Muneel Haider
I21-0640
Section G

AI Project - Timetable Scheduling Project Report

Introduction:

The timetable scheduling project aims to efficiently schedule university courses for each semester, ensuring minimal clashes between sections, professors, and rooms. The project implements a genetic algorithm approach to solve this complex optimization problem.

Steps Involved:

1. **Problem Understanding**:
    - The project begins with a thorough understanding of the problem statement and constraints provided in the project description.
    - This includes identifying hard and soft constraints, as well as other relevant details.
2. **Data Representation:**
    - Courses, professors, rooms, and sections are represented using appropriate classes in Python.
    - Each entity is modeled with relevant attributes and methods to facilitate scheduling.
3. **Initialization**:
    - The population of timetables is initialized using binary encoding, where each chromosome represents a possible timetable solution.
    - The initial population is generated randomly, adhering to constraints such as room availability and professor workload.
4. **Fitness Evaluation**:
    - A fitness function is defined to evaluate the quality of each timetable solution.
    - The fitness function considers the number of conflicts or clashes present in the timetable, with penalties for violating hard constraints.

5. **Selection**:
   - Tournament selection is used to select parent chromosomes for crossover.
   - This method ensures that chromosomes with higher fitness values have a higher chance of being selected for reproduction.
6. **Crossover**:
   - Two-point crossover is implemented to create offspring from selected parent chromosomes.
   - This process combines genetic information from the parents to produce new timetable solutions.
7. **Mutation**:
   - The mutation is applied to introduce diversity in the population and prevent premature convergence.
   - Random mutations alter certain genes in the chromosome, leading to potentially better solutions.
8. **Genetic Algorithm**:
   - The genetic algorithm iterates through multiple generations, gradually improving the population of timetables.
   - Selection, crossover, and mutation operations are applied iteratively to evolve better timetable solutions.

Results:

1. The best timetable solution obtained after a specified number of generations is analyzed. The solution is evaluated based on the fitness score and its adherence to the given constraints.
2. The results are saved in separate .csv files.

Muneel Haider
I21-0640
Section G

Conclusion:

The timetable scheduling project demonstrates the application of genetic algorithms to solve complex optimization problems. By intelligently encoding solutions, evaluating fitness, and evolving better solutions over multiple generations, the algorithm efficiently schedules university courses, ensuring smooth operations and minimal conflicts.

Output Screenshots:

Files > friday.csv

Time Slot	Class Details
08:30-09:50	Algorithms - Section S3, Room R1, Prof Faisal
10:00-11:20	No Class
11:30-12:50	Algorithms - Section S3, Room R1, Prof Faisal
13:00-14:20	No Class
14:30-15:50	Internet of Things Lab - Section S4, Room R9, Prof Kamran
16:00-17:20	Computer Vision Lab - Section S3, Room R10, Prof Faisal --|-- Internet of Things Lab - Section S1, Room R1, Prof Usman

Files > thursday.csv

Time Slot	Class Details
08:30-09:50	Computer Vision Lab - Section S4, Room R12, Prof Nadia --|-- Algorithms - Section S1, Room R7, Prof Hassan
10:00-11:20	Algorithms - Section S4, Room R1, Prof Ayesha
11:30-12:50	No Class
13:00-14:20	Game Development Lab - Section S3, Room R7, Prof Hassan
14:30-15:50	Internet of Things Lab - Section S3, Room R11, Prof Sana
16:00-17:20	No Class