**CLASSROOM ALLOCATION TO SECTIONS**

Project submitted to the

SRM University – AP, Andhra Pradesh

for the partial fulfillment of the requirements to award the degree of

**Bachelor of Technology/Master of Technology**

In

**Computer Science and Engineering**

**School of Engineering and Sciences**

Submitted by

**Vijaya Vyshnavi Muvvala -AP21110010119**

**Abhiram Rallabandi -AP21110010120**

**Akhil Sai Kapa-AP21110010125**

**Neeli Meghana Nandigam – AP21110010127**

**Manoj Bhaskar Reddy-AP2111010129**

**A picture containing text

Description automatically generated**

Under the Guidance of

**Dr. Ashu Abdul**

**SRM University–AP**

**Neerukonda, Mangalagiri, Guntur**

**Andhra Pradesh – 522 240**

**[Nov, 2023]**

# Certificate

Date: 23-Nov-23

This is to certify that the work present in this Project entitled “**CLASSROOM ALLOCATION TO SECTIONS**” has been carried out by **Vijaya Vyshnavi Muvvala, Abhiram Rallabandi, Akhil Sai Kapa, Neeli Meghana Nandigam, Manoj Bhaskar Reddy.** Under my/our supervision. The work is genuine, original, and suitable for submission to the SRM University – AP for the award of Bachelor of Technology/Master of Technology in **School of Engineering and Sciences**.

**Supervisor**

(Signature)

Dr. Ashu Abdul

Assistant Professor,

Computer Science and Engineering

**Co-supervisor**

(Signature)

Prof. Niraj Upadhayaya

Head of the Department,

Computer Science and Engineering

**Acknowledgements**

The satisfaction that accompanies the successful completion of any task would be incomplete without introducing the people who made it possible and whose constant guidance and encouragement crowns all efforts with success.

We are extremely grateful and express our profound gratitude and indebtedness to our project guide, Dr. Ashu Abdul, Assistant Professor, Department of Computer Science & Engineering, SRM University, Andhra Pradesh, for his kind help and for giving us the necessary guidance and valuable suggestions in completing this project work.

**Vijaya Vyshnavi Muvvala -AP21110010119**

**Abhiram Rallabandi -AP21110010120**

**Akhil Sai Kapa-AP21110010125**

**Neeli Meghana Nandigam – AP21110010127**

**Manoj Bhaskar Reddy-AP2111010129**

# Table of Contents

[Certificate i](#_Toc105799088)

[Acknowledgements iii](#_Toc105799089)

[Table of Contents v](#_Toc105799090)

[Abstractvii](#_Toc105799091)

[List of Publications ix](#_Toc105799092)

[Statement of Contributions xi](#_Toc105799093)

[Abbreviations xv](#_Toc105799095)

[List of Tables xvii](#_Toc105799096)

[List of Figures xix](#_Toc105799097)

[List of Equations xxi](#_Toc105799098)

[1. Introduction 1](#_Toc105799099)

[1.1 Heading 2 1](#_Toc105799100)

[1.1.1 Heading 3 1](#_Toc105799101)

[2. Methodology 3](#_Toc105799102)

[2.1 Heading 2 3](#_Toc105799104)

[2.1.1 Heading 3 3](#_Toc105799105)

[3. Discussion 5](#_Toc105799106)

[4. Concluding Remarks 7](#_Toc105799107)

[5. Future Work 9](#_Toc105799108)

[References 11](#_Toc105799109)

**Abstract**

In any educational institution, the effective allocation of classrooms to different sections is crucial for maintaining an organized and conducive learning environment.

The main aim of our project is to develop a systematic solution using Constraint Satisfaction Problem (CSP) techniques to address the challenges associated with classroom allocation.

