iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 **Issue:** VI **Month of publication:** June 2023

DOI: https://doi.org/10.22214/ijraset.2023.53821

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue VI Jun 2023- Available at www.ijraset.com

Automatic Timetable Generator

Prof. Jyothi Patil[1], Shambhavi V[2], Sneha N T[3], Sweta Jadhav[4], Tahura Sadaf[5]
Department of Electronics & Communication Engineering, P D A College of Engineering, Kalaburagi, Karnataka, India

Abstract: ***The goal of this project is to create a time table generator for colleges. The creation of schedules is a very common issue that affects all educational institutions.***
The conflict between staff members' preferences is precisely where the issue arises. Every semester, colleges are required to create time tables, which used to be an extremely time-consuming task. Once the timetables are set for a given semester, the student is allowed to access them. Once the timetables are established for a particular semester, employees are also permitted to check the class allotment schedule.
The Time Table Assignment for Any Department project's goal was to create an application that would allow staff and student allotment subject to classes. Following information was added by the administrator for Add the student, the staff, the subject, enter the timetable, and update the timetable. The majority of colleges offer a variety of programmes, each of which has a number of disciplines.
There are now a limited number of faculties, each of which teaches many disciplines. Therefore, the timetable now has to include the instructors at the appropriate times. the timetable schedule, which makes the most use of all faculty subject demands, slots so that their timings do not cross. For this, a genetic algorithm is employed. We suggest using a timetable object in our method for creating timetables.
This object consists of classroom objects, their respective schedules, and a fitness rating for the schedule. Additionally, in order to further describe the imperatives, we used a composite configuration design that is easily expandable to include or uproot as many duties.
Every obligation class now checks the condition found in our investigation between two timetable objects. In the unlikely event that the requirement is met, the score is raised by one if a crash is available.
Keywords: Genetic Algorithm, Artificial Intelligence, Initial Population

I. INTRODUCTION

Every semester, colleges are required to create time tables, which used to be an extremely time-consuming task. The "Time Table 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 time table 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 time table details in this website is the time table generator system.

This undertaking Python and SQLite have been used in the development of the online automatic timetable generator. A programme written in Python called Automatic 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. LITERATURE SURVEY

1) 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. Tabus 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.
2) 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.
3) W. F. Mahmudy and R. E. Febrita (2017), Use fuzzy logic to create and carry out timetable scheduling that incorporates multiple genetic operators. The constraints are resolved using fuzzy logic, a multivalent logic. This is derived from fuzzy set theory and is used to replace exact reasoning with approximative reasoning. The proportion of truth of a statement may vary between 0 and 1, depending on the membership values of formal fuzzy logic variables, which may not always be 0 or 1. Fuzzy logic is not limited to two value reasoning, in contrast to classical logic. Results show that this method can be used to maximise difficult scheduling objectives and produce outcomes that are realistic. By employing linguistic factors, a more stable state is reached in a shorter amount of time. Membership function evaluation is difficult, hard to create a fuzzy logic model, more calibrate tuning and simulation is required before using for any application. These are some of the major drawbacks of this method.
4) T. Elsaka (2017), utilise constraint satisfaction modelling for the schedule of automated generating. Instead of concentrating on the target function, constraint programming focuses on the restrictions and variables domain. It also relies on viability rather than improvisation. By regulating the constraints along a system of constraints propagation, the author depicts the time table generation technique that minimises variable domains combined with backtracking search. A key benefit of this programming method is constraint programming, which is essentially a clear declaration of the restrictions that are used in the programme. This is unacceptable in terms of scheduling concerns because it makes changing the programme simple. Data and constraints are the two key parts of this process. The authors Abhishek M. B. and N. S. V. Shet have presented data gathering and computing methods that play a crucial role in smart data management and monitoring water distribution from a cyber-physical perspective. The approach is hampered by difficulties identifying soft limitations, possibly complex issues with deepening the initial suitable answer, and time constraints.
5) S. Ab Saad, F. A. Adnan, W. Z. A. Wan Muhamad and Z. R. Yahya (2018), By conducting an investigation, the use of genetic algorithms in tough scheduling problems is analysed. Genetic algorithms are a subset of gradual developmental algorithms that are influenced by the process of natural selection. The selection of suitable individuals from a population is the first step in the process of natural selection. This process continues to iterate until production is established with the proper people. The primary timetabling challenge eliminates some of the variables and consolidates them into constraints with a large number of binary variables that shrink to a manageable size. By combining several binary data into a single gene value, the genetic algorithm approach significantly reduces the size of the individual. Involving lot of parameters and Time consuming as it gives results with iterative approach are some of the flaws in this approach.
6) D. Apostolou and E. Psarra (2019), provided a method for addressing the challenge of creating a school's timetable utilising a hybrid optimization strategy. PSO is a technique for artificial intelligence that is used to get a substantial outcome in a situation with both increased and decreased problems. PSO eliminates the issue by using local search. By combining the traditional PSO upgradation method with prototype methodology, this challenge is successfully addressed. The optimization process takes a reasonable amount of time—between a few seconds to a few minutes. The advantages of this strategy are that particles improve themselves with their own internal velocities and possess memory, which is superior to algorithms. One disadvantage of PSO is that it lacks genetic operators like crossover and mutation.

