Faizan Pervaz 20I-0565 Section E

The given Python code implements a genetic algorithm to solve an exam scheduling problem. The problem involves scheduling exams for N courses into K exam halls, with T time slots available each day. The goal is to minimize the total number of conflicts between exams while satisfying all constraints of the problem. The constraints include limiting the number of hours a hall can be used in a day and ensuring that no two exams are scheduled at the same time for the same course or for courses in conflict\_pairs.

A list of tuples is used, where each tuple represents an exam and contains information about its course, time slot, and exam hall. The fitness function is designed to calculate the fitness of a solution based on the number of conflicts between exams and the number of hours each exam hall is used.

The genetic algorithm function starts by generating a random population of solutions. Then, it selects parents for crossover using tournament selection and performs crossover and mutation to generate offspring as stated in the assignment. The fitness of each offspring is evaluated, and the next generation of the population is selected using elitist selection. This process is repeated for a fixed number of generations 100 in our case, and the best solution found is returned.

The main function in the code is genetic\_algorithm(), which takes several input parameters that can be adjusted to improve the performance of the algorithm. The values used in the code are population\_size=100, tournament\_size=5, crossover\_prob=0.8, mutation\_prob=0.1, and num\_generations=100.

Finally, the code prints the best solution found by the genetic algorithm, which represents the best available exam schedule that satisfies all the constraints of the problem and minimizes the number of conflicts between exams.

Disadvantages:

* Perimeter Sensitive
* Tendency to converge to local optima
* Computational cost can be high

Advantages:

* Can Handle large and complex searches
* Can find good solution to complex optimization problems
* Can handle wide range of problem types