# CHAPTER 1

**INTRODUCTION**

## Introduction to Project

A timetable is a key administrative component in any educational institution. Creating one is typically a complex and demanding process. Coordinators or administrators must consider various factors during scheduling, which becomes particularly stressful in colleges due to multiple departments, diverse subjects, and faculty members with varying designations. They must carefully allocate courses, classrooms, and faculty based on subject requirements, and patiently construct an efficient timetable to ensure an organized schedule and smooth functioning of the institution. Despite the increasing automation of administrative tasks in colleges, lecture timetables are still often prepared manually due to the inherent complexities of the process.

Timetable scheduling problem is inherently complex due to the numerous interrelated factors and constraints, including teaching plans, courses, instructors, administrative and teaching classes, classrooms, and time slots. In universities, the complexity increases significantly during semester scheduling, as it must accommodate multiple student batches, various student groups, elective subjects (with different groups choosing different electives), and common mandatory courses taken by all the students. It is essential to ensure that there are no clashes among student groups, faculty members, and lecture halls during the scheduling process.

Timetable scheduling is widely recognized as a combinatorial optimization problem and a constraint satisfaction problem where the goal is to find a solution that satisfies a predefined set of constraints. These constraints are generally categorized as hard and soft constraints. Hard constraints are mandatory and must be satisfied for a timetable to be feasible, such as preventing classes from clashing in the same room or a teacher having more than one class simultaneously. Soft constraints, on the other hand, represent preferences or desired conditions that ideally should be met but are not strictly required for feasibility. Meanwhile violating soft constraints is acceptable, but an optimal timetable minimizes such violations.

This project aims to address the difficulties of generating timetables by providing an automated scheduling system. Automating the process helps save time, avoid the complexity of manual management, and reduce documentation work. The goal is to generate timetables that are more accurate, precise, and free of human errors. An automated system ensures up-to-date and accurate information.

To achieve automatic timetable generation, the project proposes the use of algorithms such as genetic, heuristic, and resource scheduling. Genetic algorithms (GAs) are frequently applied to timetabling problems, which are known to be NP-hard optimization problems, because GAs are effective for finding optimal or near-optimal feasible solutions among a complex set of variables and constraints. They are based on natural selection and evolution principles and are known for their robustness in solving complex combinatorial problems. Heuristic approaches are also commonly used in timetabling, either independently or as components within algorithms like GAs, often focusing on scheduling the most constrained elements first. Resource scheduling is another algorithmic approach listed for addressing these problems. These algorithms incorporate various strategies aimed at improving the efficiency, scalability and reliability of the timetable generation process.

The system will generate the timetables based on various inputs, including the number of subjects and teachers, teacher workload, semester details, and subject priorities. Additional necessary inputs include faculty details, subject details (including name and code), workload based on faculty designation, faculty and subject allotment based on time slots, and details of theory and lab courses handled by each faculty. Classroom or room details, including availability and capacity, are also crucial inputs. By relying on these inputs and utilizing optimization algorithms, the system will generate possible timetables for the working days.

In conclusion, the proposed solution aims to streamline the scheduling process, enhance accuracy, and offer a more efficient, scalable, and reliable approach to timetable generation, ultimately contributing to smoother academic operations and a better teaching-learning experience.

# CHAPTER 2

**LITERATURE SURVEY**

## 2.1 Introduction

Timetable scheduling is an NP-hard problem, meaning it involves a high level of computational complexity, and finding an exact solution is often infeasible within a reasonable time for large instances. Therefore, the goal is typically to find an optimal or near-optimal solution within a complex set of variables and constraints. As a result, extensive research has been conducted on applying optimization techniques to address this challenge.

This literature review explores various existing approaches to automated timetable generation, with a particular focus on Genetic Algorithms (GAs). GAs have gained popularity due to their effectiveness in solving NP-hard optimization problems like timetabling. They operate on the principles of natural selection and evolution, using mechanisms such as selection, crossover, and mutation to iteratively improve solutions. Their ability to efficiently explore large and complex search spaces makes them well-suited for generating feasible and optimized timetables under multiple constraints.

## 2.2 Literature Survey

Mittal et al. [1] addressed the timetable scheduling problem by proposing the use of Genetic Algorithms (GA) to automate and optimise the scheduling process. Their observation says that manual scheduling is time-consuming and prone to errors due to complexity of satisfying various constraints. The authors classify these constraints into two categories: hard constraints such as avoiding conflicts while assigning the classes and rooms; and soft constraints, such as minimizing idle gaps for students lectures and distributing teaching assignments evenly across the week. The authors applied GA where the algorithm initialises the population of guesses, then three operators are applied- selection, crossover and mutation to create an optimal timetable. The system was tested with a real data within the author’s institution. The results demonstrated significant improvements in both efficiency and accuracy compare to manual scheduling. Although the study succeeded in creating a more efficient alternative to manual scheduling, it lacked mechanisms for dynamic adaptability and deeper real-world constraint handling.