III.PROBLEM STATEMENT

The existing system produces or generates a series of timetables, however it frequently struggles to provide a timetable that is conflict-free and complete. The bottlenecks in this situation are the laborious chores of data introduction and adjustment of frequently partial solutions (Luisa et.al, 2006). The majority of educational institutions still manually create their timetables, which according to statistics takes far longer to accomplish and is less efficient. There are still some conflicts in the manually generated timetable, even at its best point, and it is the lecturer who is taking the conflicting course that arranges the course's logistics in order to resolve the conflict.

Despite the fact that human scheduling takes a lot of time and is unreliable, tiny colleges have found ways to construct their schedules. It will be vital to use computer approaches to streamline timetabling as academic complexity rises. It should be noted that the number of constraints grows, leading to an exponential increase in the computational time, making it an NP-complete operation as the student population with diverse interests and requirements grows and the teaching programmes become more complex with the expansion of universities.

A. Problem Objective

The major objective of a college timetable generator is to create an efficient and effective schedule for classes and other activities in a college or university. This objective is achieved by taking into account various constraints and criteria, such as

1) Resource availability,
2) Student enrolment,
3) Faculty preferences and availability,
4) Optimization criteria.

The college timetable generator may produce a schedule that satisfies the requirements of the students, teachers, and administrators while maximising efficiency and effectiveness by meeting these goals. This may lead to better academic achievement, greater student satisfaction, and less work for administrators and professors.

B. Existing System

Each operation must be completed manually under the current system, and processing is a time-consuming effort as well. In the prior method, institutions had to manually manage their timetable information on paper and ink, which took time and money. The organisation can't meet its demands in a timely manner, and the outcomes might not be reliable. Numerous issues and shortcomings with the system are caused by manual upkeep.

C. Proposed System and Solution

The purpose of this work is to illustrate how effectively evolutionary algorithms can be used to find the best solutions for scheduling timetables in general. Despite the abundance of commercial scheduling software, its lack of generality makes it difficult for it to satisfy the needs of varied institutions. The main challenge to overcome is the demand of particular coding as per the distinct colleges. An institute must deal with a number of restrictions when constructing a schedule. These constraints can be categorised as either "hard" or "soft" based on whether they are necessary or desirable.

D. System Specifications

1) Hardware Requirements

- Laptop or PC
- Windows 7 or higher
- I3 processor system or higher
- 8 GB RAM or higher
- 100 GB ROM or higher

2) Software Requirements

- Laptop or PC
- Python 3.6 or Higher
- VS code or PyCharm IDE
- Mysqlte3.

IV. DESIGN METHODOLOGY

A. Block Diagram Of The System

The software can simply be compared to a schedule computing tool that accepts simple data sets as input and produces organised results. Using the straightforward input-process model We fill out the form with all the information, and a genetic algorithm applies it to the data on the backend to create class schedules.

