IJNRD.ORG ISSN : 2456-4184

INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG

An International Open Access, Peer-reviewed, Refereed Journal

# SUMMER-TERM TIMETABLE GENERATION

**[1]Naveen Kumar M, [2]Mohan S G, [3]Madhu K, [4]Asad Mohammed Khan**

[1]UG Students SCSE, [2]UG Students SCSE, [3]UG Students SCSE, [4]Assistant Professor SCSE

Presidency University, Banglore-560 064

*Abstract:* The Timetable Generation is designed to be a software solution to make the operation of creating timetables easier. Currently, it is done manually, thus time-consuming and susceptible to error. This system automates the insertion of periods into the timetable, allowing smooth management and easy access for the faculty through a mobile application. Additionally, in cases where a teacher doesn't show up, arrives late, or leaves early, the timetable adjusts automatically. For purposes of optimum workload distribution, the system therefore specifies maximum and minimum teaching hours on a daily, weekly, and monthly basis for each faculty member. The software will also allow users to request leaves with information on date, reason, and a substitute faculty member. When choosing a substitute, the system allows access to their schedule confirming if they are available during such a requested period. The chosen substitute can accept or decline the request. Also, the headmaster has a right to study leave requests and the responses of substitutes before arriving at a conclusion. This integrated schedule management system is quite helpful for colleges as it avoids the complications that are associated with manual schedule preparation. This software helps faculty members easily view their schedules on their cell phones, making this process of school improvement very efficient and well-organized.

***IndexTerms* - Component,formatting,style,styling,insert.**

## INTRODUCTION

The Automatic Timetable Generator is the new software application that autonomously creates and manages academic schedules. Conventionally, management of the timetable was quite labor-intensive, time-consuming, and prone to inefficiencies and scheduling conflicts. This application erases these problems because it comes as a reliable and efficient solution for dynamically generating optimized timetables based on constraints such as faculty availability, subject schedules, and institutional policies. It simplifies and automates timetabling to such an extent that the backend can be seen to reduce greatly the administrative load on the whole task, boosting productivity. Automatic Timetable Generator has a backbone based on Python and Django as well, due to their capacity for handling difficult tasks of processing and scaling well in complex processes. SQLite3 is used, because it assures the security of retrieving and storing data efficiently. This architecture would allow a lot of scheduling data to be stored while ensuring the possibility of updating it in real time with the help of automated conflict resolution. Scalable and flexible software enables the handling of the requirements of any type and size of educational institution: small schools and huge universities alike. During the times of maximum demand, when unexpected calendar updates or changes of personnel are suddenly requested, the application is effective, responsive, and adaptable to modification. The Automatic Timetable Generator also allows for easy user navigation and is therefore quite accessible to both faculty members and administrators, allowing for easy generation of timetables with little effort. It supports advanced filtering capabilities, so the users can get their preferred timetables with regard to particular requirements such as subject preferences, room availability, and faculty workload constraints. This approach is so simple that all stakeholders in an institution can access and manage as well as edit schedules without technical knowledge. In addition to creating timetables, the application improves resource allocation by preventing scheduling overlaps and optimizing faculty workload distribution. It considers multiple constraints so that no faculty member is overburdened while maximizing classroom utilization. This functionality contributes to a more balanced and efficient academic environment with minimal disruptions caused by scheduling conflicts. Data security and privacy are critical aspects of the Automatic Timetable Generator. Given that academic schedules and faculty availability data contain sensitive information, the software employs robust encryption techniques and adheres strictly to data protection protocols. Secure access mechanisms and stringent data security policies build trust among educational institutions, faculty members, and administrative staff, ensuring that all information remains confidential and well-protected. Looking ahead, future enhancements for the Automatic Timetable Generator include integrating predictive analytics to optimize future schedules based on historical data trends. The system could leverage machine learning algorithms to identify patterns in scheduling, helping administrators make proactive decisions that improve efficiency. Additionally, integrating faculty leave management systems and real-time notifications will further streamline the scheduling process. These developments make the system perform not only automatized generation but also proactive prevention of scheduling confict before actual conflicts arise in a schedule. Some of the possible upgrades are that the cloud-based functionality would enable access and management of timetables from any location where internet access is available. The existing LMS and SIS could be integrated into this to allow seamless coordination of scheduling with other academic activities. The mobile-friendliness or even a specific mobile application can be a way to enhance access so that faculty members and administrators can easily view and edit their schedules using their smartphones or tablets. This could also be a future update with an AI-based optimization, where the artificial intelligence algorithm can constantly change the timetabling based on the dynamic factors such as faculty unavailability, emergency closure, or preference of students. Smart algorithms in the system may automatically suggest the very best possible alternative for scheduling, hence further improving efficiency and reducing manual interventions. The Automatic Timetable Generator is, in short, an innovative academic scheduling tool that helps minimize administrative work, improve efficiency, and minimize scheduling conflicts. It changes the face of educational institution management in terms of timetable management by virtue of its real-time data processing, user-friendly design, and optimization techniques. The more it evolves to incorporate predictive analytics, cloud-based capabilities, and AI-driven decision-making, the more it will redefine the academic scheduling landscape as an indispensable asset for educational institutions around the world.

