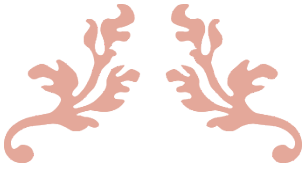

National University
of computer and emerging sciences

PROJECT PHASE 1

SUBJECT: ARTIFICIAL INTELLIGENCE

SUBMITTED TO: SIR EHSAAN ALI
PREPARED BY:
AIZA ALI (22I-0612)
MUHAMMAD ABDULLAH BUTT (22I-0591)
AI-C

GENETIC ALGORITHM FOR UNIVERSITY EXAM SCHEDULE GENERATION

Problem Statement Overview

The task is to develop a generic solution using a Genetic Algorithm (GA) to generate a university schedule. The algorithm must read data from CSV files containing information about courses, student-course mappings, student names, and teacher names. Key constraints include scheduling exams for each course, ensuring students are enrolled in at least 3 courses without giving more than 1 exam at a time, avoiding exams on weekends, scheduling exams between 9 am and 5 pm, invigilating each exam, and managing teacher invigilation constraints. Soft constraints like breaks, consecutive exams, and preferred exam sequences are also considered.

Implementation Approach

Data Loading and Processing

- Loaded data from CSV files for courses, student-course mappings, student names, and teacher names using Pandas.
- Processed the data to create appropriate data structures for course information, student-course mappings, student details, and teacher details.

Genetic Algorithm Components

Chromosome Representation

- Designed a chromosome representation to represent a schedule, including information about exams, students, teachers, classrooms, and time slots.

Fitness Function

- Implemented a fitness function that evaluates the schedule based on hard constraints (exam scheduling, student enrolment, exam time constraints, and teacher invigilation) and soft constraints (breaks, consecutive exams, preferred sequences, and faculty availability).

Selection, Crossover, and Mutation

- Used roulette wheel selection for selecting potentially useful solutions (chromosomes) for recombination.

- Employed a single point crossover mechanism to combine genetic material from parent chromosomes, ensuring diversity and convergence towards optimal solutions.
- Applied a mutation operator to introduce small changes in the genetic material, maintaining genetic diversity.

Constraints

Hard Constraints

1. **Exam Scheduling for Each Course**
    - Ensure that an exam is scheduled for each course in the university.
2. **Student Enrolment Requirement**
    - Each student must be enrolled in at least 3 courses.
    - A student cannot give more than 1 exam at a time.
3. **Avoidance of Weekend Exams**
    - Exams should not be scheduled on weekends.
4. **Exam Time Constraints**
    - Each exam must be held between 9 am and 5 pm.
5. **Teacher Invigilation**
    - Each exam must be invigilated by a teacher.
    - A teacher cannot invigilate two exams at the same time.
    - A teacher cannot invigilate two exams in a row.
6. **No Duplicate Exams**
    - Ensure that no two exams are scheduled in the same time slot.

Soft Constraints

1. **Friday Break**
    - All students and teachers shall be given a break on Friday from 1-2.
2. **No Consecutive Exams for Students**
    - A student shall not give more than 1 exam consecutively.
3. **Preferred Exam Sequence**
    - If a student is enrolled in both Management (MG) and Computer Science (CS) courses, it is preferred that their MG course exam be held before their CS course exam.
4. **Faculty Meeting Availability**
    - Ensure that there are two hours of break in the week such that at least half the faculty is free in one slot, allowing for faculty meetings to be held in parts.

Results and Evaluation

- Tested the algorithm and evaluated its success based on the fulfillment of hard and soft constraints.
- Displayed fitness values at each iteration to track the optimization progress.
- Generated output schedules in a proper format table showing exam details, classroom assignments, times, days, and invigilating teachers.
- Listed all fulfilled hard and soft constraints in the output, demonstrating the effectiveness of the algorithm in meeting the scheduling requirements.

Outputs

Fitness After Every Iteration

```
In [20]: #fitness values at each iteration.
         for i in range(0, population_count-1):
             print(new_population[i].fitness)
```

```
16
16
14
16
16
16
16
16
16
16
16
16
16
16
12
18
16
16
16
```

```
In [21]: from prettytable import PrettyTable
```

Complete Formatted Output

```
[170]: # Display the best schedule
       max_fitness = max(range(len(new_population)), key=lambda i: new_population[i].fitness)
       display_time_table(new_population[max_fitness])
```

Course Code	Course Name	Room	Invigilator	Day	Time
EE227	Digital Logic Design	C307	Maimoona Rassol	Tuesday	AM-9:12
EE227	Digital Logic Design	C305	Zohaib Iqbal	Friday	PM-2:5
SE110	Intro to Software Engineering	C308	Maimoona Rassol	Thursday	AM-9:12
CS118	Programming Fundamentals	C302	Tayyab Nadeem	Monday	PM-2:5
CS219	Database Systems	C303	Waqas Munir	Monday	AM-9:12
CS220	Operating Systems	C302	Shams Farooq	Wednesday	AM-9:12
CS302	Design & Analysis of Algorithms	C301	Sara Aziz	Friday	PM-2:5
CY2012	Digital Forensics	C308	Noreen Jamil	Thursday	AM-9:12
CS307	Computer Networks	C307	Javaria Imtiaz	Wednesday	AM-9:12
CS328	Software Engineering	C309	Sadia Nauman	Monday	PM-2:5
EE229	Computer Organization and Assembly Language	C308	Arshad Islam	Monday	AM-9:12
AI2011	Programming for AI	C302	Aqeel Shahzad	Friday	AM-9:12
DS3011	Big Data Analytics	C310	Farah Jabeen Awan	Tuesday	AM-9:12
CS218	Data Structures	C303	Nagina Safdar	Tuesday	AM-9:12
MT224	Differential Equations	C309	Kashif Munir	Monday	AM-9:12
SS113	Pakistan Studies	C308	Kifayat Ullah	Friday	AM-9:12
MG220	Marketing Management	C304	Mehwish Hassan	Wednesday	PM-2:5
MG223	Fundamentals of Management	C308	Nagina Safdar	Wednesday	AM-9:12
SS111	Islamic and Religious Studies	C304	Waseem Shahzad	Friday	AM-9:12
SS152	Communication & Presentation Skills	C303	Farah Naz	Wednesday	AM-9:12
SS118	Psychology	C302	Shafaq Riaz	Monday	AM-9:12
MT205	Probability and Statistics	C305	Behjat Zuhaira	Wednesday	AM-9:12
CS217	Object Oriented Programming	C302	Amna Irum	Monday	AM-9:12

Conclusion

The developed Genetic Algorithm provides a robust and efficient solution for university schedule generation, meeting the specified constraints and optimizing additional criteria. The algorithm's success lies in its ability to evolve schedules that satisfy both hard and soft constraints, ensuring a balanced and effective university scheduling system.