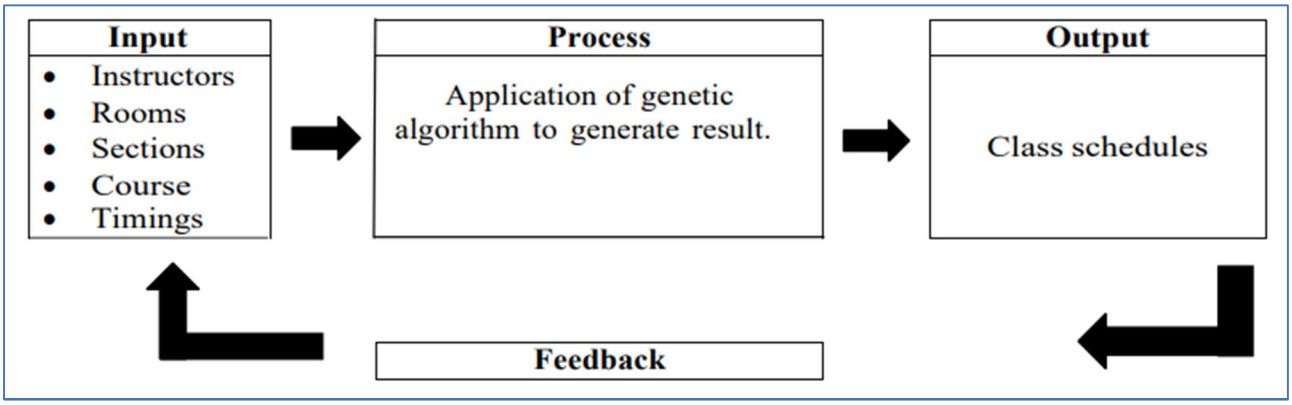


Fig 1: Simplified IPO-Model of the Program

B. Genetic Algorithm

Natural selection, a process in biological evolution, is a process that the genetic algorithm mimics. The best way to visualise the repeating process is through a flowchart. Instead of using the term "person," the term "chromosome" is used in genetic algorithms. The fundamentals of a genetic algorithm are depicted in the diagram below. However, in actual use, the system adds an additional phase known as environment modification following evaluation. A strong programming tool for issue solving is the genetic algorithm (GA). It belongs to the class of evolutionary algorithms, which is a subset of artificial intelligence's evolutionary computation. Professor John Holland of the University of Michigan created it in 1960. In the 1970s, his book Adaptation in Natural and Artificial Systems helped launch the field of genetic algorithm (GA) study. The Darwinian theory of natural evolution, which holds that species in the world multiply in geometric proportions resulting to a battle for life mostly owing to a lack of food and space, served as the inspiration for this technique. The strongest will prevails in this conflict. The variants that are most suited to survival are those whose accumulation led to the evolution of species. Organisms with harmful mutations have very little chance of surviving. Natural selection is the mechanism that leads to evolution.

A random beginning point will be chosen for the population generation, and a mixture of greedy and random approaches will be used to fill the table with any suitable entities. All rigid restrictions must be adhered to throughout population generation. While soft constraints are completely disregarded, medium requirements will go through several tries to be followed

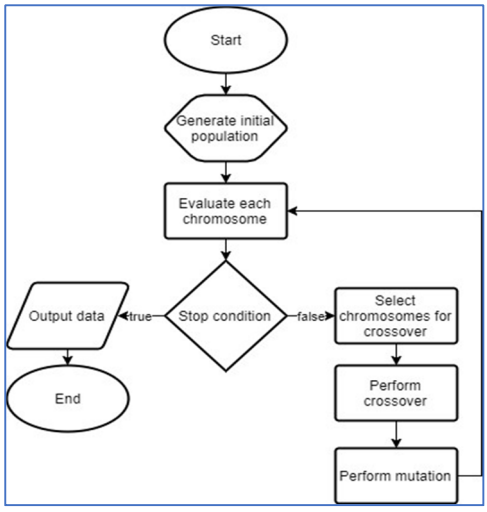


Figure 2: Genetic Algorithm

The whole method of scheduling based on genetic algorithm is explained in detail in this section. A scheduling procedure is divided into several important modules are as follows,

1) Data Encoding and Decoding

The initial step before beginning a Genetic Algorithm is data encoding. To obtain a straightforward value, such as a string, it converts a solution into a chromosome. It is used to increase the algorithm's speed. Making the data into a binary string is a simple technique to accomplish this. A gene is a component of a chromosome and can also be transformed into a binary string. The algorithm can be treated more easily by converting the data to this kind. Side-by-side gene strings make up the chromosomal string.