### 1.1 The need for Automatic Timetable Generator in society

The Automatic Timetable Generator plays a key role in most modern educational institutions by automating complex academic scheduling tasks. Removal of inefficiencies in manual scheduling ensures that the faculty and students receive properly structured timetables that can avoid conflicts, making proper use of resources. Dynamic real-time adjustments of schedules in such a system ensure seamless operation in academic settings and have benefited both educators and students. Other than the individual advantage, the application helps to have an efficient institution by preventing overcrowding in classes and having balanced faculty workloads. The management of administration by the application ensures proper academic planning and assists the institutions to manage their resources properly. Also, by integration with mobile applications, the faculty can access their schedule conveniently and improves time management and communication. More so, an Automatic Timetable Generator increases institutional flexibility by accepting changes at the last minute due to faculty unavailability or room reassignments. Through artificial intelligence and optimization algorithms, the system may analyze faculty preferences, batch sizes of students, and subject priorities to create schedules that are best possible. This also eliminates the possibility of human error and biases in the creation of the schedules, hence creating fairness and transparency. Such a system easily integrates with the learning management platform, enhancing further academic coordination such that students and faculty are able to receive current updates through notifications and modifications in real time.

### 1.2 Contemporary Usage of the Application in Academic Scheduling

The Automatic Timetable Generator uses complex algorithms and data processing techniques to efficiently manage the scheduling of academics. The entire process is automated, taking into account faculty availability, course requirements, and institutional policies. In real-time, any changes in faculty schedules or course allocations are reflected, thereby reducing scheduling conflicts and improving academic coordination. The integration of mobile platforms helps faculty and students to access their schedules from remote locations, with automatic updates and notifications. Features such as automatic room allocation, workload distribution analysis, and customization of the schedule are also offered by the software, making it highly adaptable to various academic environments. By streamlining academic scheduling, the system enhances institutional efficiency and improves the overall learning experience. The Automatic Timetable Generator also reduces administrative workload by not requiring manual amendments and constant rescheduling. It maximizes the optimal use of all available resources, including classrooms and faculty, by balancing workloads and avoiding the overlapping of sessions. Besides these benefits, the system can generate reports and insights on faculty engagement, classroom occupancy, and course distribution that help make strategic decisions. Providing an organized and conflict-free schedule is one of the reasons the software improves academic productivity while making learning more structured and effective.

### 1.3 Benefits and Challenges of Automatic Timetable Generator

The Automatic Timetable Generator brings several benefits that range from enhanced efficiency, lightened administrative workloads, and faculty-student coordination. Automated generation of timetables by institutions can minimize human errors and inconsistencies, resulting in a better structured academic calendar. The capacity of the software to adjust the schedules dynamically as faculty members leave or change the requirements of a course makes the product even more practical and applicable. However, the application also faces challenges, such as the need for accurate data input and the complexity of managing multiple constraints simultaneously. Institutions must ensure that faculty availability and course details are consistently updated to maintain schedule accuracy. Additionally, integrating the system with existing academic management software may require technical expertise and infrastructure investment. Despite all these challenges, the Automatic Timetable Generator remains a very valuable tool for optimizing academic scheduling with further development and refinement.