The primary objective is to create an automated system that efficiently assigns classrooms to various sections based on predefined constraints, ensuring fair and optimal utilization of resources. The code presented here is an automatic algorithm for the learning process, solving the class assignment problem within defined constraints The main function of the program is findPattern(), which provides the optimal pattern considering the number of classes, departments, daily times, and classes. The process involves defining the problem, starting with departments, dates, timelines, and classes. Using an iterative and recursive approach, the algorithm finds the appropriate allocation for each category, checking that it matches the existing order and the available classes The ordering function repeats at possibilities, classes, and departments until possible configurations are identified or all possibilities are exhausted. Once resolved, the system generates a scheduling matrix representing the classes identified in each block, day, and time frame. If a valid schedule is within the specified constraints, the job returns the schedule and schedule. If no solution is found after the obstacles are identified, it legitimately reflects the absence of a proper timeline. This scheduling framework illustrates the use of an iterative backward algorithm to solve scheduling problems, and provides a possible framework for academic institutions seeking automatic scheduling generation solutions

**Statement of Contributions**

Problem Definition Functionality: Developed the define\_problem feature to initialize and outline the problem area for the scheduling algorithm. This function units up the sections, days, time slots, and school rooms, forming the foundational framework for timetable generation.

Scheduling Algorithm Implementation: Designed and implemented the schedule\_class feature, which employs a recursive backtracking approach to clear up the class allocation trouble. This algorithm iterates through ability class placements inside the described problem area even as making sure agenda integrity and adhering to classroom availability.

Timetable Generation and Output: Integrated the scheduling good judgment inside the findpattern function to generate viable timetables primarily based at the furnished magnificence, phase, and slot counts. The characteristic orchestrates the scheduling procedure and constructs a readable timetable matrix representing scheduled instructions throughout sections, days, and time slots.

Error Handling and Reporting: Implemented capability within findpattern to hit upon cases wherein a viable timetable cannot be generated due to constraints. It correctly reviews the absence of a solution, making sure transparency in cases in which scheduling conflicts cannot be resolved.

This contribution encapsulates the development of a complete scheduling system, encompassing problem initialization, algorithmic good judgment, timetable generation, and mistakes handling, facilitating efficient and automatic timetable introduction for instructional institutions.

# 1.Introduction

In the dynamic landscape of educational institutions, efficient allocation of classrooms to various sections based on both day and time considerations is a critical undertaking. This project, titled "Classroom Allocation to Sections using Constraint Satisfaction Problem," delves into the realm of optimizing classroom usage through the application of advanced computational techniques.

The primary objective of this project is to devise a systematic and intelligent solution to address the complexities associated with scheduling classrooms for different sections over a five-day week, taking into account specific time slots. The methodology employed here centres around the utilization of Constraint Satisfaction Problem (CSP), a powerful computational approach that ensures the fulfilment of a set of constraints while maximizing the overall efficiency of classroom allocation.

The identified constraints serve as the cornerstone of this project, emphasizing the need for a uniform and consistent allocation of classrooms among various sections. The project acknowledges the importance of addressing conflicting scenarios, ensuring that the same classroom is not allocated to more than one section within the same time slot. This proactive approach not only mitigates scheduling conflicts but also strives to optimize the overall utilization of available resources.

Through the lens of Constraint Satisfaction Problem, this project endeavours to provide a robust, flexible, and intelligent solution to the intricate task of classroom allocation. By leveraging computational techniques, the aim is to enhance the efficiency of educational institutions, streamline the scheduling process, and ultimately contribute to a conducive learning environment for both students and faculty.

# Methodology

The methodology for the Classroom Allocation to Sections project using Constraint Satisfaction Problem (CSP) revolves around the systematic allocation of classrooms to various sections based on specified constraints. The code provided employs a backtracking algorithm to explore and test possible solutions iteratively. Below is an explanation of the key components and steps in the methodology:

1. **Problem Definition:**

The project defines the scheduling problem using the function define\_problem(). It establishes the initial schedule, sections, days, time slots, and classrooms, forming the foundation for subsequent scheduling operations.

1. **Constraint Checking:**

The function can\_schedule() is responsible for checking whether a given section can be scheduled in a particular classroom at a specified day and time. It verifies constraints such as classroom availability, absence of conflicts and equal utilization of all the classrooms by different sections.

Equal utilization of classrooms is satisfied my keeping a track of the classroom allocation count making sure that the equation

classroom\_counts[classroom] >= len(schedule) // sectionCount

is satisfied.