2) Initial Population

It is the initial phase of GA. It involves producing a large number of randomly chosen people under strict limits. The user's needs determine the population choice. Due to evolution, a tiny population will eventually become even smaller and exterminate the entire population. On the other side, a huge population will produce better outcomes but will be slower and need more resources. A set can be used to represent the population.

3) Evaluation of Population

A solution's fitness is an assessment of its quality made utilising soft restrictions. The answer is appropriate for this range. The core of the genetic algorithm is population evaluation. In this stage, a fitness function is used to determine which solution is superior to others. The fitness can be expressed using a range from 0 to 1, with 1 being the population's best estimate, and using the other individuals to range them. There will always be a solution where the fitness is 1 in this situation, and a solution where the fitness is 0.

4) Crossover Evolution

A technique for creating a new population based on an older population is crossover evolution. The two-chromosome simple crossover evolution allows for the creation of X additional chromosomes. The two chromosomes are divided into pieces, and new chromosomes are made from various pieces. Mutation: The algorithm is made to move by means of mutation. By randomly altering a gene's values, a novel, unexpected solution is produced. These solutions present the fitness function from a fresh perspective. Only the chromosome is altered by the mutation; other solutions are unaffected.

5) New Population

The crossover and the mutation permit to create a new population of original solutions.

V. ADVANTAGES AND DISADVANTAGES

A. Advantages

1) Unlike the manual timetabling system, the system offers flexibility.
2) It utilizes minimal processing/computing power.
3) It greatly reduces the time needed to generate maximum error free timetables.
4) It provides an easy means for data entry and revision through an intuitive interface.
5) It increases productivity.
6) Timetables generated are between to 60% – 80% best solution.
7) It almost eliminates paperwork.
8) It simplifies the timetabling process.
9) Gives accurate information.
10) Simplifies the manual work.
11) It minimizes the documentation related work.
12) Provides up to date information
13) Friendly Environment by providing warning messages.

B. Disadvantages

1) Adding raw content is mandatory.
2) It requires a large amount of memory.
3) It requires the internet.

VI. APPLICATIONS

Large applicability in an institution where different class schedules are necessary. Within minutes, automatically establish and maintain student academic calendars. You can quickly construct a different timetable for each class and subject using an automated timetable management system.

There exist a lot of diverse timetable problems such as:

1) University Timetable
2) Exam Timetable
3) School Timetable
4) Sports Timetable
5) Worker Timetable

All of the timetables can be created easily and productively using genetic algorithm.

VII. RESULT AND DISCUSSION

This system generates separate timetables for each class, faculty member, and lab automatically. The project is designed to prevent slot conflicts and provides tools for customising the schedule as needed. The project employs a genetic algorithm to meet the scheduling-related limitations. The following restrictions are met by the programme: -

Hard Constraints	Soft Constraints
Unique class timing	classes are allotted according to section requirements
Course.students <= room. seating capacity and Teachers are allocated to their course accordingly	All courses are according to their department
Two classes don't have same room and Class timing for each teacher is unique	Even distribution of course in a section per week

Models were employed to test the artificial intelligence's capabilities. Despite the constraints put forth by the setting of the algorithm, it is reasonable to claim that the system has operated within its capabilities. The final output screens, which include several options for adding instructor details, adding course details with a department, and adding room and time details, are as shown in the figures below. On the home page, we can create the schedule.

