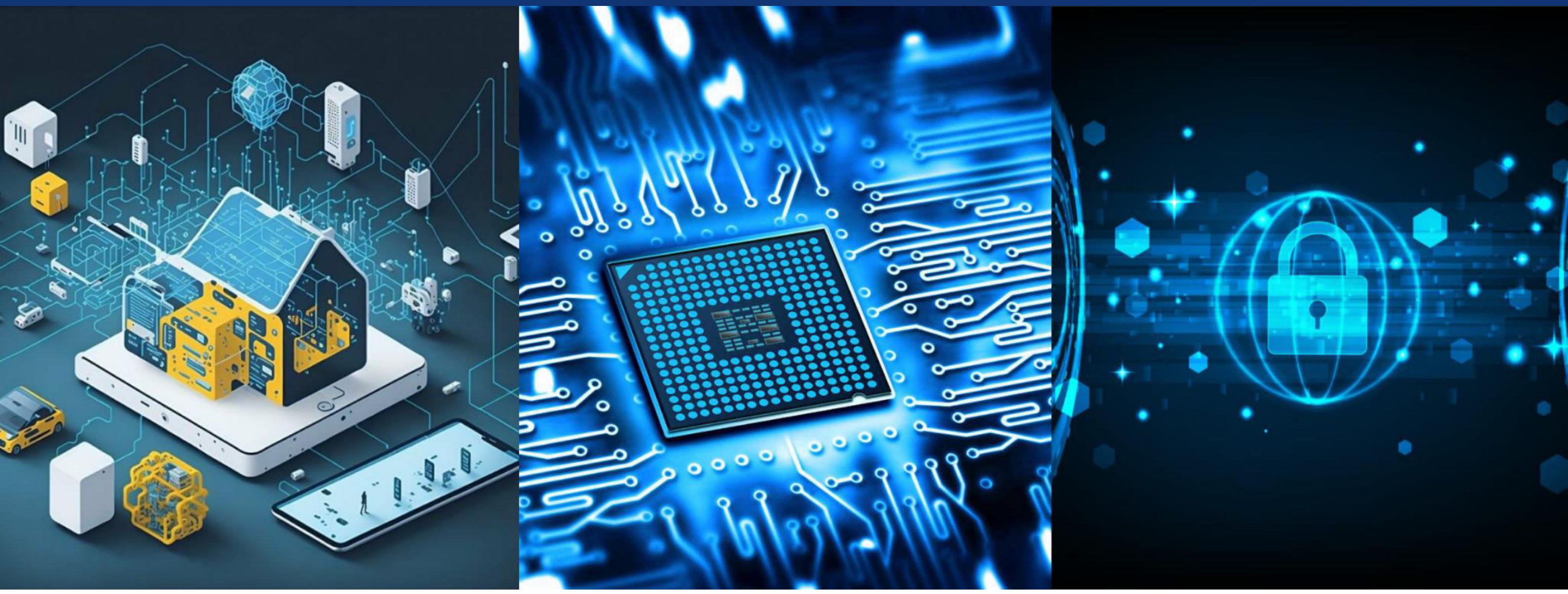
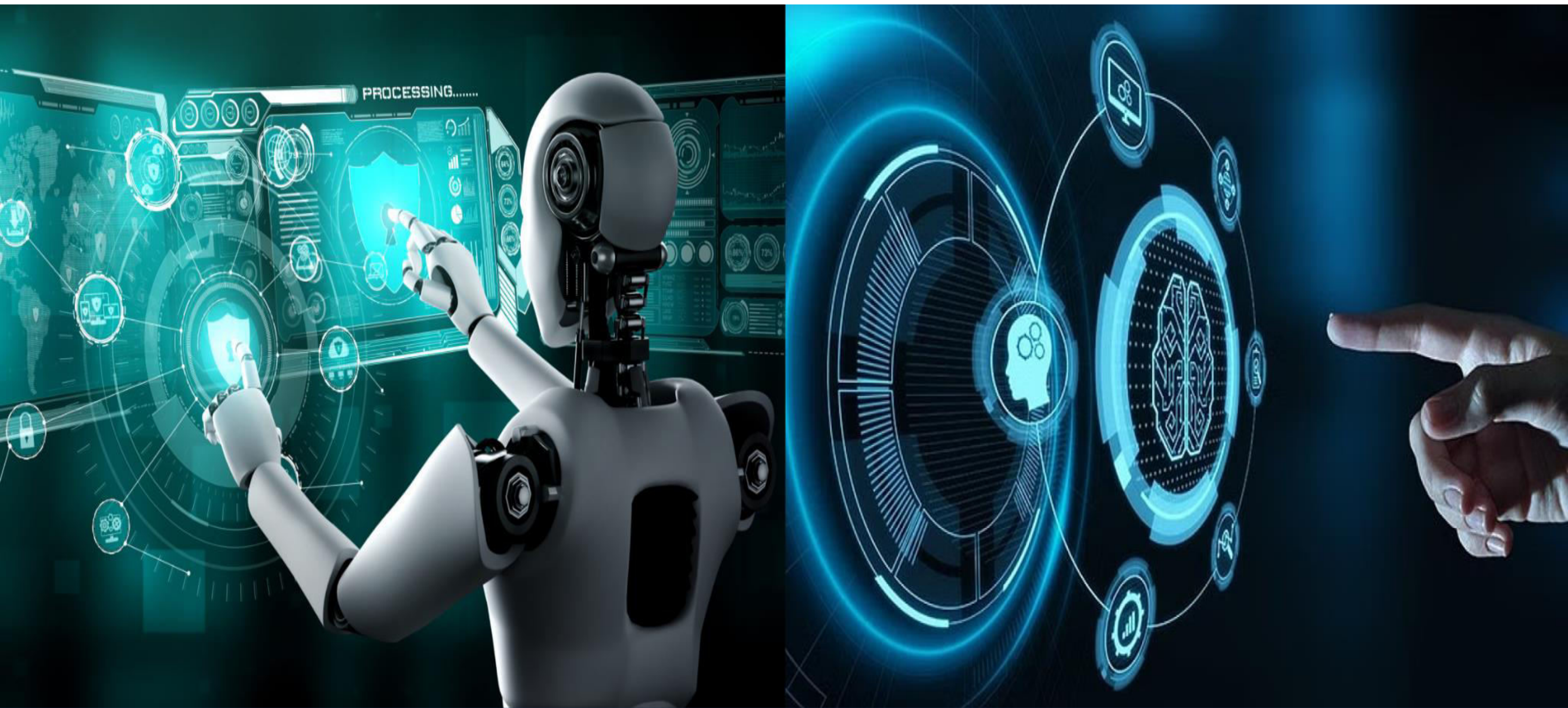
ISSN(O): 2320-9801

