

**NAME:** Rayed Muhammad Saeed

**ROLL NO:** 20i-1822

**DEGREE:**  BSCS

**SECTION:** F

**SUBJECT:** Artificial Intelligence

**SUBMITTED TO:**  Maam Labiba

**SUBJECT CODE:** AI-2002

**ASSIGNMENT NO:** 2

**DATE:** 25th March 2023

**Report for Assignment 2**

We were tasked with solving the exam scheduling problem in this assignment.

**The Exam Scheduling Problem:**

The exam scheduling problem is an optimization problem that involves scheduling exams for a set of courses into time slots and exam halls. The problem is usually subject to several constraints, such as time slot availability, hall capacity, and student conflicts. This was done by genetic algorithm method and the details explaining the code are mentioned below.

The code contained many functions with different functionality that are briefly described below

**Problem Instance:**

The problem instance is defined at the beginning of the code by specifying a set of courses, exam halls, and time slots. Additionally, there are some constraints to consider, such as a maximum number of hours a hall can be used and a set of conflicting students between some courses. This information is used by the genetic algorithm to generate and evaluate candidate solutions.

**Fitness Function:**

The fitness function is responsible for evaluating the quality of a candidate solution by assigning a score based on the constraints and objectives of the problem. In this case, the score is calculated based on the number of hours each hall is used, the conflicts between courses, and the number of hours each hall exceeds the maximum allowed.

**Initialization:**

The initial population is generated by creating a list of random solutions. Each solution is represented as a list of tuples (course, time\_slot, exam\_hall), where each tuple represents the assignment of a course to a specific time slot and exam hall. The population size is determined by the pop\_size parameter.

**Selection:**

The selection process is performed using the tournament selection method. In this method, a group of candidates is randomly selected from the population, and the best candidate is chosen as the parent. The number of candidates in each tournament is determined by the k parameter.

**Crossover:**

The crossover operation is performed between two parents, generating two offspring. The single-point crossover method is used, which selects a random point in the chromosome to exchange information between the parents.

**Mutation:**

The mutation operation randomly changes the value of some genes in the offspring chromosome. This operation helps to introduce diversity into the population and avoid local optima. The mutation rate is determined by the mutation\_probability parameter.

**New Population:**

The new population is created by repeating steps 4 to 6 until the desired population size is reached. The new population replaces the previous generation, and the process repeats for a set number of generations determined by the ngen parameter.

**Execution:**

Finally, the algorithm is executed with some predefined parameters, and the best solution found is printed, along with its fitness score. The main() function controls the execution of the genetic algorithm, and the parameters can be adjusted as necessary.

Below is the output shown for the code:

