IJFANS INTERNATIONAL JOURNAL OF FOOD AND NUTRITIONAL SCIENCES
ISSN PRINT 2319 1775 Online 2320 7876
 UGC CARE Listed (Group -I) Journal Volume 11, Iss 11, 2022

SCHEDULERA NEXT GEN SCHEDULING FOR TIMETABLE GENERATION USING GENETIC ALGORITHM

[1]Venkata Kishore Puli, [2]JanapriyaNagapuri, [3]MeeraAlphy, [4]MareeduSunitha
[1,2,3,4]Department of Computer Science and Engineering, St. Peter's Engineering College, Hyderabad, Telangana, India
E-Mail: pvkishorehod@gmail.com

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

1. INTRODUCTION

The timetable generator, powered by a genetic algorithm, provides an efficient way to organize schedules for classes, exams, and events within a given timeframe. It takes into account various constraints, such as the availability of teachers and rooms, capacity limitations, and specific sequence requirements. The goal is to create a timetable that minimizes idle gaps, distributes workloads fairly, and respects individual preferences. In this system, timetables are modelled as chromosomes, where each gene represents an activity, its assigned teacher, and the room allocation. The genetic algorithm improves the schedule by simulating natural evolution, using processes like selection, crossover, and mutation to enhance fitness. This iterative process continues until a stopping point is reached—whether it's a maximum number of generations or achieving a desirable fitness level. By addressing the complexities of scheduling, this method provides a structured and effective way to optimize timetables while meeting diverse requirements.

Using a genetic algorithm for timetable generation can sometimes become computationally demanding, particularly when managing complex scheduling constraints. This may lead to longer processing times. While the algorithm often produces near-optimal solutions, it doesn't always guarantee the perfect or most ideal timetable.Expanding the system to accommodate a large number of academic departments, courses, or campuses can also impact its efficiency, potentially slowing down the generation process. Additionally, the system's flexibility might be limited if there are frequent, unforeseen changes in scheduling requirements that were not accounted for during the initial design of the algorithm.

In recent years, genetic algorithms have gained recognition as a powerful and innovative tool for timetable generation. Drawing inspiration from the principles of natural selection and evolution, these algorithms simulate processes like genetic recombination and mutation to progressively refine solutions across multiple generations. Unlike traditional scheduling

methods, which often struggle with complex constraints, genetic algorithms provide a more dynamic and effective approach to achieving better results.For optimal timetable creation, it's essential to define the maximum and minimum workload for faculty members over specific time frames, such as an afternoon, a week, or a month. This ensures a balanced distribution of tasks and prevents overburdening. Additionally, the system allows users to conveniently request leave by specifying the desired dates. By leveraging machine learning and artificial intelligence, this system enhances optimization and adaptability. It also ensures that additional workload can be redistributed efficiently among other faculty members, maintaining fairness and effectiveness.

2. LITERATURE REVIEW

This paper delves into the use of genetic algorithms for optimizing timetable scheduling in educational institutions. It highlights the challenges involved in creating efficient schedules, particularly due to the numerous constraints that need to be addressed. Genetic algorithms are presented as an effective solution, capable of tackling complex optimization problems involving multiple constraints. The study also provides a comprehensive evaluation of the algorithm's performance, comparing it with traditional scheduling methods to demonstrate its advantages.

This study offers valuable insights into the effectiveness of genetic algorithms in solving large-scale timetable scheduling challenges. By evaluating the algorithm's performance with real-world data, the research highlights its potential for managing complex scheduling scenarios. The findings demonstrate how genetic algorithms can efficiently handle intricate constraints and deliver near-optimal solutions.However, scaling the system to accommodate numerous academic departments, courses, or campuses may impact its efficiency, potentially leading to longer processing times. Additionally, the system's adaptability might be restricted when faced with frequent or unforeseen changes in scheduling requirements that were not accounted for during the initial design of the algorithm.

2.1 Existing System:

The current timetable generation system primarily uses traditional methods and heuristic approaches to tackle the complexities of scheduling in educational institutions. Unlike systems based on genetic algorithms, these conventional methods often rely on deterministic or rule-based processes to optimize schedules. The approach typically involves a predefined set of rules and constraints to assign courses, allocate instructors, and manage resources.In such systems, data pre-processing plays a critical role, focusing on cleaning and organizing input data to maintain consistency and accuracy. However, without incorporating evolutionary principles, these methods may face limitations in efficiently handling the dynamic and intricate nature of scheduling challenges.

While the system's reliance on fixed rules and algorithms helps maintain structure, it may struggle to manage complex scheduling scenarios or adapt effectively to real-time changes. The lack of advanced techniques, such as randomized initialization or dynamic mutation rates, increases the likelihood of settling on suboptimal solutions. Despite these challenges, the existing system serves as a valuable benchmark for comparing traditional methods with

more advanced genetic algorithm-based approaches. This comparison highlights both the strengths and limitations of each methodology, offering insights into the potential benefits and challenges of adopting evolutionary algorithms for scheduling.