ISSN(P): 2320-9798



International Journal of Innovative Research in Computer and Communication Engineering

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.625

Volume 13, Issue 1, January 2025

www.ijircce.com | ijircce@gmail.com | +91-9940572462 | +91 63819 07438

© 2025 IJIRCCE | Volume 13, Issue 1, January 2025| DOI: 10.15680/IJIRCCE.2025.1301067

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Examination Time Table Generation

Dr. Bhavana A, Manur Yashas Sreevatsa, Priya K R, Anirudh Manjunath Sandilya, Perepi Bhanu Vaishnavi

Assistant Professor, School of Computer Science and Engineering, Presidency University, Bengaluru, India

UG Student, School of Computer Science and Engineering, Presidency University, Bengaluru, India

ABSTRACT: The purpose of this project is to create an automated exam scheduling system for Presidency University, dealing with the logistical difficulties in a manner that enhances fairness and well-being among faculty and students. The system utilizes technologies like Flask, MySQL, Python, and constraint programming libraries such as OR-Tools to consider the input of faculty availability, elective preferences, and course registrations to ensure fair distribution among faculty members and avoid overlapping exams while limiting daily exam times. Featuring an intuitive interface, the tool effectively allocates resources and optimizes schedules. The initiative reduces administrative burdens and ensures fairness, thereby setting a benchmark for exam scheduling in academic institutions.

