Project Report

Genetic Algorithm (GA) based timetable scheduling system for a university or college. The main components of the project are spread across three Python files: constants.py, timetable.py, and ga.py.

1. constants.py: This file contains all the constant values used throughout the project. It includes the days of the week, timeslots for classes, room capacities, sections, course types, and professor details. Each professor has a list of courses they can teach, a maximum number of courses they can teach, and a count of the courses they are currently teaching.
2. timetable.py: This file defines the Timetable and DayTimetable classes.
   The Timetable class represents a weekly timetable, with each day having its own DayTimetable. The DayTimetable class represents a single day's timetable, with different timeslots and rooms. The generate_timetable method is used to generate a timetable for a day. The print method is used to print the timetable in a tabular format.
3. ga.py: This file contains the implementation of the Genetic Algorithm.
   The objective function calculates the fitness score of a timetable. The lower the score, the better the timetable. The GA iteratively generates new populations of timetables, selects the best ones based on their fitness scores, and uses them to create the next generation.

Objective/Fitness function:

The objective function returns the negative of the sum of all the conflicts/clashes.

Constraints used:

- Make sure all Courses and Sections have been exhausted.
- Room can accommodate class strength.
- Professor should not be assigned two courses in the same timeslot.
- Same section should not have two classes in different rooms at the same time.
- Professors cannot teach more than 3 courses.
- Section can only have upto 5 classes a week.
- 2 Lecture per week for a course.
- Same sections cannot have different professors for the same course.
- Labs take consecutive slots

Mutation function:

The mutation function used in this project works in the following way:

1. The mutation process begins with a random choice. If the result is True (which has an 80% chance of happening), the mutation process continues.
2. The function selects an entity from the timetable of a given day, timeslot, and room.

3. If the selected entity is a 'theory' class, it attempts to swap it with another 'theory' class from a different day, timeslot, and room. If the second location does not have a 'theory' class, it moves the 'theory' class to the second location and leaves the first location empty.
4. If the selected entity is None (i.e., the slot is empty), it attempts to move a 'theory' class from a different day, timeslot, and room to the empty slot.
5. If the selected entity is a 'lab' class, it attempts to swap it with another 'lab' class from a different day, timeslot, and room. However, this swap only occurs if both labs are in the same slot ('S1' or 'S2') and there is a valid next (for 'S1') or previous (for 'S2') timeslot. The swap also includes the next or previous timeslot to accommodate the two timeslots that a 'lab' class typically occupies.

Crossover Function:

1. Crossover function chooses a random point in the timetable (not at the ends). This is the crossover point.
2. It copies the days from the first parent (p1) into the offspring's timetable up to the crossover point. This means that the start of the offspring's timetable is the same as the start of the first parent's timetable.
3. It then copies the days from the second parent (p2) into the offspring's timetable from the crossover point onwards. This means that the end of the offspring's timetable is the same as the end of the second parent's timetable.

The project uses a Genetic Algorithm to find an optimal solution to the timetable scheduling problem. The GA starts with a population of random timetables, evaluates their fitness using the objective function, and then uses selection, crossover, and mutation to generate new populations. This process is repeated until a satisfactory timetable is found or a maximum number of generations is reached.