EXAM SCHEDULE GENERATOR USING GENETIC ALGORITHM

AI-Project

Ahmad Islam

Table of Content:

Data Reading and Preprocessing:

In the initial phase of the project, data loading and preprocessing play a pivotal role in setting the groundwork for subsequent algorithmic operations. Leveraging the robust capabilities of Python libraries such as Pandas and NumPy, the provided code snippet efficiently reads and organizes essential data from CSV files. Through the pd.read_csv() function, data pertaining to university courses, teachers, student enrollment, and student names are extracted into distinct Pandas DataFrames. This systematic approach enables structured access to critical information, including course codes, durations, teacher names, and student enrollments. Furthermore, the extracted data undergoes preprocessing steps to facilitate subsequent algorithmic operations seamlessly. By converting relevant columns into lists, essential data elements such as course codes and teacher names are readily accessible for algorithmic processing. This meticulous data handling ensures that the subsequent stages of the project, including algorithm implementation and schedule generation, are built upon a solid foundation of accurately captured and organized data.

Screenshots:

Courses Data:

	Course Code	Course Name
0	CS217	Object Oriented Programming
1	EE227	Digital Logic Design
2	CS211	Discrete Structures
3	SE110	Intro to Software Engineering
4	CS118	Programming Fundamentals
5	CS219	Database Systems
6	CS220	Operating Systems
7	CS302	Design & Analysis of Algorithms
8	CY2012	Digital Forensics
9	CS307	Computer Networks
10	CS328	Software Engineering
11	EE229	Computer Organization and Assembly Language
12	AI2011	Programming for AI
13	DS3011	Big Data Analytics
14	CS328	Software Engineering

Exam Schedule Generation Function

The generate_exam_schedule() function serves as a pivotal component in the scheduling process, orchestrating the creation of a comprehensive exam timetable within the specified timeframe. This function efficiently constructs the schedule while adhering to predefined constraints and time slot durations. The process begins by initializing the exam schedule list, which will store the generated schedule entries. Utilizing a while loop, the function iterates over each date within the designated range, ensuring that only weekdays (Monday to Friday) are considered for scheduling. For Fridays, distinct time slots are allocated, accommodating the specified friday_time_slot_duration. Meanwhile, for other weekdays, standard time slots of duration time_slot_duration are assigned, spanning from 09:00 to 17:00. Each generated schedule entry includes details such as the date, day, start time, and end time, encapsulated within a tuple and appended to the exam_schedule list. Finally, the function returns the compiled exam schedule, ready for further processing and utilization. This function encapsulates the foundational step of generating an exam timetable, providing a structured and systematic approach to schedule creation. By incorporating parameters such as start and end dates, as well as time slot durations, the function offers flexibility in tailoring the schedule to meet specific requirements. Additionally, the utilization of Pandas Timestamp objects ensures accuracy and precision in date and time manipulations, facilitating seamless integration with subsequent scheduling algorithms and processes. The generated exam schedule is presented in a Pandas DataFrame (exam_schedule_df), offering a structured representation of the timetable for visualization and analysis.

Screenshots:

Generated Exam Schedule:

	Date	Day	Start Time	End Time
0	03-05-2024	Friday	09:00	12:00
1	03-05-2024	Friday	14:00	17:00
2	06-05-2024	Monday	09:00	12:00
3	06-05-2024	Monday	12:00	15:00
4	07-05-2024	Tuesday	09:00	12:00
5	07-05-2024	Tuesday	12:00	15:00
6	08-05-2024	Wednesday	09:00	12:00
7	08-05-2024	Wednesday	12:00	15:00
8	09-05-2024	Thursday	09:00	12:00
9	09-05-2024	Thursday	12:00	15:00
10	10-05-2024	Friday	09:00	12:00
11	10-05-2024	Friday	14:00	17:00

Student Course Data Processing:

The section of code provided demonstrates the process of parsing and organizing student-course enrollment data extracted from the "studentCourse.csv" file. Employing a ‘defaultdict’ structure, the code efficiently aggregates course codes based on individual student names. Within the context of the with statement, the CSV file is opened in a newline-separated mode, and each row is interpreted as a dictionary using the csv. DictReader. As the loop iterates over each row, the student's name and corresponding course code are extracted. Utilizing the ‘defaultdict’, the course code is appended to the list associated with the respective student's name. This aggregation ensures that all courses enrolled by a particular student are appropriately grouped together. Finally, the resulting defaultdict is converted into a standard dictionary, providing a structured representation of student-course relationships. This processed data serves as a fundamental input for subsequent scheduling tasks, facilitating efficient access and utilization of student enrollment information in generating optimized exam schedules tailored to individual student needs and course requirements.