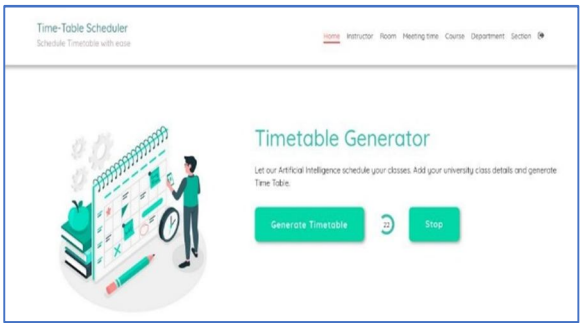


Fig 3: Home Page

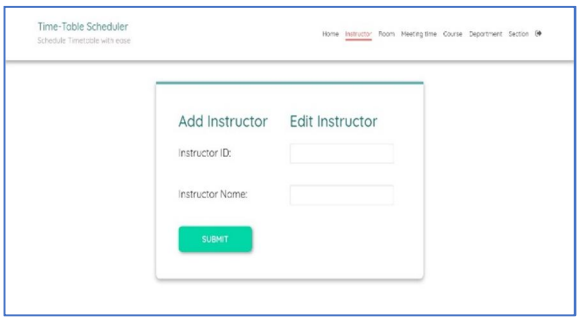


Fig 4: Instructor details input

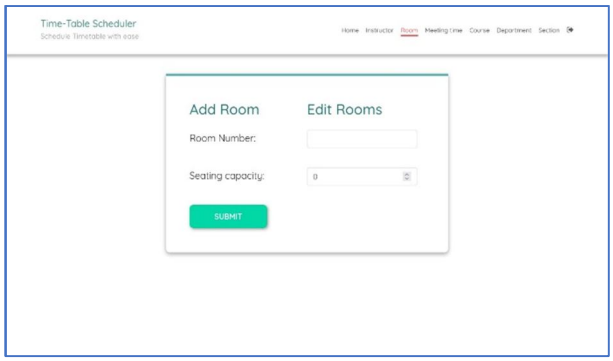


Figure 6 – Meeting details input

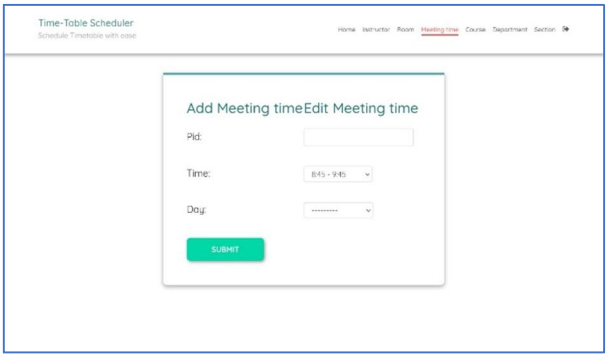


Figure 5 –Room details input

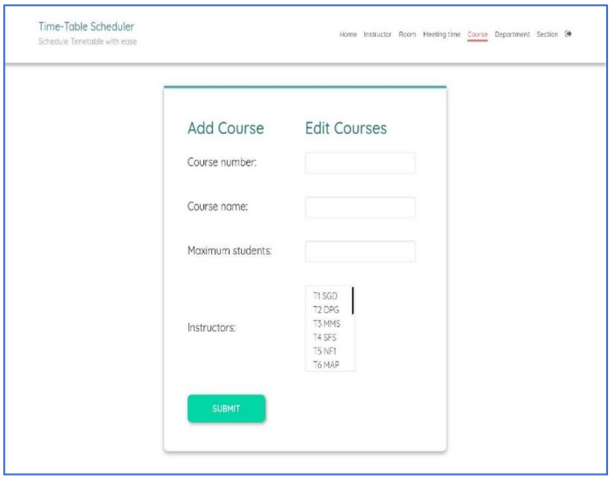


Figure 8 – Department details input

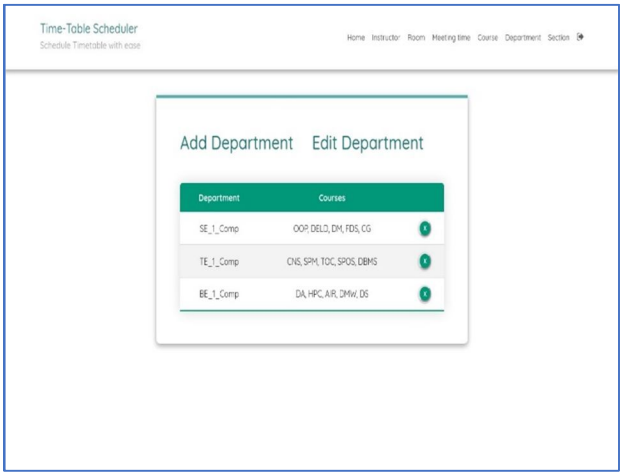


Figure 7 – Courses details input