The program provides an easy-to-use interface for entering data and creates optimum schedules with little assistance from the user. Elective courses with shared enrolments are allocated to distinct slots, faculty resources are efficiently used, and students are not scheduled for overlapping exams.

The objectives of this effort are to improve equity and well-being, simplify the administrative burden, and expedite the examination process. The results will establish a standard for scheduling in educational establishments, strengthening the academic experience for all parties involved at Presidency University and assisting in the effective and seamless administration of exams.

KEYWORDS: Automated exam scheduling, Presidency University, fairness, faculty availability, course registrations, constraint programming, OR-Tools, optimized schedules, administrative burdens, intuitive interface.

I. INTRODUCTION

Efficient scheduling of exams is the need of academic administration, more so in a diverse institution like Presidency University. This project introduces an automated timetable generator to address the complexities of faculty schedules, course registrations, and elective overlaps that manual methods cannot handle efficiently. Advanced optimization techniques like Constraint Programming and Genetic Algorithms are used with modern technologies such as Flask, ReactJS, and MongoDB. The system avoids conflicts, being fair and capping its exam conduct time in a day for well-being. It is intuitive, allowing easy data entry, conflict resolution, and even customization in schedule, thus improving operational efficiency and equity while giving out a benchmark for scalable academic scheduling solutions.

This research presents a novel solution to these problems: an automated examination schedule generator. In order to meet the complex requirements of test scheduling, this system makes use of state-of-the-art optimization techniques and contemporary technology. The solution guarantees that all factors, including course registrations, elective choices, faculty availability, and student preferences, are taken into consideration by using data-driven algorithms. To ensure fairness and accessibility for all stakeholders, the system is further improved with limits to prevent typical problems like overlapping tests, faculty fatigue, and invigilation overloads.

The utilization of cutting-edge programming tools and optimization libraries, along with an intuitive user interface, form the foundation of this project and enable administrators and faculty members to access the system. The software streamlines the procedure by enabling smooth data entry, automatic dispute resolution, and an easy-to-understand schedule display. Additionally, it offers the freedom to modify the schedule as necessary, guaranteeing that the finished schedule complies with the operating standards and regulations of the university.

II. RELATED WORK

Exam scheduling is a very difficult combinatorial optimisation problem, balancing many competing constraints, such as room availability and faculty and student schedules with conflict such as exam clashes. Many approaches have been tested with genetic algorithms, including constraint satisfaction models and even hybrid methods. Optimizations such as simulated annealing and metaheuristics provide quality solutions but usually require much computation, whereas heuristic methods focus on speed and practicality for real-time applications. The recent development of CSPs and hybrid models enhances efficiency and adaptability, which is an area of ongoing research with significant practical importance for equitable and effective scheduling in academic institutions.

III. PROPOSED ALGORITHM

3.1. Open AI Algorithm

The AI-based schedule generator optimizes and automates scheduling for educational institutions by utilizing the concepts of a Genetic Algorithm (GA). The system guarantees a smooth schedule generating process by assessing important inputs such course registrations, instructor availability, classroom capacity, and elective selections. Each potential schedule is viewed as a "chromosome," and different restrictions serve as "fitness functions" to assess the quality of each solution. This approach views scheduling as a sophisticated optimization problem. The GA uses genetic operators like as crossover, mutation, and selection to iteratively optimize schedules.

The system collects administrator input in the first step, which includes student registrations, faculty preferences, course lists, and classroom supplies. A population of possible schedules, each representing a distinct combination of the input parameters, is initialized by the Genetic Algorithm. These timetables are then assessed in light of limitations such as classroom availability, student group conflicts, and instructor weariness. Less ideal alternatives are either eliminated or improved, while schedules that satisfy more requirements and score higher on optimization measures are kept for the following iteration.

It ensures a varied pool of solutions by combining and modifying schedules to create new ones through crossover and mutation procedures. Last-minute modifications, like teacher absences or new course enrollments, are dynamically incorporated by the Genetic Algorithm. This flexibility enables the system to generate effective and conflict-free schedules while taking equity in resource distribution and staff and student interruptions into account.

