International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 04, ICTEM 2.0 (2026)

EDUCHRONO: Smart Timetable Generator

Ajeet Singh[1], Vanshika[2], Yashi Gupta[3], Madhav Tomar[4]

[1,2,3,4]Department of Computer Science and Engineering Moradabad Institute of Technology University Moradabad, India

[1]ajeetsingh252@gmail.com
[2]vc315315@gmail.com
[3]yashigupta0730@gmail.com
[4]madhavtomar3007@gmail.com

Abstract:

The Smart Timetable Generator is a system that automatically creates academic timetables by assigning subjects, teachers, classrooms and time slots in a proper way. Making a timetable by hand is very difficult and takes lot of time. It also creates many problems like the same teacher getting two classes at the same time, room clashes and uneven workload for faculty members.

To solve these issues, this system uses a smart rule-based method with the help of Google-OR-tool CP-SAT solver. The system will take all the data which is required through CSV files such as subject list, teacher details, lab details or requirements, section details and room or labs availability. After taking the all the data, the system follows fixed rules to avoid teacher and class clashes, assigning lectures or labs in continuous manner such that there is no gap between two lectures or labs.

I. Introduction

The Smart Timetable EduChrono is a smart system that automatically generates timetable for schools and colleges. Generally making a timetable manually takes lot of time and sometimes create problems like clashes of faculty or lectures and uneven workload and also difficult to managing lab classes. To avoid such type of issues, the EduChrono takes only teacher load and room or labs list through CSV files it will generate error free and clash free timetable. It uses an Google OR-Tools CP-SAT to make a clean and clash-free timetable**.**

The EduChrono system also provides a simple interface where we can easily upload files and see the generated Master timetable, we can also see it as a class-wise and faculty-wise. It saves time, reduces manual workload for creating timetable and helps in better use of teachers and classrooms or labs. Compared to old manual method, the EduChrono is faster and give more accurate timetable. That's why this EduChrono is very simple and fast and useful solution for generating timetable.

1.1 Background

Making a college timetable is difficult to make manually which takes lot of time to generate it. There are many lectures, subject, labs and rooms so. Mistakes can happen easily like room clash, faculty clash, uneven gaps and workload. Also, if single change was needed then the whole timetable needs to be re-generated.

The EduChrono Smart Timetable Generator solve all the problems and give output as a clean and accurate timetable even it generates the timetable within the second. It works on fixed rules and constraints.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 04, ICTEM 2.0 (2026)