### NEED OF THE STUDY.

In academic institutions, the design of a summer-term course schedule is a challenging and time-consuming process. Manual scheduling tends to result in such problems as conflicts of interest among classrooms, faculty overloads, and poor room utilization, particularly when there are few resources available and several constraints have to be satisfied. To overcome these problems, this research aims to implement a Summer-Term Timetable Generation System through web technologies such as HTML, CSS, JavaScript, and Bootstrap, coupled with a Genetic Algorithm (GA) for optimization on the backend. The main intention is to optimize the process of generating timetables automatically, meeting various constraints like course specifications, instructor schedule, room size, and time slot preferences. Using Genetic Algorithms, the system mimics natural selection principles to arrive at near-optimal scheduling solutions.

### 3.1 Population and Sample

The population to be used for this study entails all departments, instructors, classrooms, and summer-term courses across an academic organization. These make up the vital building blocks that are needed in order to produce a working timetable. The system is designed to support multiple departments, varying numbers of instructors who have varying availability, and several classrooms with varying capacities and restrictions. For the purposes of testing and development, a sample dataset has been developed involving three academic departments, ten instructors, and fifteen courses taught in the summer term. Each course is linked with particular time periods and department affinities, and instructors by their availability and subject matter specialty. Classrooms are classified according to capacity and availability by time slots. This organized sample replicates a practical scheduling situation and is the input to the Genetic Algorithm employed by the system.

### 3.2 Data and Sources of Data

The information employed within this project is artificial and made to simulate realistic conditions for use in academic timetabling generation. The data contains course information like course name, allocated departments, and durations required. Instructor data involves instructor name, subject area expertise, and time slots available. Room data contains room numbers, room capacities, and availability during the day. The user interface of the system is constructed with HTML for structuring content, CSS and Bootstrap for styling and responsive design, and JavaScript for dynamic interaction. The scheduling algorithm is fully written in JavaScript based on a Genetic Algorithm, which works on the dataset to create and optimize timetable solutions. This artificial dataset provides a controlled test environment and guarantees that the performance of the algorithm can be tested and optimized step by step.

### 3.3 Theoretical framework

The theoretical context for this research relies on an optimization model where a Genetic Algorithm is used to create conflict-free and resource-aware academic timetables. Here, the dependent variable is the optimality and feasibility of the created timetable and the independent variables are the courses, instructors, rooms available, and time slots. The Genetic Algorithm works through an imitation of natural selection. It starts by generating an initial population of random timetabling configurations. Each configuration is assessed against a fitness function that verifies for constraints like conflicting classes, room capacity violations, or absent instructors. The most promising solutions are chosen for crossover and mutation to generate new generations of timetables. This evolutionary process goes on until an optimal or near-optimal solution is achieved. The model allows for flexibility and adjustment, which allows the system to generate valid timetables even if inputs vary or constraints are increased.

## RESEARCH METHODOLOGY

### 3.1 Population and Sample

Population of this research consists of scheduling needs of educational institutions over the summer term, including courses, teachers, rooms, and time slots. The emphasis is placed on automated timetable building by combining web development technologies with an optimization algorithm to solve usual scheduling problems of overlapping classes, uneven workloads, and non-optimal use of space. The test sample for system development and testing consists of artificial data for three departments, ten teachers with different time availability, fifteen summer-term courses, and five classrooms with different capacities. Every item in the data set is crucial in creating valid and optimized schedules. The scheduling logic logic views every course-instructor-room-time allocation as a possible gene in a chromosome, which is processed through a Genetic Algorithm to develop the best-fit scheduling solution. This instance is adequate to mimic a realistic academic timetable problem and test the system's feasibility.

### 3.2 Data and Sources of Data

The research uses synthetically produced secondary data customized to replicate the situation of an actual academic institution's summer-term scheduling procedure. This data comprises course details such as course codes, names, lengths, and departmental affiliations; instructor profiles such as names, departments, and availability of times; and classroom details such as room numbers, capacities, and availability of times. The system's user interface is built in HTML for structure layout, CSS and Bootstrap for styling and responsive design, and JavaScript for interactivity at the front-end. The scheduling rules are coded all in JavaScript by employing a Genetic Algorithm, where the sample data is input and used to generate and improve timetables. No actual institutional data was utilized, allowing room for flexibility when developing and testing. The system's output is assessed based on constraint satisfaction such as prevention of class clashes, instructor availability matching, and room capacity adequacy. Utilization of secondary, well-defined sample data facilitates effective testing of the system's optimization and front-end rendering features.