VIII. 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.

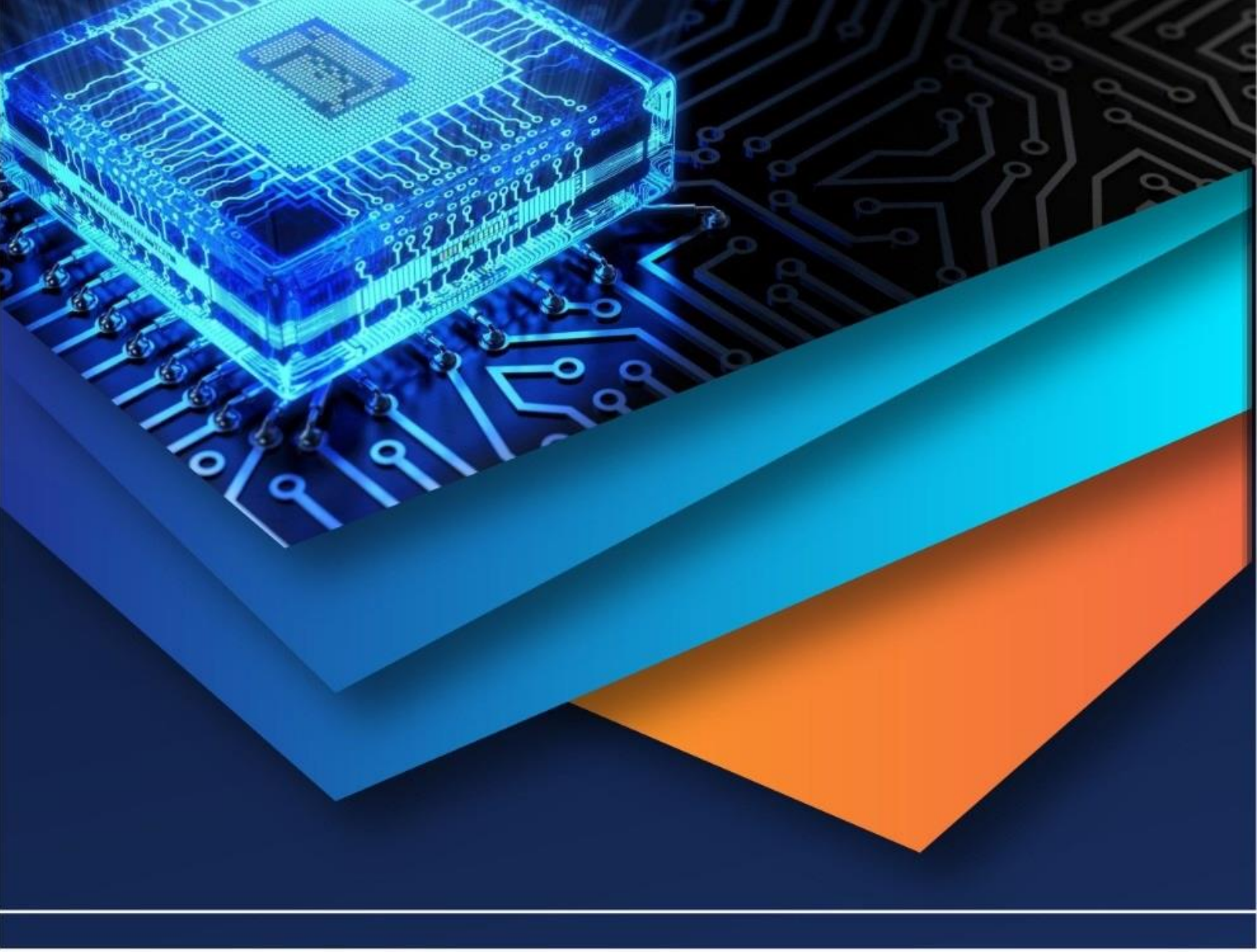
We have made significant progress, and we can say with confidence that it is highly fulfilling. We gained a solid understanding of the fundamentals of web construction as well as how evolutionary and genetic algorithms can be combined to get the best goal-finding performance in terms of time and space 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.