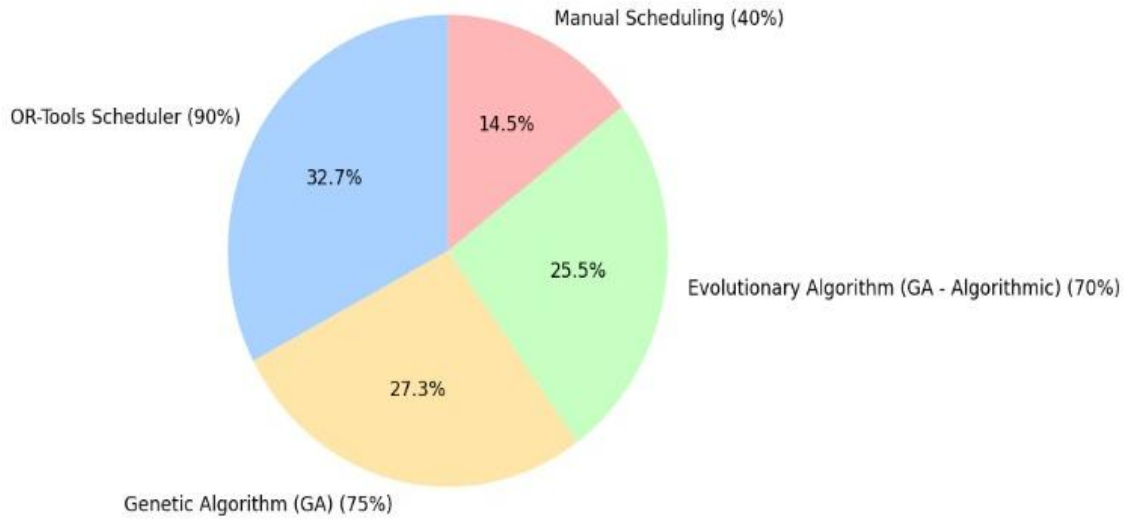

Figure 1: Efficiency comparison of Timetable Scheduling

1.2 Problem Statement

Generally, a manually making timetable is slow and may contain errors. It is difficult to manage the workload of faculty, rooms or subject hours. Many it happens that faculty given two lectures at the same time which create clash or uneven distribution.
Other methods like AI and genetic algorithm were tested, but they do not work with all the constraints or rules. Also, these scheduling with methods like PSO required metaheuristic algorithm that do not guarantee an optimal or fully conflict-free solution [2]. A genetic algorithm-based automated timetable generator not handles all the constraints and may require high computational resources, complex parameters[4]. So, we need Smart Timetable Generator which uses a strict-rule base method to generate correct and clash-free timetables. We uses Google OR-Tools for the timetable generation which makes an clash-free and which follows all the constraints[5].

1.3 Contributors

This project gives a simple automatic generated system which uses python. It takes teacher load which includes faculty name and code, faculty workload, sections, department and subject name and code and room list which includes capacity and assigned section or semester in the form of CSV files.
It follows all the constraints like two hours continuous lab and make sure that COE and PDP classes are not on the same day. It provides a simple interface for login and view timetable based on their role.

II. Related Work

For this project, we studied many research papers on automatically generating timetables making to understand how timetables were created using different methods. Earlier people used different methods like Genetic Algorithm, but now we have more advanced tools like Google OR-Tools and CP-SAT solver which are used for better and faster results. In PyJobshop research paper shows that constraints programming is effective and good for scheduling problems our project work on the same idea but we don't use PyJobshop we focus on scheduling for college using OR-Tools CP-SAT solver [3]. From this study, we learned that rule-based system gives more accurate and clash-free timetables.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 04, ICTEM 2.0 (2026)

Table 1: Comparative Table of Related Works

METHOD	DATA IT USED	HOW IT WORKS	OPTIMIZATION METHOD	LIMITATIONS	YEARS
Simulated annealing (SA)	It takes courses, teacher, rooms datasets	Constraint satisfaction & rule enforcement	Stochastic local search is used	It required tuning parameters, also it require many iterations and still not guarantee best solution	1983
PSO (particle swarm Optimization)	Course & room allocation datasets	Birds or swarm movement to search for food (better solution)	Particle Swarm Optimization	Reduce efficiency when constraints are increased also it does not guarantee conflict-free scheduling	2012
Genetic Algorithm / Evolutionary algorithm Simulate (SA)	Various academic scheduling datasets	Select best schedules, change	Evolutionary algorithm based on selection, crossover, mutation	When data is large it becomes slow Sometimes it can violate constraints	2021
EduChrono: Google OR-Tools (proposed)	Just input csv files (Teachers load and Room data)	Automated schedule generation	Google OR-Tools (CP-SAT)	Needs proper constraints	2025

Literature Review

Scheduling a timetable is a complex problem and it becomes more difficult when multiple constraints are added. Generation of manual timetable is time consuming and even may contain errors. Previous genetic and AI scheduling methods also faced many problems such as not generating perfect timetable, struggling with hard constraints and may be slow in generating timetable(scheduling) [1][2]. Moreover, existing research methods do not guarantee that they always generate a conflict-free timetable. But in EduChrono uses Google OR-Tools and CP-SAT solver to generate accurate and clast-free timetable which strictly follows all the rules and constraints such as no gaps, continuous labs scheduling, no faculty have two lectures at a same time. It offers a user-friendly interface with is suitable for- HODs, faculty and students.

III. Proposed Framework

The EduChrono Smart Timetable Generator which generates the accurate and clash-free timetable using Google OR-Tools and CP-SAT solver. It takes the data in the form of CSV files and follow all the rules and constraints and generates a class-free accurate timetable. It makes sure that classes will start from 9AM, subject is evenly distributed, labs get full 2hours slots, faculty do not clash, and student do not get any free gaps.