1. **Backtracking Algorithm:**

The core scheduling logic resides in the schedule\_class() function. This recursive backtracking algorithm iterates through sections, days, and time slots, attempting to assign classrooms while adhering to constraints. It shuffles the order of classroom assignment to explore different possibilities.

1. **Solution Identification:**

The findpattern() function orchestrates the entire process. It initializes the scheduling parameters and attempts to find a valid solution using the backtracking algorithm. If successful, it constructs a timetable and returns it along with the corresponding time slots.

1. **Output Handling:**

The results are presented in a human-readable format. If a solution is found, a timetable is printed, displaying the allocation of classrooms to sections for each day and time slot. If no solution is found, an appropriate message is displayed.

1. **Usage of Randomization:**

Randomization is introduced in the assignment of classrooms to sections to explore different scheduling possibilities. This approach helps in diversifying the search space and potentially finding more optimal solutions.

By combining these elements, the methodology strives to systematically explore and allocate classrooms to sections, meeting the specified constraints and providing an efficient scheduling solution using the Constraint Satisfaction Problem paradigm.

# Concluding Remarks

In conclusion, our endeavor to optimize classroom allocation through a genetic algorithm represents a significant step toward addressing the complexities inherent in educational scheduling. The project has successfully demonstrated the efficacy of computational methods in achieving efficient and feasible solutions, contributing to the enhancement of educational resource management.

* **Key Contributions:**

The genetic algorithm, with its foundation in natural selection principles, has proven instrumental in evolving diverse and effective classroom allocation schedules. The incorporation of entropy in fitness calculations has enhanced the algorithm's ability to generate balanced timetables, fostering an equitable distribution of classes.

* **Efficiency and Automation:**

By automating the traditionally labor-intensive task of classroom allocation, the project significantly improves efficiency and reduces the likelihood of errors. The genetic algorithm's adaptability and optimization capabilities empower educational institutions to navigate the complexities of scheduling with greater ease.

* **Insights Gained:**

The implementation details, including the Chromosome, Population, and Genetic Algorithm classes, have provided valuable insights into the mechanics of the genetic algorithm. Understanding the interplay between selection, crossover, and mutation has shed light on the algorithm's ability to iteratively refine solutions over multiple generations.

* **Impact on Educational Management:**

The significance of our project lies in its potential to revolutionize how educational institutions approach classroom allocation. By providing a tool that automates and optimizes this critical task, we contribute to the broader goal of fostering a more conducive and efficient learning environment.

* **Call to Action:**

As we conclude, the call to action is clear - the integration of computational solutions, such as the genetic algorithm presented here, into educational management practices can lead to substantial improvements. Further research, collaboration, and implementation efforts are warranted to fully realize the transformative potential of these methods.

# Future Work

Building upon the current project, there are several avenues for expansion and improvement to enhance its practicality and usability. The following outlines potential future works for the Classroom Allocation to Sections project:

1. **Limited Number of Scheduled Classes:**

One significant enhancement would involve introducing constraints on the maximum number of classes scheduled for each section and each day. This refinement reflects a more realistic scenario where sections may have restrictions on the number of classes, they can attend in a given time frame.

1. **Dynamic Constraints and Preferences:**

Implementing a more sophisticated constraint system that considers dynamic factors such as instructor preferences, room capacities, and other real-world constraints. This would contribute to a more flexible and adaptable scheduling system that aligns with the diverse requirements of educational institutions.

1. **Optimization Algorithms:**

Exploring advanced optimization algorithms to improve the efficiency and speed of the scheduling process. Techniques such as genetic algorithms, simulated annealing, or other metaheuristic approaches could be integrated to find optimal or near-optimal solutions within shorter time frames.

1. **User Interface Enhancements:**

Enhancing the user interface of the web application to provide more interactive features, allowing users to input constraints, preferences, and other parameters directly. A user-friendly interface could empower administrators and planners to customize the scheduling process according to their institution's specific needs.

1. **Scalability and Performance Optimization:**

Addressing scalability concerns and optimizing the performance of the scheduling algorithm to handle larger datasets and more complex scenarios. This would make the system applicable to a broader range of educational institutions with varying sizes and structures.