The schedule generator achieves scalability, precision, and efficiency through the use of a genetic algorithm. It guarantees fair allocation of classes and tests, removes manual scheduling mistakes, and saves administrative work. The system is perfect for both small and large colleges since it can adjust to real-time inputs and settle issues. The application offers a reliable and adaptable solution to academic scheduling issues by integrating AI-driven optimization strategies.

3.2. Genetic Algorithm.

Darwin's theory of natural selection served as the inspiration for genetic algorithms (GAs), which are optimization approaches used to address challenging issues including resource allocation, scheduling, and routing. These algorithms iteratively improve possible solutions (individuals) within a population, simulating biological evolution. Because GAs don't need gradient information and work well in a variety of problem domains, they are very good at handling non-linear and multi-modal optimization problems. Population initialization, fitness assessment, selection, crossover (recombination), and mutation are the fundamental elements of GAs.

A wide range of possible solutions are produced at random during the Population Initialization stage. Every solution (or chromosome) is assessed according to a fitness function that establishes how well it meets the goals and limitations of the challenge. In order to ensure that the best features are passed on to future generations, high-fitness solutions are then chosen using techniques such as Tournament Selection or Roulette-Wheel Selection. By combining desired qualities, crossover—the process of genetic material exchange between chosen solutions—creates new progeny. Single-Point and Uniform Crossover are popular crossover techniques that enable the independent or targeted exchange of genes.

© 2025 IJIRCCE | Volume 13, Issue 1, January 2025| DOI: 10.15680/IJIRCCE.2025.1301067

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

In order to preserve diversity and avoid an early convergence to less-than-ideal solutions, mutations add random modifications to their progeny. For instance, to investigate new regions of the solution space, one variable in a chromosome may be changed. Until a termination condition is satisfied, such arriving at a suitable solution or finishing a maximum number of generations, the Genetic Algorithm repeats these steps. GAs are excellent in identifying near-optimal solutions in a variety of areas, such as engineering design, machine learning, land-use optimization, and scheduling.

GAs have limits in spite of their advantages. They can be computationally costly since it takes a lot of resources to evaluate big populations across several generations. Additionally, populations may prematurely converge to less-than-ideal solutions if there is insufficient diversity. For example, a GA in an exam timetable generator makes sure that schedules adhere to limitations such as room capacity, student enrollments, and teacher availability. It ensures that there are no overlapping tests for teachers or students by evaluating solutions using a fitness function to identify and punish conflicts. GAs are an effective tool for resolving challenging scheduling issues because of this procedure, which produces an optimum and conflict-free schedule.

3.3. Constraint Satisfaction Model :

Mathematical frameworks for resolving issues with variables, domains, and constraints are known as constraint satisfaction models, or CSMs. Limitations are the guidelines that dictate acceptable combinations of variable assignments (e.g., no two events in the same room at the same time), domains specify potential values for these variables (e.g., available rooms), and variables represent decision points (e.g., time slots). CSMs provide systematic and formal methods for addressing intricate multi-variable issues like resource allocation and scheduling. They use techniques like constraint propagation to reduce variable domains and backtracking for systematic exploration to make problems simpler.

By investigating nearby choices, CSMs also use heuristic strategies like local search and hill climbing to incrementally enhance solutions. CSMs are used in scheduling (e.g., optimizing hospital surgeries to prevent conflicts), AI planning (e.g., sequencing activities in robots), and resource allocation (e.g., assigning workers or equipment effectively). A formal depiction of problems and the capacity to manage complex, interrelated restrictions are among its strong points. Real-world solutions, such optimal test scheduling, are made possible by libraries like Google's ortools.sat.python.cp_model, which offer strong tools for defining and resolving CSM-based challenges.

3.4. Simulated Annealing (SA):

The annealing process in metallurgy serves as the inspiration for the probabilistic optimization approach known as Simulated Annealing (SA). In order to reach a stable, low-energy state, a material must first be heated to permit free movement of its atoms and then gradually cooled. In order to obtain an optimal or nearly optimal solution, SA imitates this process by first exploring a solution space and then gradually reducing it. Temperature regulates the extent of investigation, and an objective function is used to assess the quality of the answer. While lower temperatures concentrate on refining solutions, higher temperatures provide unlimited investigation.