A. System Parts

- **User Interface:** A simple interface where HODs, faculty and students were login and see their timetables.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 04, ICTEM 2.0 (2026)

- **Backend System:** This system using Python and Flask. It reads CSV file and generates a timetable.
- **Solver:** Google OR-Tools and CP-SAT solver checks all rules and given constraints and makes best timetable
- **Data Storage:** All data like subject name or code, faculty, rooms, labs and sections is saved in CSV files in MongoDB.

B. Input Data

1. **Teacher Load file**
2. **Room and Lab file**

These files contain:

- Subject name and weekly lectures and labs count.
- Teacher and subject mapping.
- Section details.
- Room and lab numbers and capacity.

C. Rules or Constraints follows

- No faculty takes two lectures at a same time
- No single class can have two lectures at a same time.
- Lab should be is of two hours
- Labs will be continuous manner with lectures.
- Student should not get any free gaps in the timetable.
- COE and PDP will never be come on the same day.
- Labs and Tutorials will be split and follow batch-based rules.

D. Architecture

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 04, ICTEM 2.0 (2026)

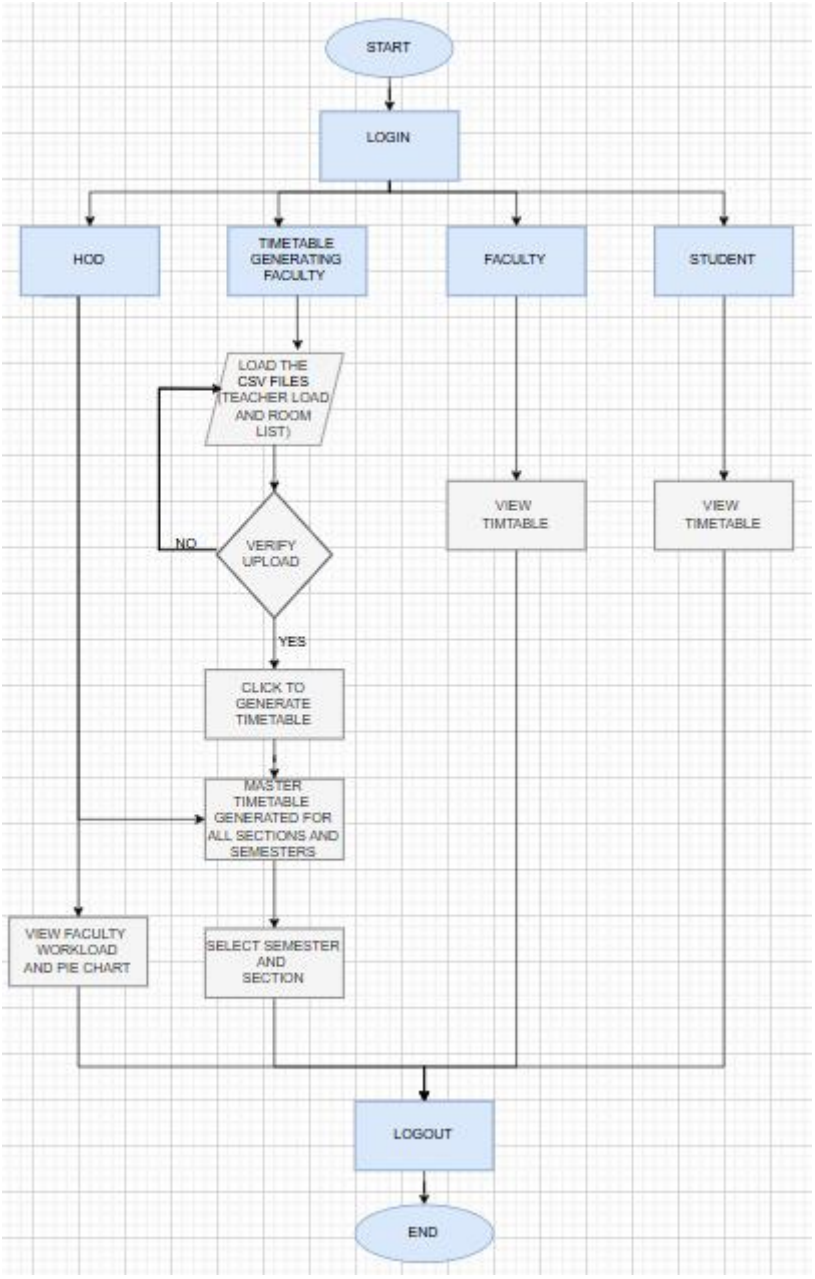

Figure 2: Architecture of EduChrono

E. Optimization Layer

EduChrono uses Google OR-Tools and CP-SAT solver to assign properly:

- Subjects
- Faculty
- Rooms or Labs
- Time slots

F. Output and User Access

- System creates timetable for student and faculty.
- Timetables can be seen and downloadable.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 04, ICTEM 2.0 (2026)

- Teacher load can be shown in the form of pie chart.
- Faculty Convener upload the CSV files and generates timetables.
- Faculty see their scheduled lectures.
- Student see their class timetable.

IV. Implementation

EduChrono were tested using college sample data to check whether the system works properly or not. This testing help to check the accuracy and speed of the system.

A. Technologies Used

Python: For coding.
React: For website design (Frontend)
Flask: For backend work.
Google OR-Tool: For proper scheduling
MongoDB: For data storage.

B. Data Collection and Input Format

Real-time college data was used for testing the system. The system used the following data:

- Subject Details: Subject name and code, lecture or labs.
- Faculty Data: Faculty name and code.
- Room/Lab Data: Rooms and lab availability.
- Section Details: Number of sections and their capacity.

C. Timetable Generation Process

EduChrono follows a rule-based scheduling approach

- No teacher gets two lectures at the same time.
- Each section receives only single lecture of a particular subject.
- Lab sessions are assigned in continuous manner.
- Faculty workload is evenly distributed across the week.

D. Final Output

- Generated a mater Timetable for the entire department.
- Section-wise timetable for individual classes.
- Faculty-wise timetable for individual faculty.
- Student-wise timetables for each student.

E. System Performance Evaluation

EduChrono were tested using real-time college data and the system showed:

- Accurate and conflict-free scheduling
- Generate fast timetables even with large datasets.
- Same and correct result every time.

F. Prototype Development

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 04, ICTEM 2.0 (2026)

EduChrono were made as a web application.

- Frontend: For login and viewing timetable.
- Backend: Use flask for timetable generation.
- User Roles:
  - Admin: Upload CSV and generates timetable
  - HOD: Will see all the timetable and faculty workload graph in the form of piechart.
  - Faculty: View their weekly schedules.
  - Student: View their section-wise timetable.

G. Key Features

- Automatic timetable was generated within few seconds only.
- There is no clash in faculty and rooms assign.
- There is proper allocation of rooms and labs.
- Easy to upload of data through CSV files.
- Section-wise, faculty-wise, and master timetable can be viewed.
- Simple and user-friendly interface.

H. Limitations of the Current System

- Any changes needs then we need to re-generate the timetable.
- Advanced Features like AI-based prediction features are not added yet.

V. Result and Discussion

The performance of the EduChrono Smart Timetable Generator was tested using sample academic data created is CSV format. The development of EduChrono Smart Timetable Generator clearly shows that the generation of automatic timetable reduce the manual work which takes lot of time to generate the timetable but EduChrono generated the accurate timetable within the seconds. This system works with fixed rules so it doesn't make human mistakes.

A. Testing

- Real-time college data were used.
- Data contains subject name or code, rooms, labs, section.
- Different faculty workload was tested.
- Getting accurate data even with large datasets.

B. Performance Measure

Measure	**Result**
Clash-Free Timetable	100%
Teacher Overlap	0%
Room Clash	0%

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 04, ICTEM 2.0 (2026)

Measure	Result
Lab Continuity	Proper
Workload Balance	Balanced
Timetable Generation Time	< 5 seconds

The result shows that the EduChrono generates the 100% clash-free timetable with no faculty overlap and even there is no room clashes. Labs are scheduled in continuous manner and the workload are evenly balanced among all the faculty members. The most important thing is that the EduChrono generates the timetable within 5 seconds with full accuracy and reliable.

C. Clash Detection Results

- No faulty were assigned to two classes at the same time.
- No same room and lab was used by two classed at the same time.
- No section had more than one lecture in the same time slot.

D. Time Performance Analysis

- EduChrono generates the full timetable very fast and accurately.
- EduChrono was very useful system because manual timetable making takes many times.