### 3.3 SOFTWARE SPECIFICATIONS:

**a) FRONT-END PART:**

- HTML (Hyper-Text Markup Language)
- CSS (Cascading Style Sheets)
- JAVASCRIPT
- Bootstrap

**b) BACK-END PART:**

- Genetic Algorithm
- Sqlite-3 (Database)

## OUTCOMES:

### 8.1. Time Efficiency

Manual creation and management of timetables take a lot of time, especially for large institutions with many courses, students, and faculty members. The automatic generation of timetables by the system reduces the time spent on these activities since scheduling and resource allocation are automated.

**Important Features Contributing to Time Efficiency:**

**Automated Scheduling:** The system draws data from the database and automatically creates a whole schedule in a matter of minutes, without manual input and alteration.

**Bulk Data Management:** Simultaneous input of multiple teachers, rooms, and courses will make setup much easier.

**Real-time Modifications:** Changes to schedules, such as adding a new course or modifying a time slot, are instantly reflected across the system.

**Benefits:**

Faster response times for last-minute changes or adjustments.

Saves ample time for the administrators to devote to other major tasks.

Prepares timetables ahead of time for the academic term, thus avoiding late bookings.

### 8.2. Errors:

Errors are inherent when using manual scheduling. They include overlapping classes, breaking room capacities, and the simultaneous assignment of many tasks to a single instructor. The automated system eliminates this by allowing stringent validation checks during the generation of timetables.

**Error-Prevention Mechanisms:**

**Conflict Detection:** The system ensures that no two classes are assigned to the same room or instructor at the same time.

**Capacity Management:** Rooms are allocated based on their seating capacity, which prevents overcrowding.

**Workload Limits:** The system enforces pre-defined teaching workload limits so that faculty members are not overloaded.

**Benefits:**

Minimizes disruptions caused by schedule conflicts.

Hospital Finder

### 8.3. Increased Accessibility:

A mobile application will facilitate real-time accessibility of schedules and notifications for faculty and administrators. This makes it accessible to anyone interacting with the communication and coordination process.

**Accessibility Features:**

Application Friendly Design: It is responsive, and access can be gained on smartphones, tablets, and computers.

**Push Notifications:** Always alerting users regarding changes in schedule, cancellation, or other announcements.

**Cloud Integration:** Allows remote access to timetables from anywhere with an internet connection.

**Benefits:**

It ensures that the faculty and administrators are always aware of the updated schedules.

Reduces the number of printed timetables, thus promoting sustainability.

It provides flexibility for the users to access information on the go.

### 8.4. Balanced Workloads:

It ensures that the teaching load is spread impartially among all faculty members, preventing burnout. It also keeps everyone at peak productivity based on availability, expertise, and limits of load.

Workload Management Features

**Pre-set Limits:** Administrators can predefine the maximum teaching hours for each faculty member.

**Dynamic Adjustments:** The system always adjusts to the eventual changes in faculty absence or courses added, maintaining a balance.

**Graphical Reporting:** Visualizing workloads aids administrators in identifying and managing overload.

**Advantages:**

Prevents overloading the faculty.

Resulting faculty morale and job satisfaction are higher.

Quality teaching will be delivered because no instructor will be overwhelmed with work.

Meets institutional guidelines on the allocation of work.

**8.5. Scalability and Flexibility:**

It is developed with the idea that the system grows with the institution, encompassing more courses, faculty members, students, and rooms. The flexibility it exhibits ensures it adapts to evolving needs and changes in trends within education.

**Scalability Capabilities:**

**Modular design:** New functionality or modules can be added smoothly, such as online attendance recording or resource monitoring.

**Optimal Performance:** As the data input increases, this system does not lose its velocity.

**Customized:** The software can be changed to suit individual needs of institutes.

Adaptability Features

**Policy Updates:** The system integrates new scheduling policy or academic systems easily.

**Multiple Configurations:** Supports different types of timetabling models: block scheduling as well as semester-based systems

**Benefits:**

Keeps the usability of the system for a very long time.

Eliminates the requirement of frequent overhauls or replacement.

Facilitates growing demands of expanded institutions.

**8.6. User Satisfaction:**

The effectiveness, reliability, and simplicity of the system contribute to an overall better experience of faculty, administrators, and students. Intuitive interface, automation, and no errors create higher user satisfaction.