IX. 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 utilised 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 time table 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] K. Nguyen, D. Nguyen, K. Trieu, and N. Tran, “Automating a real-world university timetabling problem with Tabu search algorithm,” in 2010 IEEE RIVF International Conference on Computing & Communication Technologies, Research, Innovation, and Vision for the Future (RIVF), 2010
[2] A. Azlan and N. M. Hussin, “Implementing graph coloring heuristic in construction phase of curriculum-based course timetabling problem,” in 2013 IEEE Symposium on Computers & Informatics (ISCI), 2013.
[3] R. E. Febrita and W. F. Mahmudy, “Modified genetic algorithm for high school time-table scheduling with fuzzy time window,” in 2017 International Conference on Sustainable Information Engineering and Technology (SIET), 2017.
[4] T. Elsaka, “Autonomous generation of conflict-free examination timetable using constraint satisfaction modelling,” in 2017 International Artificial Intelligence and Data Processing Symposium (IDAP), 2017.
[5] F. D. Wihartiko, H. Wijayanti, and F. Virgantari, “Performance comparison of genetic algorithms and particle swarm optimization for model integer programming bus timetabling problem,” IOP Conf. Ser. Mater. Sci. Eng., vol. 332, p. 012020, 2018.
[6] A. Aminu et al., “Design and implementation of an automatic examination timetable generation and invigilation scheduling system using genetic algorithm,” in 2nd International Conference on Applied Engineering (ICAE), 2019.
[7] D. Wang, D. Tan, and L. Liu, “Particle swarm optimization algorithm: an overview,” Soft Comput., vol. 22, no. 2, pp. 387–408, 2018.
[8] K. Y. Junn, J. H. Obit, and R. Alfred, “The study of genetic algorithm approach to solving university course timetabling problem,” in Lecture Notes in Electrical Engineering, Singapore: Springer Singapore, 2018, pp. 454–463.
[9] E. Psarra and D. Apostolou, “Timetable scheduling using a hybrid particle swarm optimization with local search approach,” in 2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA), 2019.
[10] T. W. Ekanayake, P. Subasinghe, S. Ragel, A. Gamage, and S. Attanayaka, “Intelligent timetable scheduler: A comparison of genetic, graph coloring, heuristic and iterated local search algorithms,” in 2019 International Conference on Advancements in Computing (ICAC), 2019.
[11] A. Schaerf, Local search techniques for large high-school timetabling problems, IEEE Transactions on Systems Man. And Cybernetics, Part A, 29 (4), 368-377 (1999).
[12] H.P. Hariyadi, T. Widiyaningtyas, et. al., Implementation of Genetic Algorithm to Academic Scheduling System, IEEE, Region 10 Conference, (2016).
[13] S. Daskalaki, T. Birbas, E. Housos, An integer programming formulation for a case study in university timetabling, Eur. J. Oper. Res., 153, 117-135 (2004)
[14] A.F. AbouElhamayed, A.S. Mahmodd, et. al., An Enhanced Genetic Algorithm-Based Timetabling System with Incremental Changes, IEEE, (2016).
[15] M. Dimopoulou, P. Miliotis, An automated university course timetabling system developed in a distributed environment: A case study, Eur. J. Oper. Res., 153 136– 147 (2004).
[16] E. Burke, K. Jackson, J. H. Kingston, R. Weare, Automated university timetabling: The state of the art, The Computer Journal, 40 (9): 565-571, (1997).
[17] Rosen, Kenneth H. (2012). Discrete Mathematics and Its Applications (7th ed.). New York: McGraw-Hill. p. 119. ISBN 978-0-07-338309-5
[18] D. Shiffman, The Nature of Code, Chapter 9, 2012, Creative Commons, 444 Castro Street, Suite 900, Mountain View, California 94041, USA. ISBN-13: 978-0985930806.
[19] E. R. Pianka, Evolutionary Ecology, 6th ed. San Francisco, CA: Addison-Wesley- Longman, 2000; Deepti Gupta, Shabina Ghafir, International Journal of Emerging Technology and Advanced Engineering, 2(5), (2012).
[20] Shraddha Thakare, Tejal Nikam, Prof, Mamta Patil, Automated Timetable Generation using Genetic Algorithm, International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181,Vol. 9 Issue 07, July-2020



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY