National University
of computer and emerging sciences

Assignment 3

AI2002-Artificial Intelligence

Name: Mirza Humayun Masood(i221749)

Section : CY-A

Submitted to: Sir Shoaib Saleem Khattak

Department of Cyber Security BS(CY)

FAST-NUCES Islamabad

1. Introduction

Scheduling university exams is a complex task that requires balancing constraints such as student course enrollments, teacher availability, and time slots. This project utilizes a **Genetic Algorithm (GA)** to automate the scheduling process while minimizing conflicts and optimizing efficiency.

2. Problem Statement

The objective is to generate an exam schedule that satisfies essential constraints while optimizing additional criteria. The input includes **course enrollments, student-course associations, teachers, and available time slots**. The final output must allocate **courses, teachers, and exam times** while ensuring conflict-free scheduling.

3. Genetic Algorithm Overview

A Genetic Algorithm is an evolutionary optimization technique inspired by natural selection. It operates through the following steps:

- **Initialization:** Generate a population of possible schedules.
- **Selection:** Choose schedules with better fitness scores.
- **Crossover:** Combine schedules to create new offspring.
- **Mutation:** Introduce random changes to maintain diversity.
- **Evaluation:** Assess fitness based on constraints and optimization goals.

4. Constraints and Fitness Function

4.1 Hard Constraints (Must be Satisfied)

1. No student should have overlapping exams.
2. Exams must be scheduled on weekdays (Monday–Friday).
3. Exam timings must be between **9 AM and 5 PM**.

4. Each exam must have **one invigilating teacher**.
5. A teacher cannot invigilate multiple exams at the same time.
6. Teachers should not be assigned consecutive exam invigilators.

4.2 Soft Constraints (Optimization Goals)

1. A **common break on Friday from 1–2 PM**.
2. Students should avoid **back-to-back exams**.
3. **Management (MG) exams should be scheduled before Computer Science (CS) exams** if both are enrolled.
4. Ensure **faculty meeting slots** by keeping at least half of the faculty free at a time.

5. Implementation Details

5.1 Data Processing

- Read input files: `courses.csv`, `studentCourse.csv`, `studentNames.csv`, `teachers.csv`.
- Extract course lists, student-course mappings, and teacher lists.

5.2 Population Initialization

Each schedule is initialized by:

- Randomly assigning **exam times and days** from Monday to Friday.
- Ensuring no conflicts for students and teachers.
- Using a **randomized approach** to distribute teachers.

5.3 Selection Mechanisms

One selection strategies were implemented:

1. **Tournament Selection** – Randomly select a group of schedules and pick the best one.

5.4 Crossover and Mutation

- **Crossover:** A single-point crossover is used to mix parent schedules and produce offspring.
- **Mutation:**
    - **Randomly change exam times or teachers**.
    - Ensure new assignments do not create conflicts.

6. Fitness Function Calculation

The fitness function assigns penalties for violating hard constraints and rewards for meeting soft constraints:

- **Penalty:** Overlapping exams (+50), invalid time slots (+20), consecutive teacher invigilation (+20).
- **Rewards:** Friday break (+50), avoiding back-to-back exams (+20), MG before CS exams (+30).

7. Results and Evaluation

- The algorithm was tested over **500 generations** with a population size of **50**.
- **Fitness values improved over generations**, reducing conflicts and optimizing soft constraints.
- The final schedule satisfies all **hard constraints** and meets at least **3 soft constraints**.

8. Conclusion

This project successfully implemented a **Genetic Algorithm-based exam scheduler**. The algorithm **evolved schedules iteratively**, improving efficiency while ensuring fair exam distribution. Future improvements could include **additional constraints, classroom allocations, and hybrid optimization techniques**.