2.2 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.

2.3 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.

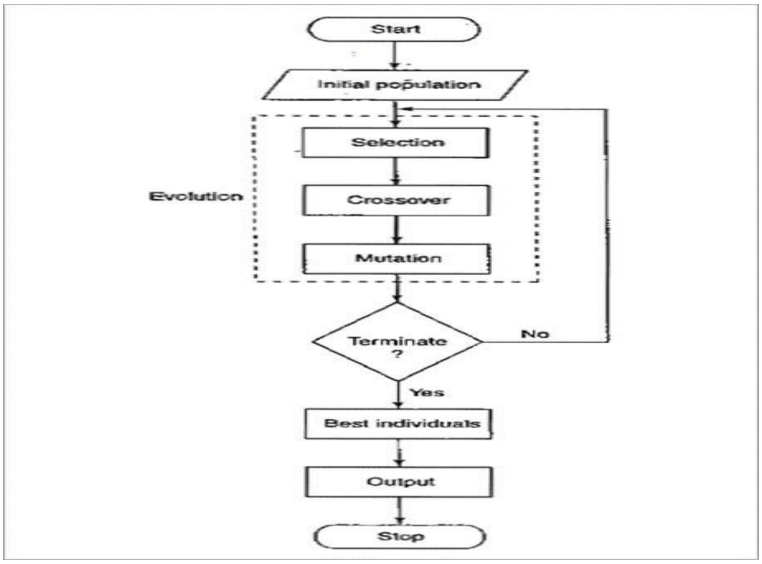


Fig 1: Working of Genetic Algorithm

3. PROBLEM STATEMENT

Timetable generation using machine learning focuses on creating an intelligent and adaptable system capable of efficiently organizing courses, instructors, and resources in educational institutions. Traditional methods often fall short when addressing the complexities of diverse constraints and individual preferences, resulting in less-than-ideal schedules. By

ISSN PRINT 2319 1775 Online 2320 7876
 UGC CARE Listed (Group -I) Journal Volume 11, Iss 11, 2022

incorporating genetic algorithms, the machine learning component aims to significantly improve the efficiency and effectiveness of timetable generation, offering a more robust solution to these challenges..

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.

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

4. METHODOLOGY

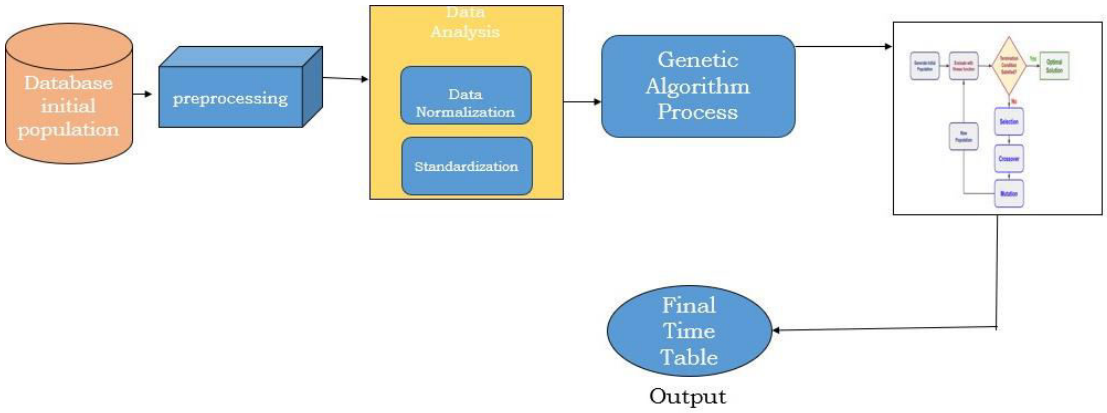


Fig 2 System Architecture

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).

4.1 Genetic Algorithm:

This algorithm was invented by John Holland. Hewrote 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.

4.2 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.

4.3 Pre-Processing steps:

1. Clearly define the constraints and objectives of the timetable. Constraints may include room availability, teacher availability, and class restrictions.
2. Each chromosome in the genetic algorithm population could represent a possible timetable.
3. Implement visualization techniques to interpret and present the generated timetables.
4. Design a fitness function that evaluates how well a timetable satisfies the given constraints and objectives.

