Assignment 3 Report

1 Travelling Sales Person and Genetic Algorithm

1.1 Part a

Explain the important operations of the employed algorithm.

The algorithm starts with creating a population and then tries to evolve the population by mutating and doing crossover on the individuals. Calculates the individual's fitness score and chooses some individuals to pass on to other generations. Then new generation and selected ones continue this process until the fitness calculation count reaches the given maximum fitness count.

1.1.1 Initialization

The algorithm begins by initializing a population of individuals which are candidate solutions. In this case, each individual represents a possible route for the salesman to travel through all the cities. The initialization process is done randomly, for each individual has a random permutation of the cities. The number of population is selected as 100 for this implementation.

1.1.2 Fitness Evaluation

Once the initial population is generated, the fitness of each individual solution is evaluated. Fitness is defined as the total distance traveled along the route. Lower distances indicate better fitness. Therefore, their fitness score is equal to $\frac{1}{\text{total\_distance\_of\_route}}$, in that way, the lower distances get higher fitness scores. Total_distance_of_route is calculated by summing each city pair's distances in the permutations.

1.1.3 Selection

Selection is the process of choosing individuals from the current population to serve as parents for producing the next generation. There is more than one selection algorithm like roulette selection or tournament selection. In my implementation, I choose two parents for crossover. One of the parents is chosen from the first "numBests" best fitness-valued individuals and the other is selected randomly from the population. The "numBests" is selected as 10 for this implementation.

1.1.4 Crossover

Crossover is the genetic operation where two parent solutions are combined to create offspring solutions. In the crossover, parts of the paths from two-parent routes are chosen and merged to create a new individual as a new route. This operation helps explore the solution space efficiently. In this crossover implementation, two parents are taken, and a random segment from parent 1 is extracted and placed into a new individual. The remaining elements are then filled with elements from parent 2 in sequential order. The first parent is always selected from the best individuals.

1.1.5 Mutation

Mutation adds random alterations to individual solutions. Mutation can involve swapping two cities in a route like what is done in this implementation. This action ensures diversity among solutions, preventing the population from prematurely converging to less optimal outcomes. But this mutation is not done for each individual, for there is a mutation_rate which is given to the algorithm. It is selected as 80% for this implementation. Mutation_rate means the probability that a newly created individual is going to have a mutation or not.

1.1.6 Replacement

Once offspring are generated via crossover and mutation, they replace the previous generation. Different replacement strategies exist, like elitist replacement, which preserves the best individuals, or generational replacement, where the entire population is substituted with the new offspring. In this implementation, elitist replacement is employed. The top "numBest" individuals from the previous generation are retained for the next generation, while the rest are generated through crossover and mutation.

1.1.7 Termination

The algorithm stops when it meets a specific condition. This condition could be reaching a maximum number of generations, achieving a satisfactory solution quality, or reaching a computational time limit. For this implementation reaching the maximum fitness calculation is used for termination. It restricts the number of fitness calculations.

1.2 Part b

Explain the representation of the individual, a solution to the problem, in your algorithm

For my implementation, each individual is a representation of a path. There were 52 cities and each city had a specific representation of char according to their order in the given distance input. Therefore our individuals have a string to represent the path. Each individual is started with a given start city and finished with the started city. Example individual is:

$$P = [p_1, p_2, \ldots, p_{n+1}]$$

where n is the city number and $p_1 = p_{n+1}$. p_i is a character generated according to the given cities' distances input. The formulation to calculate the exact character value that represents the city is:

$$\text{char} = \text{CHAR\_START} + \text{index of city in input}$$

which is a char in ascii table. CHAR_START is selected 48 for this question for better ascii character choices.

1.3 Part c

Give the equation of the fitness function used by your algorithm. P is an example individual to calculate the fitness. $X(p_i)$ is a function to calculate the x coordinate, and $Y(p_i)$ is a function to calculate the y coordinate. These operations are done by turning the char value to the index value at a given input and checking the coordinate there. For calculating fitness value the Euclidian distance is used which is :

$$ED(P_i, P_j) = sqrt(X(P_i) - X(P_j))^2 + (Y(P_i) - Y(P_j))^2)$$

The fitness function is calculated by summing up each pair's Euclidian distance in the route. n is the city count, the path has n+1 element because it should return the starting point.

$$F(C) = \sum_{i=1}^{n} ED(P_i, P_{i+1})$$

1.4 Part d

Give the parameters used in your algorithm. Examples: Population size, crossover rate . . .

Pop_SIZE = 100 which is the population size. MutationRate = 80 which is used for the calculating probability of mutation in terms of percentage. numBests = 10 which is used to detect the number of elites in the population. MAX_FITNESS_CAL = 250000 which is used for terminating the point of the algorithm.

1.5 Part e

Illustrate how the performance of the population evolves with generations (with a figure.) Write text with the figure as well There are 2500 generations. It started too fast with getting better then the pace of getting better got slowed.

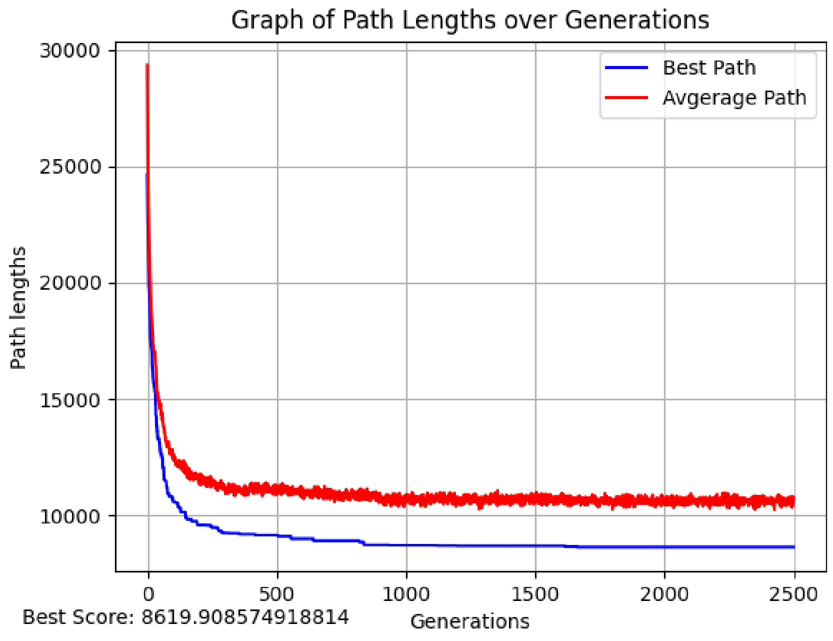

Figure 1: Population evaluation with generations

This implementation holds the generation's best scores and the averages. Both of them reduced in time. This implementation found a path with a value lower than 9000 which is 8619. Each generation had 100 population therefore to calculate 2500 generation it is done 250000 fitness calculations.