SA generates adjacent solutions, computes energy changes, and accepts solutions based on transition probability in an iterative process. This aids in locating global optima and escaping local minima. Exponential or linear decay cooling programs lower the temperature gradually.

SA can be used to optimize VLSI circuit design, schedule work or resources, and solve the Traveling Salesman Problem (TSP). Although SA is easy to use, adaptable, and good at avoiding local minima, it is slow to converge, sensitive to parameters, and might not provide global optimality. SA is still a flexible technique for handling challenging optimization issues in spite of its drawbacks.

3.5. MongoDB vs. MySQL for Examination Timetable Generator

MySQL was selected in the beginning for the Examination Timetable Generator project because of its relational database structure. MySQL works effectively with projects that have relationships that are steady and organized. But as the project progressed, MySQL's shortcomings became more apparent due to the dynamic nature of data including student registrations, faculty schedules, and optional courses. It was challenging to adjust to the project's requirements because to its stringent schema constraints and need on intricate joins for nested data.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

A better option turned out to be MongoDB, a NoSQL database. Dynamic, semi-structured data was easier to store and query thanks to its schema-less architecture, which enabled flexible data modeling. By allowing related data to be stored in a single document, MongoDB's document-based design decreased the requirement for joins and increased query performance. This was especially helpful for managing hierarchical and nested data, such faculty availability and scheduling restrictions.

Another significant benefit of MongoDB over MySQL was scalability. MongoDB provides horizontal scalability, dispersing data across several nodes, whereas MySQL necessitates vertical scaling, which involves updating hardware to handle additional data. This feature was essential for handling sizable datasets produced by computational techniques that need a lot of data processing capacity, such as the Genetic Algorithm, Simulated Annealing, and Constraint Satisfaction Problem (CSP).

The project's switch to MongoDB resulted in increased flexibility in data management, quicker query execution, and better performance. Because of these advantages, MongoDB was able to easily integrate with optimization algorithms, making it the best option for creating effective, student-friendly exam schedules that could be modified to meet evolving needs.

IV. PSEUDO CODE

```
Step 1: Generate all possible time slots for each course based on the exam duration, room availability, and faculty availability.
Step 2: For each time slot of each course, calculate the feasibility score (Fscore) using constraints like:
        - Faculty availability
        - Room capacity
        - No overlapping exams for students
        - Daily time limits for exams
Step 3: Check the following condition for each course until all exams are scheduled:
        if (Fscore < Threshold):
            Mark the time slot as unavailable
        else:
            Select all feasible time slots for the course
        end
Step 4: Assign the course to the most feasible time slot based on the highest Fscore.
Step 5: Update availability data:
        - Mark the selected time slot as occupied for the assigned room and faculty
        - Remove the assigned course from the list of unscheduled exams
Step 6: Recalculate feasibility scores (Fscore) for remaining unscheduled courses considering updated availability.
Step 7: Repeat Step 3 to Step 6 until all courses are scheduled or no valid slots are available.
Step 8: If conflicts exist (e.g., overlapping exams, faculty overload):
        Resolve conflicts by reassigning exams to alternative slots with acceptable feasibility scores.
        If no valid alternative exists, flag the conflict for manual resolution.
Step 9: End.
```

V. SIMULATION RESULTS

The system effectively automizes scheduling by resource optimization, constraint satisfaction, and conflict resolution. Testing on datasets of varying complexity-small, medium, and large-showed its ability to produce conflict-free, fair schedules while strictly following constraints such as avoiding overlaps and faculty availability. Soft constraints such as reducing gaps between exams were generally maintained with minimal compromises on performance. The execution times ranged from less than a minute for small datasets to about ten minutes for large, complex ones.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

By effectively automating the scheduling process, the examination timetable generation system tackles important issues including resource optimization, constraint satisfaction, and conflict resolution. The findings were verified using a variety of test datasets with a range of circumstances, such as:

1. Limited Dataset: Few faculty, students, and courses.
2. Medium Dataset: There are a fair amount of elective overlaps and courses.
3. Big Dataset: Numerous classes, electives, and instructors with intricate restrictions.

