ADDIS ABABA INSTITUTE OF TECHNOLOGY

Artificial intelligence group Assignment

The knapsack and traveling salesperson problem

Report 2: TSP

Group members

1. Hayat Tofik ---------- UGR/ 2987/12
2. Paulos Dessie -------- UGR/6912/12
3. Tsega Yaekob -------- UGR/8485/12

Submitted to – Ammanuel

Contents

2.1 Genetic Algorithm to solve Traveling salesman problem

To solve the TSP with genetic algorithm the steps followed are provided below:

First, the ‘population’ function generates a list population using the size of the node list. The size of this initialized population is directly related to the number of cities which we specified as nodes.

The ‘fitness_value’ function is basically used to find the better solution. It uses a loop based on a specified condition and if the condition is satisfied then the fitness of each chromosome is calculated using the Euclidean distance. It means that calculating the distance between one node and the next node using Euclidian distance. After the fitness of the chromosome is calculated it is stored in a variable ‘chrift’.

The ‘selection’ function basically tries to generate a new population with higher fitness than the current population. Using a loop, it updates the fittest value to the one that have higher fitness value which is calculated using the function ‘fitness_value’. Then it returns the relatively fittest chromosome.

The ‘cross_over’ function is used to cross the individual node with higher quality since genetic algorithm is all about finding a better individual solution. So, first from a parent population we select best individual nodes and another new node and then we cross the two or its basically changing the position of the two individuals to create a high-quality chromosome. Since we are using two parents, we need to cross each individual from each parent population. So, we used a loop and this process continues for each individual item. And finally returns the ‘listOfChidren’ which contains the crossed chromosome that are supposed to have high quality.

Then comes the ‘mutation function’. Since mutation is performed in a single population by changing one bit with another. What this function does is the same it mutated the crossed population.

The final output of the algorithm is provided below. Here as we can see what we want is to visit each city ones and to return to the original city given that the path is the minimum. And as we can see in the output the Average value which is the distance of the path that we chose is minimized and the average time it takes is also minimized. This is basically what we wanted the algorithm to do.

Final Output

Average time (s)	Average distance
0.0032110214233398438	138.20498669395724

2.1 Simulated annealing to solve Traveling salesman problem

In this algorithm, we first set the initial temperature and the rate at which the temperature changes to be 1. The first is then generated randomly and compared to the second solution; if the current value is better, it will remain that way; if the second value is better, the current value will be replaced with the second value; and if the second value is less favorable than the current value, the current value will be set based on the heuristic-based random condition.This will help not to get stuck on local maxima or plateaus.

Final Output

Average time (s)	Average distance
0.23340559005737305	150.84767450228713

Situated 20

time-0.47373223304748535

dis-207.2631037878411

Tsega, [6/22/2022 11:38 PM]

Simulate 8

Tim 0.14261817932128906

Dis dis-198.2677293532136

2.12 Hill climbing to solve Traveling salesman problem

In this solution we have used simple hill climbing algorithm that goes from one point to the other or one solution to the other based on the path's total distance if the next generated solution is better than the current we are going to take it as current solution we have repeated this until no more better solutions are there.

Average time (s)	Average distance
0.000997304916381836	159.50209250645185

8 data hill_clim   dis-37.12910135379245

time-0.0009970664978027344

20 hill time-0.001994609832763672

dis-184.12893459456166

For different Settings:

Simulated annealing

The simulated algorithm was tested for different initial temperature for 17 cities as it was done in the knapsack problem the results where found to be as follows:

Final Output

Average time (s)	Average distance	Initial temprature
0. 1202641999698244	158.87063059294675	10

0.09509939997224137 ms	172.56908713396896	5
0.09049999061971903ms	163.56347500978018	1
0.11840980005217716ms	175.27003637809196	0.1

The Traveling Salesman Algorithm's conclusion

The hill climbing algorithm completes execution in much less time than the other two algorithms, but the path quality is not as good. The simulated annealing is also unable to be the best solution because there is a penalty for adding unexisting paths and the simulated annealing generates solutions more randomly than the genetic algorithm. Based on our results of the execution time and the total distance covered, we have found that the Genetic Algorithm gives the best result.