Screenshots:


```
{'Sam D Edwards': ['AI2011', 'SS152', 'EE229'],
 'Sheila Hughton': ['DS3011'],
 'Yasmin Ahmed': ['SE110', 'CS307', 'SS113'],
 'Sarah N Md Sallehuddin Khan': ['EE229', 'CS328', 'CS307', 'MG223'],
 'Sarah Nolasco': ['AI2011', 'CS328'],
 'Jenna Riley': ['EE229', 'CS211', 'SE110'],
 'Usman Rafiq': ['CS307', 'CS218', 'CS118'],
 'Reem N Hassan': ['MG220', 'SS118'],
 'Sarah Hinett': ['CS328', 'CS217'],
 'Kamal Anwar': ['EE229', 'MG220', 'SS152'],
 'Mika Tatsumoto': ['CS219', 'CS217'],
 'Muhammad Ijaz-Ul-Haq': ['AI2011'],
 'Abdul Gafur': ['SS118', 'SE110'],
 'Ana Vukojevic': ['CS307', 'AI2011', 'SS152'],
 'Arooba Zahoor': ['CS302', 'SS113'],
 'Ahmad F Yang Abd Talib': ['MG223'],
 'Natasha Leeson': ['CS328', 'SS113', 'CS307', 'CS211'],
 'Ramesh R Singh': ['MG220'],
 'Sara Zamberlan': ['CS211', 'CS217'],
 'Adam N Starling': ['CS217', 'EE227', 'CS211'],
 'Maria M Ponce Carpio': ['EE229', 'SS113', 'EE227'],
 'Iram Matloob': ['MT224', 'CS220', 'MG220'],
 'Sarah J Roberts': ['SS152', 'CS307', 'MG223'],
 'Maria Lytras': ['CS307'],
 'Mohammad Abir': ['CS218', 'MT205'],
```

Population Initialization and Schedule Generation:

In this section of the code, the population is initialized and exam schedules are generated for the Genetic Algorithm (GA). The population_size variable defines the number of schedules to be generated. Each schedule represents a potential solution to the exam scheduling problem. The code iterates over the population size and initializes an empty list called population to store the generated schedules. For each individual schedule, eligible students for each course are filtered using the assign_rooms_to_students() function. Then, the check_clash() function ensures there are no scheduling conflicts between students who have already been assigned to exam sessions and those eligible for the current course. Once conflicts are resolved, exam sessions are populated for each course. Rooms and invigilators are randomly assigned from available options, ensuring no repetitions within the same schedule. The resulting schedule entries include details such as date, day, start time, end time, course code, invigilator, assigned room, total number of students, and student names. Finally, each schedule is appended to the population list, representing one individual in the population. The code then displays the generated schedule for the first individual in the population using print(population[0]), providing an overview of the structure of the generated schedules.

Screenshots:

```
[['03-05-2024', 'Friday', ('09:00', '12:00'), 'CS211', 'Farah Naz', 307, 8, ['Jenna Riley', 'Natasha Leeson', 'Adam N Starling', 'Nabila Altaf',
```