Abdelhalim and El Khayat[2] introduced a Utilization-based Genetic Algorithm (UGA) to solve the university course timetabling problem with a focus on optimizing space utilization alongside satisfying scheduling constraints. Their approach employed a two-dimensional chromosome representation, mapping events to room-timeslot pairs, and generated the initial population using heuristic methods: Largest Degree First (LD) and Largest Enrollment First (LE), the authors proposed a utilization-based crossover operator aimed at reallocating underutilized events and a targeted mutation strategy employing local search to optimize room occupancy. A composite fitness function weighted towards occupancy rates, frequency rates, and minimizing scheduling gaps was used to evaluate solutions. The system was tested on real data from Alexandria University the dataset comprised of 337 events, 32 rooms and 131 professors also, the system was tested against two benchmark datasets from the International Timetabling Competition (ITC 2007). The Results demonstrated significant improvements in space utilization, reduced scheduling hours, and better overall timetable quality compared to manual scheduling. However, the study identified gaps in dynamic real-time adaptability and slight computational overhead for medium-sized problems.

Colorni et al. [3] proposed a Genetic Algorithm (GA) approach to solve the highly constrained school timetabling problem. Their method used a two-dimensional matrix representation of teachers and timeslots, with a hierarchical objective function focusing first on feasibility, second on management rules, and finally on individual teacher preferences. This method uses customized genetic operators, like a row-based crossover and order-k mutation, along with a genetic repair (filtering) process to correct infeasibilities. The integration of local search significantly enhanced timetable quality, achieving a substantial reduction in cost compared to manual scheduling and simulated annealing. However, the approach faced scalability challenges due to the computational complexity of the repair process and lacked mechanisms for dynamic timetable adjustments.

Othman et al. [4] presented a Genetic Algorithm (GA) approach for university course timetabling, ensuring all hard constraints were satisfied while minimizing soft constraint violations. Each chromosome encoded subject, section, instructor, time, and room data. The method employed rank-based selection, single-point crossover, and random mutation, with exclusiveness to retain the best solutions across generations. A fitness function was designed to normalize and weigh both hard and soft constraint violations, covering instructor time preferences, room assignments, course prerequisites, and overload control. Tested on real data from the University of Jordan, the algorithm achieved zero hard constraint violations and significantly reduced instructor overload and scheduling conflicts. However, the model lacked real-time adaptability, sensitivity to constraint weights, and user interaction capabilities, limiting its flexibility and scalability in dynamic environments.

Han and Wang [5] proposed a hybrid approach called POGA-DP, combining Genetic Algorithms (GA) with Dynamic Programming (DP) to solve the University Course Scheduling Problem (UCSP), particularly for complex joint-course timetables. The GA component optimizes time slot assignments using a swap-based mutation with a repair mechanism, while DP allocates classrooms to minimize seat wastage and improve utilization. The method achieved significant improvements in scheduling quality (up to 46.99%) and reduced classroom usage by 29.27% compared to standard GAs. Tested on data from Beijing Forestry University, the approach outperformed GA, Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and Producer–Scrounger Method (PSM) across multiple metrics, including fitness and occupancy rate. However, the model lacks real-time adaptability, parameter tuning, cross-institutional testing, and practical integration into university systems, limiting its deployment readiness.

### 1.3.1 Sub titles

### 1.3.2

### 1.3.3

## REFERENCES

[1] Mittal, Dipesh, Hiral Doshi, Mohammed Sunasra, and Renuka Nagpure. "Automatic timetable generation using genetic algorithm." *International Journal of Advanced Research in Computer and Communication Engineering* 4, no. 2 (2015): 245-248.

[2] Abdelhalim, Esraa A., and Ghada A. El Khayat. "A utilization-based genetic algorithm for solving the university timetabling problem (uga)." *Alexandria Engineering Journal* 55, no. 2 (2016): 1395-1409.

[3] Colorni, Alberto, Marco Dorigo, and Vittorio Maniezzo. "A genetic algorithm to solve the timetable problem." *Politecnico di Milano, Milan, Italy TR* (1992): 90-060.

[4] Alsmadi, Othman M. K., S. Za'er, Dia I. Abu-Al-Nadi, and Alia Algsoon. "A novel genetic algorithm technique for solving university course timetabling problems." In *International Workshop on Systems, Signal Processing and their Applications, WOSSPA*, pp. 195-198. IEEE, 2011.

[5] Han, Xu, and Dian Wang. "Gradual Optimization of University Course Scheduling Problem Using Genetic Algorithm and Dynamic Programming." *Algorithms* 18, no. 3 (2025): 158.