4.4 Data Augmentation:

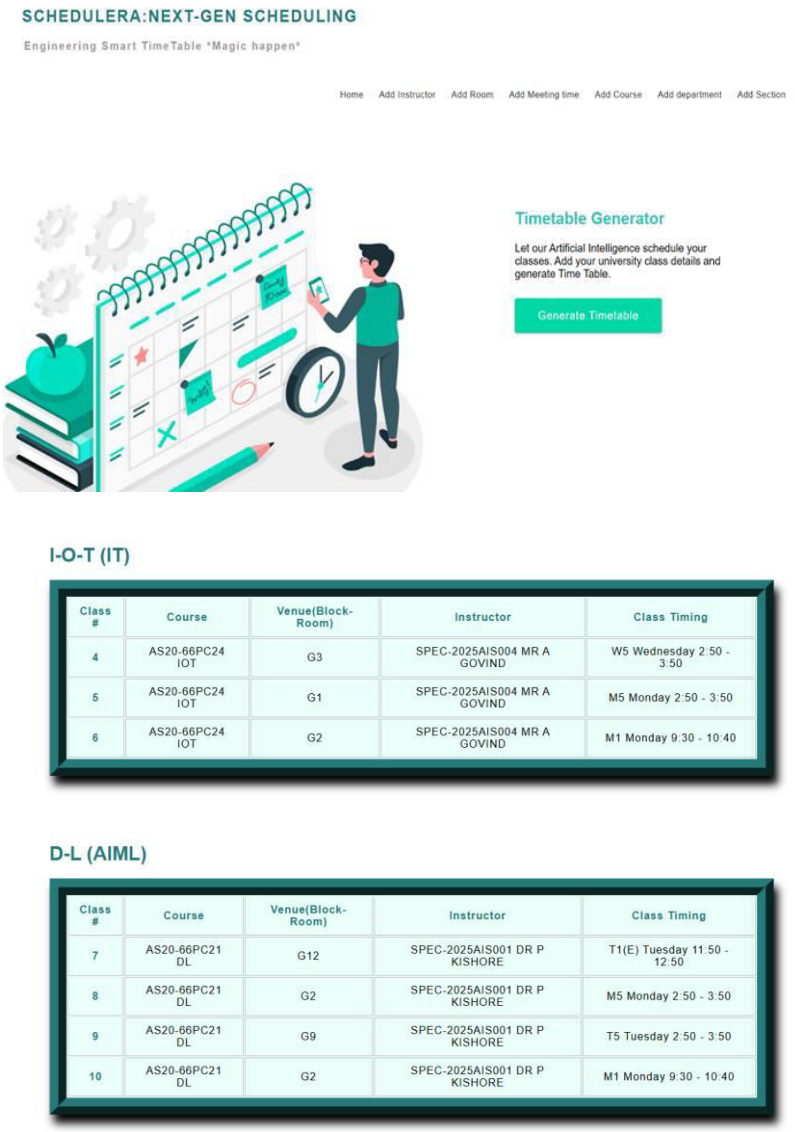
Randomized Initialization*:* Introduce randomness in the initial population generation to diversify the starting solutions. This prevents the genetic algorithm from converging prematurely to suboptimal solutions.
Dynamic Mutation Rates: Implement dynamic mutation rates based on the performance of schedules in previous iterations. If certain schedules consistently fail to meet constraints, increase their mutation rates to explore alternative solutions.
Instructor and Room Swapping: Implement swapping operations during crossover to exchange instructors or rooms between schedules.

5. RESULT

The results of this project depend on the Input parameters given by the faculty for the respective class.

ISSN PRINT 2319 1775 Online 2320 7876
 UGC CARE Listed (Group -I) Journal Volume 11, Iss 11, 2022

For Implementation of the code the below are the parameters i have given to the model as an example:



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 highquality 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 algorithmbased 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

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.

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.

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.

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.

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.co
6. M. Welling, "A First Encounter with Machine Learning"
7. P. Harrington, "Machine Learning in action", Manning Publications Co., Shelter Island, New York, 2012 [Online]. Available: www.analyticsvidhya.com
8. S.B. Kotsiantis, "Supervised Machine Learning: A Review of Classification Techniques", Informatica 31 (2007) 249-268
9. Taiwo, O. A. (2010). Types of Machine Learning Algorithms, New Advances in Machine Learning, Yagang Zhang (Ed.), ISBN: 978-953-307-034-6, InTech, University of Portsmouth United Kingdom. Pp 3 – 31. Available at InTech open website: http://www.intechopen.com/books/new-advances inmachine-learning/types-of-machine-learning-algorithms
10. M. D. Boomija, R. Ambika, J. Sandhiya, P. Jayashree, "Smart and Dynamic Timetable Generator", International Journal for Research in Applied Science and Engineering Technology, March 2019.
11. V. Abhinaya, K. Sahithi, K. Akaanksha, "Online Application of Automatic Timetable generator", International Research Journal of Engineering and Technology, February 2019.
12. Akshay Puttaswamy, H. M. Arshad Ali Khan, Chandan S. V, Parkavi. A, "A Study on Automatic Timetable Generator", International Journal of Science and Innovative Engineering and Technology, May 2019.
13. Adithya R Pai, Ashwitha S, Raksha Shetty, Geethalaxmi, "Automated college timetable generator", International Journal of Scientific & Engineering Research, vol. 9, no. 4, April 2018.
14. Saritha M, Pranav Kiran Vaze, Pradeep, Mahesh NR, "Automatic time table generator", International Journal of Advanced Research in Computer Science and Software Engineering, May 2017.