Report

Explain the important operations of the algorithm

Selection, crossover and mutation are three important operations that is included in this algorithm. The selection in my algorithm is a **Roulette selection,** that will take the number of parents and store them in a list and I add the sum of all the parents fitness value into a variable that I call sum0fFitness. For each parent I take their Fitness value and divide that with the sum of fitness, and that value will be the probability of that parent and is inserted into a probability list. That probability is later used when we choose parents indexes for the crossover. It will chooses a random number between 0 and 1 and match that with a parent matching that probability.

The crossover is done by taking two individuals as parent and in parent 1 I randomly select two indexes and take the locations that are between that interval and add it into an array. From parent two I add every location into the same array that doesn’t already exist in the array from parent 1. It returns one child.

In mutate we check if we are going to mutate that child first with the mutation probability, if the case is true, I randomly select two indexes and between that interval, flip the locations around. The two selected indexes will swap places with each other, the spots next two them will swap with each other, and the program continue to do this between the interval until it has reached the middle point.

Explain the representation of an individual

For the individual, that represents one solution of the problem, I use real representation. Each individual has access to a list of all the location each solution will go through and that list contains X and Y-coordinates to each city and an id, all those three is represented by real numbers.

Give the equation of the fitness function

d = – the distance between two cities

– the total distance for all cities for one individual

Give the parameters used in your algorithm

The parameters used in the algorithm are **POPULATION\_SIZE**, **SURVIVORS**, **NUMBER\_OF\_PARENTS** and **MUTATION\_PROBABILITY**. The population size represents the number of individuals that can be in one generation, and in my algorithm, the population size is 500, so one generation can contain 500 solutions (individuals). Survivors are the number of individuals that survive into the next generation and that is decided depending on their fitness, so I take the 60 bests from the previous generation and put them into the next one. The survivors are used to achieve elitism. The number of parents is 200 and it is those individuals that will be used in crossover and mutation. The mutation probability is used so I don’t mutate all. The mutation probability is set to 0.2 and if and if the random number, that are selected to compare with, is lower than the mutation probability, the program will mutate.

Illustrate in a figure how the performance of the population evolves with generations