Artificial Intelligence Project

Sophia Syed 18i-0835

Taimour Khan 18i-0651

Libraries used

For this project we used the Pandas, Random and os libraries

Data Collection

All data was read from the excel files into python lists but we were having trouble later on with such a vast data set, so we used a smaller dataset and decided to hardcode it instead.
The data was stored in a python class named Data and after the initialization, we created getters for the extracted data for later use.

Exam Class

The Exam class stored all the information needed for one exam. That included all the extracted data, course information, the time each exam would be held at, its allocated room number and the allocated invigilator, which was then proceeded by getters and setters for the said variables. We used setters for the room and the time variable because these variables were imported from the Data class described above. Best hao

Schedule Class

After implementing the Exam class, we made an instance of the Data class and stored it in a variable called “d” (d = Data())
Then we made the Schedule class and its purpose was to make an array of all randomly generated exams and store it in a list called “exam”.
The variables we used in the Schedule class were data (which stored the data), an array called exam, numOfConflicts, fitness, examNum and a boolean variable called isFitnessChanged.
We also made getters for each of the variables and an initialise function in which we looped over the length of all courses and assigned a new instance of Exam(Exam Class). We set the time, teacher and the room number randomly so that every exam instance would be assigned a random,time and invigilator from the given lists in the Data class.
Lastly we printed out each detail using the print function.

Population Class

To make the population needed for the genetic algorithm, we made the Population class which had a list of schedules. We will be passing the size as an argument which will make the population of that size.
We then loop over the size and initialise an instance of the Schedule class. That will fill the population class with as many schedules as the size we pass.
Lastly, we print the details of all the schedules we made by using the getters and the print.

Genetic Algorithm

In this algorithm we've implemented the crossover_population function which applies crossover to a whole population. This is done by the help of tournament_selection function.
The function tournament_selection returns a sorted list containing three schedules, with the fittest at the start.
In the crossover_population function, two schedule objects are made, each of them containing the starting value of the sorted list returned by the tournament_selection function, which is the fittest schedule amongst the three selected. These two schedules are passed in another function named crossover_schedule. The crossover_schedule function uses probability to decide which of the two schedules is to be returned.
This process returns the fittest schedule from the population to which crossover had been applied.