**User-Friendly Features:**

**Simple Navigation:** Simple dashboards with easy-to-understand instructions will help the user navigate through the system easily.

**Fast Deployment:** There is little deployment time for a new user or institution.

**Feedback Channels:** The schedules can be requested to be modified or provided with feedback, promoting teamwork.

**Benefits:**

Increased faculty and administrative acceptance

Stress reduction through automatic scheduling

The institution gains credibility through an organized and effective approach.

## IV. RESULTS AND DISCUSSION

The Automatic Timetable Generation System provides a very efficient and user-friendly way of generating and managing institutional timetables. The system minimizes errors and optimizes the use of available resources by the automation of the scheduling process and the minimizing of effort in manual operations. It uses technologies such as HTML, CSS, JavaScript, Django, and SQLite3 to provide a highly strong and scalable solution to this problem. This chapter describes the results obtained by the system and its effects along with its limitations and possible future improvements.

**Results**

**Efficiency in Time Saving:**

The program saves a tremendous amount of time for generating and managing timetables.

Automation: Assignments of time slots, conflict avoidance, and equal distribution of loads are done within seconds.

**Effect:** There is no manual interference, which conserves the most precious administrative hours.

**Outcome:** Institutions can strategize more on education rather than operational issues.

issues. This feature ensures timely access to healthcare without the need to travel long distances or navigate through busy healthcare systems, reducing delays in receiving medical attention.

**Conflict-Free Timetable Generation:**

There will be no overlapping or overlapping of timetables generated in the system.

Validation Mechanisms: Internal checks prevent over-lapping time periods, over-booking rooms, or teachers' workload to be exceeded.

**Impact:** Error-free common with manual scheduling that is smooth and reliable.

**Outcome:** Increased confidence and reliability of the scheduling process, which works in favor of administrators, teachers, and students

**Ease of Accessibility:**

The software provides real-time access to timetables to all stakeholders.

**Responsive Design:** The system is available on any gadget, from desktop to smartphones.

**Key Features:** Teachers and administrators can at any given time view their timetables, edit them, or approve the updated ones.

**Impacts:** All stakeholders are always updated and up-to-date.

**Balanced Workloads:**

The system distributes teaching work among faculty members in a balanced way.

**Workload Distribution:** The algorithms distribute classes in a manner that satisfies predefined workload limits.
**Impact:** Does not overload any one faculty member and makes sure that all the teachers are taught a reasonable number of classes.
**Outcome:** Faculty satisfaction goes up, and institutional policies are obeyed.

**Scalability and Adaptability:**
The system is created with the dynamic necessities of institutions.
**Scalability:** The system will be able to support adding extra teachers, rooms, courses, or departments while maintaining performance.
**Adaptability:** The system can be designed to meet very specific institutional needs, such as different class-lengths or some departmental constraint.
**Impact:** Makes the application future-ready with long-term use.

**User Satisfaction:**
The efficiency and reliability of the system and easy usability lead to higher user satisfaction.
**Intuitive Interface:** Very clean and simple design makes for easy navigation and operation.
**Feedback Mechanism:** The system can receive feedback from users for its development and evolution.
**Outcome:** This promotes the frequent use of the system and also helps to build confidence in the system.

**Discussion:**
**Effect on Administrative Productivity:**
The system saves a lot of administrative effort by automatically performing complicated scheduling operations.
**Key Points:** The process of assigning teachers, rooms, and resolving conflicts is automated and saves manual efforts.
**Challenges:** Proper data entry is crucial for producing accurate timetables.

**Error-Free Scheduling:**
The system has strong validation mechanisms that ensure the generation of error-free timetables.
**Effectiveness:** The system eliminates scheduling conflicts and ensures compliance with institutional policies.
**Challenges:** Unique constraints or exceptions may require more customization

**Real-Time Accessibility:**
The application ensures that timetables are readily accessible to stakeholders.
**Key Benefits:** Teachers can view their schedules, administrators can make edits, and students can check their classes in real time.
**Challenges:** Ensuring consistent uptime and performance during peak usage periods

**Integration with Institutional Systems:**
The system can be integrated with other institutional platforms for smooth operation.
**Opportunities:** Integration with attendance tracking or academic management systems can enhance functionality.
**Challenges:** Compatibility with legacy systems and data synchronization are potential hurdles.