The system proved that it could manage these situations well, generating fair and accurate schedules in a manageable amount of calculation time.

Important Performance Indicators and Efficiency in Resolving Conflicts: Schedules were completely free of conflicts in every dataset. Successfully prevented instructor and student exam times from overlapping.

Constraint Contentment: Without compromising, strict requirements (such as avoiding overlapping tests and faculty availability) were fulfilled. With sporadic compromises to preserve computing performance, soft limitations (such as reducing the time between tests for students) shown good adherence.

Time of Execution: In less than a minute, small datasets were processed. About two to three minutes were needed for medium datasets. Depending on complexity and further restrictions, large datasets could take up to ten minutes.

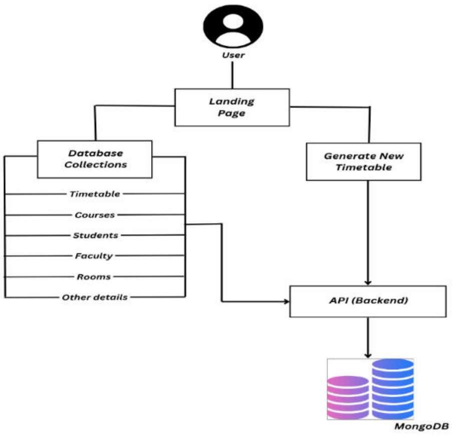


Figure 1. Front-End Development data flow

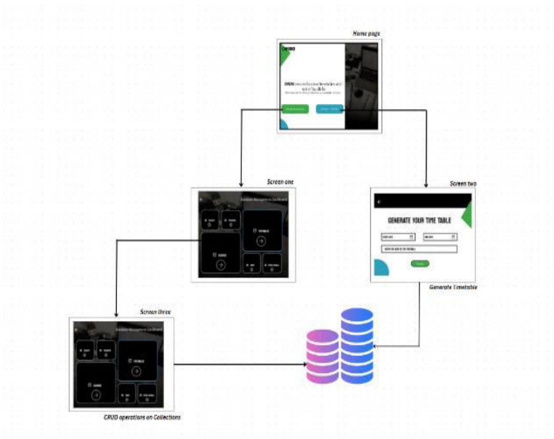


Fig 2. System Architecture

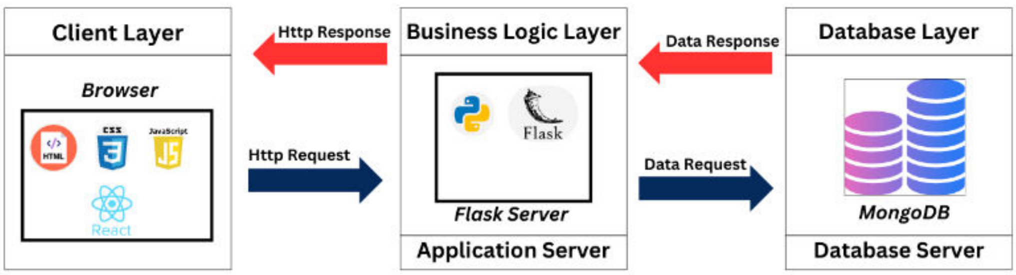


Figure 3. Back-End Development data flow

VI. CONCLUSION AND FUTURE WORK

This generator also has the benefits of scalability, fairness, and flexibility with user-friendliness, along with automation that saves time and decreases errors. But there are certain drawbacks such as computational delay on large datasets, problems with dynamic updates requiring a re-optimization process, and sometimes relaxing soft constraints. The system would have greater accuracy, speed, and flexibility when compared to the manual methods. Future scope can be dynamic constraint handling, AI integration for trend prediction and optimization, and cloud computing for improved scalability and efficiency.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

In conclusion, the improvement and execution of the Programmed Examination Timetable Era Framework has illustrated its capacity to create conflict-free timetables that regard both difficult and delicate imperatives. The discoveries appear that computerized framework can move forward operational proficiency and reasonableness in examination planning, profiting both understudies and staff. The think about highlights the significance of leveraging computational methods in tending to complex authoritative assignments, eventually contributing to a more conductive learning environment. Future advancements to the framework, such as consolidating machine learning and taking care of last-minute changes, will proceed to upgrade its commonsense pertinence and adequacy.

