CSIS, BITS Pilani K. K. Birla Goa Campus

Artificial Intelligence (CS F407)

Programming Assignment 1
Total Marks: 15
Submission Deadline: 9 PM on 14/02/2021 (Sunday)

Each student must individually do this programming assignment. Your program must be written in Python and should run (without errors) on Python 3.6 or later.
Any form of plagiarism will result in -5 marks being awarded. Note that the deadline is **9 PM** and not midnight. Five marks per day will be deducted for submissions after the deadline. It will be your responsibility to submit the assignment well in advance and avoid unforeseen problems like power failures etc.

Question 1 (7 marks)

Use genetic algorithm as given in Figure 4.8 (Page 129) of the textbook to solve the 8-queens problem.

Let, Fitness function $= 1 +$ number of pairs of queens not in attacking position.

First, implement the version of the GA algorithm given in the textbook. Assume, that initially all the states in the population are the same and correspond to the eight queens being on the same row. (That is, initially every state in the population has a fitness value of 1.) For the first algorithm, let the population size be 20.

Next, come up with a variant of the GA algorithm so that the best fitness value in a population improves in a faster manner over successive generations. In the second algorithm, use the same initial population, the same state representation and the same fitness value function as used in the first algorithm. Make changes only to the other details of the GA algorithm (i.e. size of the population, $Reproduce()$ function, $Mutate()$ function etc.).

Plot a graph that compares the first algorithm with the second algorithm. The y-axis of the graph must represent the best fitness value in a generation and the x-axis must represent the number of generations.

In the report, explain how you have improved the GA algorithm for the 8-queens problem. Include the graph mentioned above in your report.

Question 2 (8 marks)

Use genetic algorithm to solve the travelling salesman problem (TSP). There are 14 cities labelled A to N. Pairwise distances between the cities are given in the table below. The unit of distance is 1000 kilometers.

Table 1: Pairwise distances in 1000 kilometers between the 14 cities

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
A	0	∞	∞	∞	∞	∞	.15	∞	∞	.2	∞	.12	∞	∞
B	∞	0	∞	∞	∞	∞	∞	.19	.4	∞	∞	∞	∞	.13
C	∞	∞	0	.6	.22	.4	∞	∞	.2	∞	∞	∞	∞	∞
D	∞	∞	.6	0	∞	.21	∞	∞	∞	∞	.3	∞	∞	∞
E	∞	∞	.22	∞	0	∞	∞	∞	.18	∞	∞	∞	∞	∞
F	∞	∞	.4	.21	∞	0	∞	∞	∞	∞	.37	.6	.26	.9
G	.15	∞	∞	∞	∞	∞	0	∞	∞	∞	.55	.18	∞	∞
H	∞	.19	∞	∞	∞	∞	∞	0	∞	.56	∞	∞	∞	.17
I	∞	.4	.2	∞	.18	∞	∞	∞	0	∞	∞	∞	∞	.6
J	.2	∞	∞	∞	∞	∞	∞	.56	∞	0	∞	.16	∞	.5
K	∞	∞	∞	.3	∞	.37	.55	∞	∞	∞	0	∞	.24	∞
L	.12	∞	∞	∞	∞	.6	.18	∞	∞	.16	∞	0	.4	∞
M	∞	∞	∞	∞	∞	.26	∞	∞	∞	∞	.24	.4	0	∞
N	∞	.13	∞	∞	∞	.9	∞	.17	.6	.5	∞	∞	∞	0

Note: As discussed in class, a state in the state space can be represented as a permutation of $(A, B, \ldots, N)$. However, creating a child node from two parent nodes might lead to an invalid state, because some city may be missed out or may appear more than once. One way to solve the problem is as shown in Figure 1. Read the caption in Figure 1.

Credits for Figure 1 : https://www.theprojectspot.com/tutorial-post/applying-a-genetic-algorithm-to-the-travelling-salesman-problem/5

First, implement a simple version of the GA algorithm for the TSP. Your goal is to minimize the path cost for the round trip. Construct a suitable fitness value function from the cost function.

Assume that initially all the states in the population are the same and correspond to the permutation: (A,B,C,D,E,F,G,H,I,J,K,L,M,N). Assume that the population is of size 20 for the first algorithm.

As in Question 1, come up with a variant of the GA algorithm so that the best fitness value in a population improves in a faster manner over successive generations. In the second algorithm, use the same initial population, the same state representation and the same fitness value function as used in the first algorithm. Make changes only to the other details of the GA algorithm (i.e. size of the population, *Reproduce*() function, *Mutate*() function etc.).

Plot a graph that compares the two GA algorithms for the TSP. The y-axis of the graph

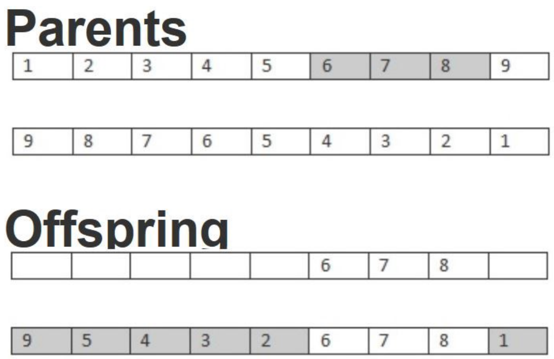

Figure 1: A possible way to generate an offspring from two parents that represent a permutation of $(1, \ldots, 9)$. First, a subset of the first parent is randomly selected. Then the remaining elements are taken from the second parent. While taking elements from the second parent, it is ensured that the elements appear in the same order as they did in the second parent. Also, it is ensured that no element gets duplicated.

must represent the best fitness value in a generation and the x-axis must represent the number of generations.

In your report, explain how you have improved the GA algorithm for the TSP. Include the graph mentioned above in your report.

Instructions for submission

- You must submit a single program file with the name “ROLLXYZ_FIRSTNAME.py.” Your program needs to include only the improved versions of the GA algorithm for solving the 8-queens problem and the travelling salesman problem. The two GA algorithms can differ in their details. The program should ask the user whether to run the algorithm for 8-queens problem or the TSP. Show the best fitness value for each generation and then show the best solution.

- Your report must be named “ROLLXYZ_FIRSTNAME.pdf”.

- Submit the two files mentioned above. There is no need to zip the two files.

- I will be comparing your submission with programs available on the internet. If there is a match, you will receive -5 marks.