time, Add room, Add Section after filling all the details, at home page click on generate time table.

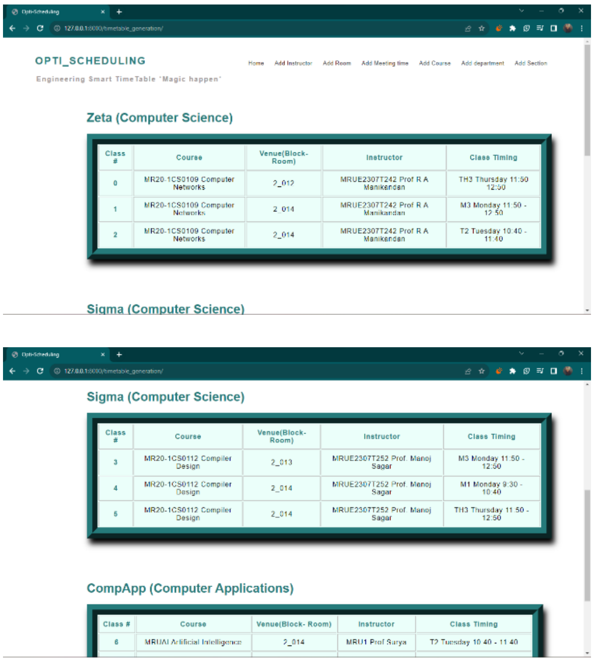


Figure 4: Final Results

*6.1 Model Evaluation and Metrics:*

- After the timetable is generated, its quality is assessed through a set of key metrics.

- The 'Convergence Rate' evaluates how quickly the algorithm reaches a satisfactory solution.

- We gauge the 'Computational Efficiency' to ensure that the generation process is fast and scalable, accommodating varying dataset sizes.

- With these metrics, we guarantee not only high-quality schedules but also a swift and reliable timetable generation experience for users on the website.

*6.2 Comparision to existing methods:*

1) *Adaptability:* The genetic algorithm-based approach is inherently adaptable. It can dynamically adjust to changes in scheduling requirements and explore diverse solution spaces, making it well-suited for handling complex constraints and preferences.

2) *Optimization Capabilities*: The genetic algorithm optimizes schedules by evolving a population of solutions over multiple generations. It leverages mechanisms such as crossover, mutation, and selection to converge towards optimal or near-optimal solutions.

3) *Scalability:* The genetic algorithm approach incorporates data augmentation techniques such as randomized initialization, dynamic mutation rates, and ensemble scheduling, enhancing its exploration of solution spaces and preventing premature convergence.

## 7. CONCLUSION

In conclusion, the dynamic landscape of timetable generation within educational institutions calls for advanced methodologies that can effectively navigate the intricacies of diverse constraints, preferences, and dynamic scheduling environments. While traditional systems rely on deterministic and rule-based approaches, they often face limitations in adaptability, scalability, and optimization capabilities.

By adopting an evolutionary methodology inspired by natural selection, genetic algorithms provide a flexible and adaptive framework for optimizing schedules.

The utilization of data pre-processing, feature engineering, and advanced data augmentation techniques, such as randomized initialization and dynamic mutation rates, enhances the genetic algorithm's ability to explore diverse solution spaces, preventing premature convergence to suboptimal schedules.

## 8. FUTURE WORK

1. *Hybrid Models:* Investigate the integration of genetic algorithms with other optimization techniques or machine learning models. Hybrid models could combine the strengths of genetic algorithms with algorithms such as simulated annealing, particle swarm optimization, or reinforcement learning to achieve improved optimization performance.

2. *Dynamic Adaptation to Real-Time Changes:* Enhance the system's capability to dynamically adapt to real-time changes in scheduling requirements. Implement mechanisms that allow the genetic algorithm to efficiently adjust schedules in response to unexpected events, resource availability changes, or modifications in institutional constraints.

3. *Customization for Different Educational Institutions:* Develop a more customizable and configurable genetic algorithm system that can be easily adapted to the specific constraints and preferences of different educational institutions. This could involve the creation of user-friendly interfaces and tools for institutions to define their own optimization criteria and constraints.

4. *Machine Learning Integration:* Explore the integration of machine learning techniques for more adaptive and intelligent timetable generation. Use historical scheduling data to train models that can dynamically adjust parameters, such as crossover and mutation rates, based on the success or failure of past schedules.

## 9. REFERENCES

[1] Leon Bambrick, Lecture Timetabling Using Genetic Algorithms

Available: http://secretgeek.net/content/bambrilg.pdf

[2] Burke, E.K. and Petrovic, S., 2002. Recent research directions in automated timetabling European Journal of Operational Research, 140(2), pp.266Y280.

[3] Chowdhary A., Kande, P., Dhone, S., Ingle, S., Rushiya, R. And Gawande, D.,2014 Timetable Generation system. International Journal of Computer Science and Mobile Computing, 3(2).

[4] Bhaduri, A., 2009, October. University timetable scheduling using genetic artificial immune network. In Advances in Recent Technologies in Communication and Computing, 2009. ARTCom'09. International Conference on (pp. 289Y292). IEEE.

[5] Prashanta Kumar, Shreedhar Sanakar, Praveen kumar, Syed Muhammad Usman, Vani K A, "Automated timetable generator using machine learning" Aug 2020 www.irjmets.com