REFERENCES

1. “Algorithms for Constraint Satisfaction Problems CSPs” by Zhe Liu, A thesis presented to the University of Waterloo, in fullment of the thesis requirement for the degree of Master of Mathematics in Computer Science, Waterloo, Ontario, Canada.
2. A. Aminu et al., “Design and Implementation of An Automatic Examination Timetable Generation and Invigilation Scheduling System Using Genetic Algorithm,” 2019 2nd International Conference on Applied Engineering (ICAE), Batam, Indonesia, 2019, pp.
3. C. Gogos, G. Goulas, P. Alefragis and E. Housos, "Pursuit of better results for the examination timetabling problem using grid resources," 2009 IEEE Symposium on Computational Intelligence in Scheduling, Nashville, TN, USA, 2009, pp. 48-53, doi: 10.1109/SCIS.2009.4927014.
4. E. Alexa, D. Zmaranda and E. V. Moisi, "Modeling and Solving the University Timetabling Problem using the Constraints Satisfaction Model: A Case Study," 2021 13th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Pitesti, Romania, 2021, pp. 1-6, doi: 10.1109/ECAI52376.2021.9515060.
5. “The Application of Genetic Algorithm in Land Use Optimization Research” by Xiaoe Ding , Minrui Zheng , and Xinqi Zheng
6. “A theoretical evaluation of selected backtracking algorithms” Grzegorz Kondrak , Peter van Beek Department of Computing Science, University of Alberta, Edmonton, Alta., Canada T6G 2HI Received February 1996; revised June 1996
7. “Heuristic search revisited” Abdel-Elah Al-Ayyoub, Fawaz A Masoud , Department of Computer Science and Information Systems, Jordan University of Science and Technology, Irbid 22110, Jordan Department of Computer Science, Sultan Qaboos University, Box: 36, Al-Khod, Muscat 123, Oman
8. “Particle Swarm Optimization: A Survey of Historical and Recent Developments with Hybridization Perspectives” Saptarshi Sengupta, Sanchita Basak and Richard Alan Peters II Department of Electrical Engineering and Computer Science, Vanderbilt University, 2201 West End Ave, Nashville, TN 37235
9. OpenAI. "API Documentation." OpenAI Platform. Accessed December 24, 2024. https://platform.openai.com/docs/
10. Khan, A. H., and T. Imtiaz. "A Novel Genetic Algorithm Based Timetable Generator for Optimized University Timetable Solution." *2024 International Conference on Engineering & Computing Technologies (ICECT)*, 2024, pp. 1–6, doi:10.1109/ICECT61618.2024.10581296.
11. D. Kusumawardani, A. Muklason, and V. A. Supoyo, "Examination Timetabling Automation and Optimization using Greedy-Simulated Annealing Hyper-heuristics Algorithm," *2019 12th International Conference on Information & Communication Technology and System (ICTS)*, Surabaya, Indonesia, 2019, pp. 1-6, doi: 10.1109/ICTS.2019.8850932.
12. S. Innet, "A novel approach of genetic algorithm for solving examination timetabling problems: A case study of Thai Universities," *2013 13th International Symposium on Communications and Information Technologies (ISCIT)*, Surat Thani, Thailand, 2013, pp. 233-237, doi: 10.1109/ISCIT.2013.6645855.
13. L. Zhang and S. Lau, "Constructing university timetable using constraint satisfaction programming approach," *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*, Vienna, Austria, 2005, pp. 55-60, doi: 10.1109/CIMCA.2005.1631445.
14. K. Banczyk, T. Boinski, and H. Krawczyk, "Parallelisation of Genetic Algorithms for Solving University Timetabling Problems," *International Symposium on Parallel Computing in Electrical Engineering (PARELEC'06)*, Bialystok, Poland, 2006, pp. 325-330, doi: 10.1109/PARELEC.2006.64.