Artificial Intelligence – Spring 2025
Assignment 3

Exams Schedule Generator Using Genetic Algorithm

Deadline: 27th March,2025

General Instructions:

- This is an individual assignment.
- The deadline will not be extended under any circumstances.
- **All the components of the algorithm being used (GA) should be mentioned in the PDF report along with the rationale of choosing that particular technique.**
- Evaluation will be done on the basis of a demo. Anyone failing to appear for the demo or unable to give up to the mark demonstration will be awarded zero.
- Four Excel files have been provided along with this assignment statement containing the dataset to be used in this assignment. (minor changes can be made)

[CLO: 4] General understanding of major concepts and approaches in knowledge representation, planning, learning, robotics and other AI areas.

[CLO: 5] Developing programming skills for AI applications.

Problem Statement:

Your task is to find generic solution that will facilitate generating schedule for university using "**Genetic Algorithm**".

- You have to read the data from the given files.
- You have to write code from scratch.
- Make sure your notebook is well documented
- You cannot use any built-in library for the implementation of Genetic Algorithm except Pandas and NumPy.
- You can use any kind of crossover discussed in class with proper reasoning.
- You can choose any rate of mutation (which can be justifiable)

- You have to use roulette wheel selection for selecting potentially useful solutions for recombination (Chromosomes).

The success of solution is estimated on fulfillment of given constraints and criteria. Results of testing the algorithm show that all essential requirements are satisfied, while additional criteria are optimized to a certain extent. **You have to submit. ipynb with a detailed report of your implementation(pdf).**

Fitness Function Description

The fitness function evaluates the quality of a generated schedule by ensuring that all necessary conditions are met while optimizing additional preferences. It is calculated based on the following factors:

Essential Requirements (Must be Satisfied for a Valid Schedule)

- Every course must have an exam scheduled.
- No student should be assigned to overlapping exams. Each student is enrolled in at least three courses and can take only one exam at a time.
- Exams are scheduled only on weekdays (Monday–Friday).
- Exam timings must be between 9 AM and 5 PM.
- Each exam must have an invigilating teacher, ensuring no teacher is assigned to multiple exams simultaneously.
- A teacher should not be assigned consecutive exam invigilation duties.

Optimization Criteria (Affect Fitness Score Positively if Satisfied)

- A common break on Friday from 1–2 PM for all students and teachers.
- Students should not have back-to-back exams whenever possible.
- If a student is enrolled in both a **Management (MG)** and **Computer Science (CS)** course, the MG course exam should preferably be scheduled before the CS course exam.
- A two-hour break in the week, ensuring that at least half of the faculty is free in one slot and the rest in another for faculty meetings.

Fitness Calculation

- A **penalty** is applied for any schedule violating essential constraints, leading to an **invalid solution** if left unresolved.
- A **reward score** is added for each optimization criterion satisfied, improving the overall fitness of the solution.

- The goal is to **maximize the fitness score** while ensuring no violations of essential constraints.

The algorithm iteratively evolves schedules, prioritizing feasible solutions with higher fitness scores. The best solution will be one that meets all essential constraints while optimizing as many additional criteria as possible.

Input & Output

Input data for each exam are *teachers' names, students 'name, exam duration*, *courses (course codes)*, and list of *allowed classrooms* (e.g. C301 to C310)

Output data are classroom and starting time for each exam along with course code and invigilating teacher. Time is determined by day (Monday to Friday) and start hour of the exam.

- Show all the individuals of each generation or at least the top 2-3 fittest individuals of each generation.
- Final Output will be a chromosome which satisfies all hard constraints and soft constraints at least three. (as much as you can)
- You have to display a list of all hard and soft constraints which are fulfilled in the output.
- Don't forget to **show fitness** values at each iteration.
- The output should be in a proper format preferably a Table or a Chart, color coded table would be a plus.

Evaluation Criteria

Requirement	Marks
Correct Implementation of GA	40
Fulfillment of hard constraints	10*6 = 60
Fulfillment of soft constraints	5*4 = 20
PDF Report	20
Properly Formatted Output	20
.ipynb file containing code in python	20
Demo/ Viva	20

Submission Instructions

- Assignment must be submitted on the google classroom.
- Submission other than google classroom won't not be accepted.
- You are required to submit a zip file named as **22i-7777_Name_BCY-A.zip** containing your notebook and document named as:
  Asg03- <22I- XXXX>.ipynb and **Asg03- <22I-XXXX>.pdf**.
- Your report must include a sample dataset along with its output upon running your program – the harshest dataset that you have tested your program on.

Your notebook must contain all the subsets of the datasets/ testcases that you have tested your code on.