E. Important System Strengths

- Automatic las were assign in continuous time slot of 2 hours.
- Even distribution of faculty and labs.
- Proper use of classrooms and labs.
- Clear class-wise and faculty-wise timetable output.

F. Comparison with Manual Timetable

Feature	Manual Method	EduChrono
Time Required	Many days	Few seconds
Teacher Clashes	Very common	Not possible
Room Clashes	Often happen	Fully avoided
Workload Balance	Uneven	Balanced
Editing	Difficult	Easy

The manual timetable takes lot of time to generate a timetable while EduChrono generates it within few seconds. In manual work faculty or room clashes may happen easily but in EduChrono completely avoids these problems. EduChrono distributes the workload evenly so that no faculty have continuous two lectures but in manual generation it will happen. Also, making changes in EduChrono is easy as compare to manual work.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 04, ICTEM 2.0 (2026)

G. Limitations Found During testing

- Big change required re-generation of the timetable.
- Currently supported limited departments.

H. Summary

The testing result were show that the EduChrono system were fast and give accurate result. As compared to manual timetable which take lot of time EduChrono generated the accurate timetable within the seconds. This proves that EduChrono is very useful solution for the generation of schools and college timetable. Even it works very well with real-time data.

VI. Conclusion

Making a timetable is one of the most important and difficult tasks in any school or college. When it is done manually, it takes a lot of time and often creates mistakes like faculty clashes, lab problems, and uneven workload. This project presented **EduChrono**, a Smart Timetable Generator that automatically creates correct and accurate clash-free timetables. EduChrono takes all required academic data like subjects, teachers, rooms, sections, and time slots in the form of CSV files. It follows fixed rules to make sure that:

- No teacher gets two classes at the same time.
- No class gets two subjects in one time slot.
- Lab classes get continuous time periods.
- Teacher workload is balanced across the week.
- There is no more than one lecture of any particular subject in a day.

The system was tested using real-time college datasets and it produced fast, clean, and accurate timetables. This proves that automated timetable generation is much better than manual timetable making. EduChrono reduces the of the timetable convenor who creates the manual timetable which takes lot of time. Overall EduChrono is a simple, effective and reliable solution for timetable generation.

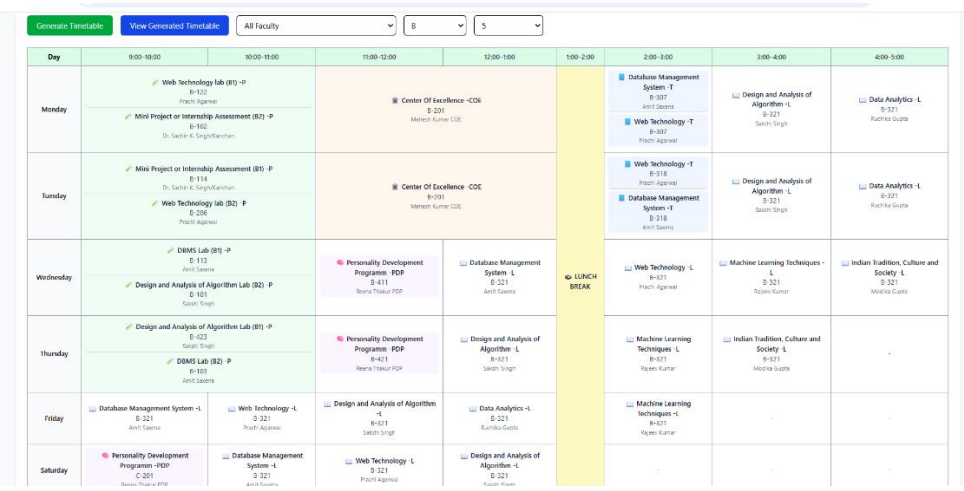

Figure 3: Generated timetable for 5th semester follows tutorial and lab split constraint

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 04, ICTEM 2.0 (2026)