**Data Privacy and Security:**
Protecting institutional and personal data is a critical aspect of the application.
**Privacy Measures:** Secure login, encrypted database storage, and compliance with data protection policies ensure data security.
**Challenges:** Preventing unauthorized access and safeguard against data breaches need constant monitoring and updates.

**Scalability and Future Enhancements:**
The system is designed to accommodate large institutions and be responsive to the changing needs.
**Future Features:** Integration of features like AI-based timetable optimization, automated notification systems, or multi-campus support.
**Challenges:** Balancing feature expansion with system performance and usability.

**Cost-Effectiveness and Affordability:**
The system helps institutions save on costs by minimizing manual effort and optimizing resources.
**For Institutions:** Cost savings in administrative time and effort get reflected in lower operations.
**For Users:** Time saved in accessing and managing their schedules contributes to efficiency as a whole Challenges: Making it affordable for smaller institutions which have limited budgets.

**User Feedback and Continuous Improvement:**
User input and usage analytics will guide the development of the system.
**Feedback Mechanism:** The system gets updated regularly based on user input so it remains relevant and effective.
**Continuous updates:** adding features and improving performance based on user needs and technological advancements.
**Challenges:** Managing user expectations while maintaining a streamlined development process.

## CONCLUSION:

The proposed The Automatic Timetable Generator brings an unmatched level of flexibility and precision to the scheduling process. The software ensures that every timetable is not only conflict-free but also optimally tailored to the needs of all stakeholders by integrating factors like faculty availability, room constraints, and institutional policies into its algorithm. The application allows for a teaching environment that avoids overburdening faculty members with too many classes, ensures each classroom is utilized to the fullest extent, and holds administrative staff to perform strategic tasks meant to maintain an academically effective and harmonious environment. Scalability allows it to be used for any educational institution of size, from a small school to a giant university. Whether one is dealing with handfuls of faculty members or thousands, this Automatic Timetable Generator will be able to handle all the demands an academic institution might have on it. This ensures that no matter what changes occur in educational institutions, this system will be relevant and useful. Looking ahead, predictive analytics will further enhance the capabilities of the platform by allowing institutions to anticipate and proactively resolve scheduling issues before they even arise. This forward-thinking approach ensures that the system not only responds to current needs but also adapts to future trends and challenges in academic scheduling. In the final analysis, the Automatic Timetable Generator is far more than just an automatic timetable generator-it is a holistic solution that transforms the overall academic planning approach of an institution. By automatically using real-time data and high-end optimization techniques, it equips educational institutions with a better ability to run their affairs and facilitates faculty and administrative staff to concentrate on what matters: delivering quality education. This is a move that represents one giant leap into the future for academic scheduling; powerful, reliable, and user-centric, which makes it work well for anyone who is engaged in the education process.

## REFERENCES

**[1].** Tavakkol, M., & Parsa, M. (2021). A Hybrid Genetic Algorithm for University Course Timetabling Problem Considering Faculty Preferences. Computers & Industrial Engineering, 157, 107327.
[https://doi.org/10.1016/j.cie.2021.107327]

**[2].** Rong, Q., & Lee, K. (2022). Multi-Objective Optimization for University Timetabling Problem: A Comparative Study of Algorithms. Journal of Scheduling, 25(1), 57-72.
[https://doi.org/10.1007/s10951-021-00788-3]

**[3].** Hassan, M., & Khalil, M. (2023). An Intelligent Course Scheduling System Using Machine Learning Techniques. Journal of Educational Computing Research, 61(3), 445-465.
[https://doi.org/10.1177/07356331221122514]

**[4].** Wang, X., & Xu, H. (2021). A Novel Memetic Algorithm for Solving University Timetabling Problems. Expert Systems with Applications, Volume 178, Article 115018.
[https://doi.org/10.1016/j.eswa.2021.115018]

**[5].** Pillay, N., & Qu, R. (2022). An Evolutionary Algorithm for the Multi-Criteria University Timetabling Problem. Applied Soft Computing, Volume 115, Article 108163.
[https://doi.org/10.1016/j.asoc.2021.108163]

**[6].** Nguyen, T. T., & Le, M. T. (2021). A Deep Reinforcement Learning-Based Approach for Automated Course Scheduling. IEEE Access, Volume 9, 2021, Pages 115765-115778.
[https://doi.org/10.1109/ACCESS.2021.3106042]