Schedule: 1

	Date	Day	Time Slot	Code	Invigilator	Room No.	Registered Students	Students
0	03-05-2024	Friday	('09:00', '12:00')	CS211	Farah Naz	307	8	Jenna Riley, Natasha Leeson, Adam N Starlir
1	03-05-2024	Friday	('09:00', '12:00')	CS219	Zohaib Iqbal	306	4	Anja Brezovnik, Yas Khaled, Zahra Faraji Ra
2	03-05-2024	Friday	('14:00', '17:00')	EE229	Arshad Islam	306	8	Sam D Edwards, Sarah N Md Sallehuddin Khan,
3	03-05-2024	Friday	('14:00', '17:00')	CS218	Sumera Abbas	304	2	Usman Rafiq, Syed M Abbas
4	06-05-2024	Monday	('09:00', '12:00')	CS220	Asif Naeem	310	3	Iram Matloob, Maria A Grenfell, Farzana Yam
5	06-05-2024	Monday	('09:00', '12:00')	CS217	Gul e Aisha	307	3	Adam N Starling, Sarah Armstrong, Dalia Rem
6	06-05-2024	Monday	('12:00', '15:00')	SE110	Shafaq Riaz	306	4	Yasmin Ahmed, Jenna Riley, Mohammed I Al Ar
7	06-05-2024	Monday	('12:00', '15:00')	CS328	Nagina Safdar	310	10	Sarah N Md Sallehuddin Khan, Natasha Leeson
8	07-05-2024	Tuesday	('09:00', '12:00')	MT205	Tayyaba Zainab	307	4	Maria A Grenfell, Nosheen Maqbool, Adam Sel
9	07-05-2024	Tuesday	('09:00', '12:00')	CS307	Maimoona Rassol	301	6	Yasmin Ahmed, Sarah N Md Sallehuddin Khan,
10	07-05-2024	Tuesday	('12:00', '15:00')	SS111	Shehreyar Rashid	302	4	Sarah Armstrong, Sarah Austin, Nosheen Maqb
...								
22	10-05-2024	Friday	('14:00', '17:00')	CS307	Javaria Imtiaz	310	6	Yasmin Ahmed, Sarah N Md Sallehuddin Khan,

Fitness Calculation:

The calculate_fitness() function assesses the quality of a given exam schedule based on predefined constraints. This function evaluates various aspects of the schedule, assigning penalties for violations and bonuses for desirable attributes. Constraints such as avoiding Friday afternoon exams, minimizing common students between consecutive exams, and ensuring a preferred sequence between

Management (MG) and Computer Science (CS) exams are enforced through the deduction of penalty points. Additionally, the function rewards schedules that facilitate faculty free time for meetings, encouraging a balanced distribution of teaching responsibilities. By summing penalties and bonuses, the fitness score provides a quantitative measure of the schedule's adherence to constraints and optimization of additional criteria, guiding the Genetic Algorithm towards viable solutions for university exam scheduling.

Screenshots:

Best Schedule:

	Date	Day	Time Slot	Code	Invigilator	Room No.	Registered Students	Students
0	03-05-2024	Friday	('09:00', '12:00')	MG223	Ejaz Ahmed	304	5	['Sarah N Md Sallehuddin Khan', 'Sarah
1	03-05-2024	Friday	('09:00', '12:00')	EE227	Ameen Chilwan	308	3	['Adam N Starling', 'Maria M Ponce Carp
2	03-05-2024	Friday	('14:00', '17:00')	MT205	Mehreen Alam	307	4	['Maria A Grenfell', 'Nosheen Maqbool',
3	03-05-2024	Friday	('14:00', '17:00')	SS113	Shehreyar Rashid	309	5	['Yasmin Ahmed', 'Natasha Leeson', 'Mar
4	06-05-2024	Monday	('09:00', '12:00')	SS111	Usman Rashid	303	4	['Sarah Armstrong', 'Sarah Austin', 'No
5	06-05-2024	Monday	('09:00', '12:00')	SS118	Gul e Aisha	309	1	['Sarah Hanley']
6	06-05-2024	Monday	('12:00', '15:00')	MG220	Sumera Abbas	308	6	['Kamal Anwar', 'Iram Matloob', 'Sarah
7	06-05-2024	Monday	('12:00', '15:00')	CS211	Rohail Gulbaz	308	8	['Jenna Riley', 'Natasha Leeson', 'Adam
8	07-05-2024	Tuesday	('09:00', '12:00')	CS328	Waqas Munir	301	10	['Sarah N Md Sallehuddin Khan', 'Natash
9	07-05-2024	Tuesday	('09:00', '12:00')	SS152	Ayesha Bano	308	7	['Sam D Edwards', 'Kamal Anwar', 'Ana V
10	07-05-2024	Tuesday	('12:00', '15:00')	CS217	Sajid Khan	302	3	['Adam N Starling', 'Sarah Armstrong',
11	07-05-2024	Tuesday	('12:00', '15:00')	CS328	Shehreyar Rashid	308	10	['Sarah N Md Sallehuddin Khan', 'Natash
12	08-05-2024	Wednesday	('09:00', '12:00')	CS218	Khadija Farooq	306	2	['Usman Rafiq', 'Syed M Abbas']