Figure 4: Generated timetable interface for 3rd semester

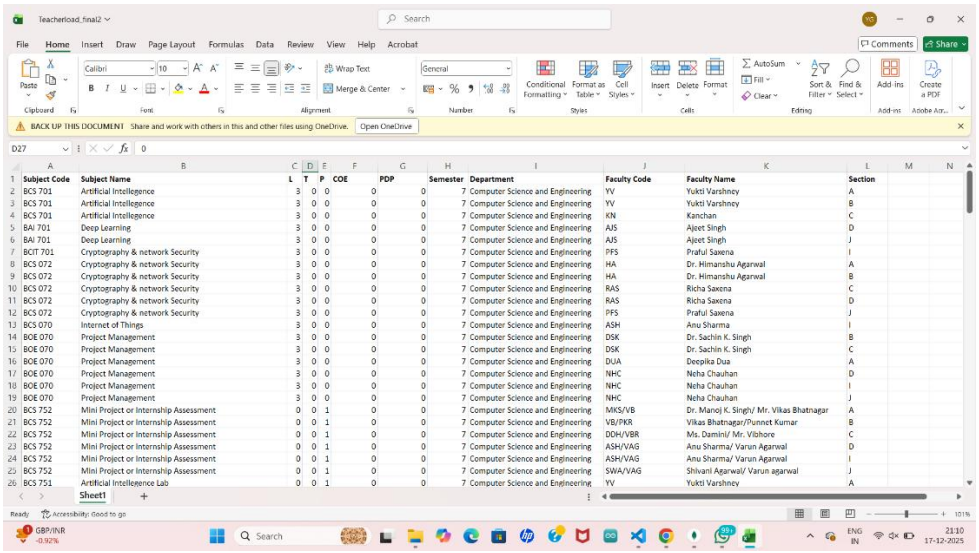

Figure 5: Teacher Load Uploaded File

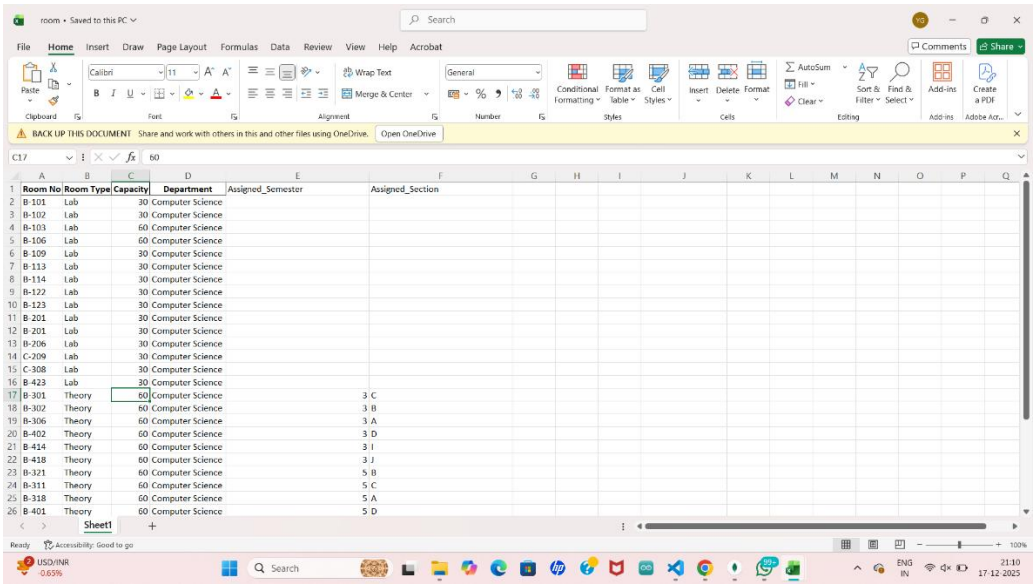

Figure 6: Room List Uploaded File

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 04, ICTEM 2.0 (2026)

VII. References

[1] Mohit K. Kakkar, Jajji Singla, Neha Garg, Gourav Gupta, Prateek Srivastava and Ajay Kumar, "Class Schedule Generation using Evolutionary Algorithms," J. Phys.: Conf. Ser., vol. 1950, no. 1, art. 012067, 2021. doi:10.1088/1742-6596/1950/1/012067
https://iopscience.iop.org/article/10.1088/1742-6596/1950/1/012067

[2] H. Alghamdi, T. Alsubait, H. Alhakami, and A. Baz, "A review of optimization algorithms for university timetable scheduling," Engineering, Technology & Applied Science Research, vol. 10, no. 6, pp. 6410–6417, 2020
https://etasr.com/index.php/ETASR/article/view/3832

[3] M. Santos and G. Paravati, 'PyJobShop: Solving scheduling problem with constraint programming in python, "arXiv:250213483,2025.
https://arxiv.org/abs/2502.13483

[4] S. Choudhary, J. S., and P. Maurya, "A study and analysis of timetable generation using a genetic algorithm," in Proc. 5th Int. Conf. on Advances in Computing, Communication Control and Networking (ICAC3N), Greater Noida, India, 2023,pp.700–703,doi: 10.1109/ICAC3N60023.2023.10541761.
https://ieeexplore.ieee.org/document/10541761

[5] Google Developers, "Google OR-Tools: Getting Started".
https://developers.google.com/optimization/introduction/get_started

[6] N. L. A. Aziz and N. A. H. Aizam, "A brief review on the features of university course timetabling problem," AIP Conference Proceedings, vol. 2016, no. 1, Sep. 2018, Art. no. 020001
https://doi.org/10.1063/1.5055403

[7] E. A. Okonji, Y. M. Oluwatoyin, and O. I. Patricia, "Intelligence Classification of the Timetable Problem: A Memetic Approach," International Journal on Data Science and Technology, vol. 3, no. 2, pp. 24-33, 2017
https://doi.org/10.11648/j.ijdst.20170302.12

[8] R. R. Iwankowicz and M. Taraska, "Self-classification of assembly database using evolutionary method," Assembly Automation, vol. 38, no. 3, pp. 268-281, Jan. 2018
https://doi.org/10.1108/AA-06-2017-071

[9] W. Tian, W. Guo, and M. He, "On the classification of NP complete problems and their duality feature," International Journal of Computer Science & Information Technology, vol. 10, no. 1, pp. 67-78, 2018.
https://doi.org/10.5121/ijcsit.2018.10106

[10] J. Soyemi, J. L. Akinode, and S. A. Oloruntoba, "Automated Lecture Time-tabling System for Tertiary Institutions," International Journal of Applied Information Systems, vol. 12, no. 5, pp. 20-27, 2017.
https://doi.org/10.5120/ijais2017451700

[11] A. E. Phillips, C. G. Walker, M. Ehrgott, and D. M. Ryan, "Integer programming for minimal perturbation problems in university course timetabling," Annals of Operations Research, vol. 252, no. 2, pp. 283-304, May 2017
https://doi.org/10.1007/s10479-015-2094-z

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 04, ICTEM 2.0 (2026)

[12] M. F. Syahputra et al., "Genetic algorithm to solve the problems of lectures and practicums scheduling," IOP Conference Series: Materials Science and Engineering, vol. 308, Feb. 2018, Art. no. 012046 https://doi.org/10.1088/1757-899X/308/1/012046

[13] R. A. Oude Vrielink, E. A. Jansen, E. W. Hans, and J. van Hillegersberg, "Practices in timetabling in higher education institutions: a systematic review," Annals of Operations Research, vol. 275, no. 1, pp. 145-160, Apr. 2019
https://doi.org/10.1007/s10479-017-2688-8

[14] O. Y. M. Al-Rawi and T. Mukherjee, "Application of Linear Programming in Optimizing Labour Scheduling," Journal of Mathematical Finance, vol. 9, no. 3, pp. 272-285, Jun. 2019
https://doi.org/10.4236/jmf.2019.93016

[15] I. R. Cassar, N. D. Titus, and W. M. Grill, "An improved genetic algorithm for designing optimal temporal patterns of neural stimulation," Journal of Neural Engineering, vol. 14, no. 6, Nov. 2017, Art. No. 066013 https://doi.org/10.1088/1741-2552/aa8270

[16] S. Muniyappan and P. Rajendran, "Contrast Enhancement of Medical Images through Adaptive Genetic Algorithm (AGA) over Genetic Algorithm (GA) and Particle Swarm Optimization (PSO)," Multimedia Tools and Applications, vol. 78, no. 6, pp. 6487-6511, Mar. 2019
https://doi.org/10.1007/s11042-018-6355-0