CSCI 4610 Assignment 2 Report

Abinash Narendiran 100588875
Saijeeshan Ketheeswaran 100589716
Rohil Arya 100585195

Discussion

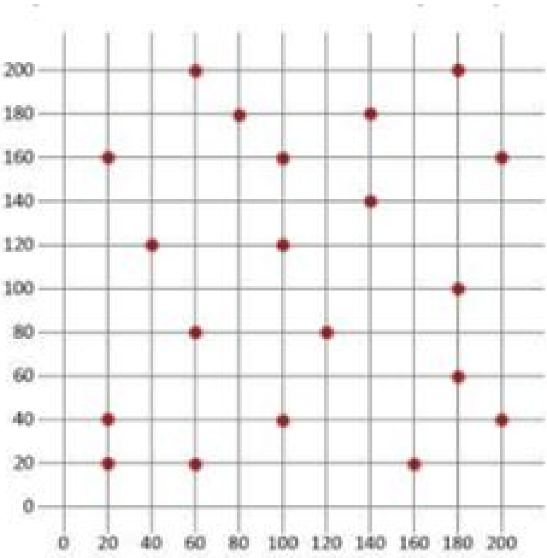
After getting the coordinates of all of the cities and adding them to a list array, the population of the sample was created randomly using the random () function in Python. Afterwards, the fitness for each of the cities in the sample population was evaluated. This fitness was evaluated by determining the route distance between the chosen city and another city in the same sample. Based on the route distances of all the cities in the population the cities were then numerically ranked based on their fitness. After determining the fitness of each of the cities in the population, Selection was used to improve the overall fitness of the population sample. This was done by selecting the cities that had a fitness greater than a randomly generated threshold. The chosen cities were appended to a new list array which consisted of cities with routes that are “fitter”. After completing the Selection component of the genetic algorithm, we construct a crossover of two “fit” cities in the Mating component. This is done by taking adding the selected population to the index of the initial sample population. This crossover will produce a child offspring that is appended to a new list array. The final step of the genetic algorithm was the mutation process which involved swapping coordinates of a select sample of cities, which was done in order to add randomness into the population sample. The genetic process is repeated for a certain amount of generations, which in our case for problem domain 1 was set at 100. The amount of generations set for problem domain 2 was 50.

The termination condition of our algorithm is the number of generations our population goes through the genetic algorithm cycle. Essentially every time the population of cities goes through the genetic algorithm cycle it becomes stronger. The generations variable is used to determine a point at which the population can no longer become stronger, and therefore must stop repeating the cycle. Considering the sizes of the population for both our domains, we chose a relatively high number variable for our generations

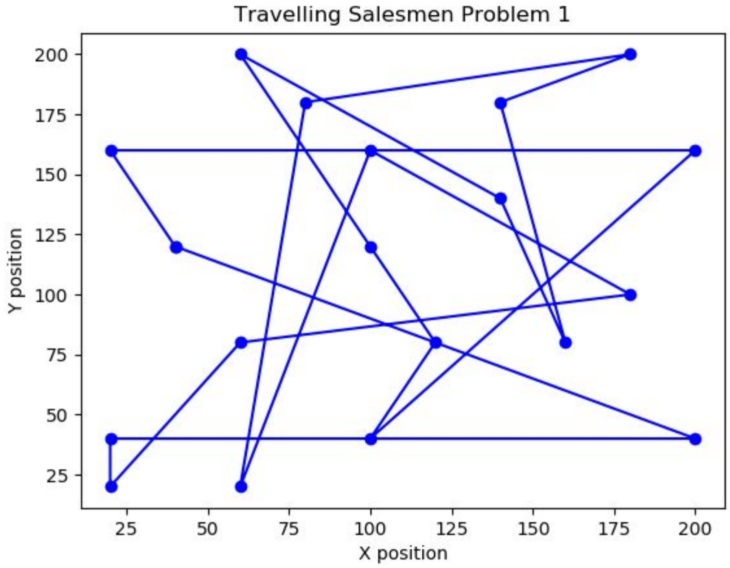
Graphical Output

Problem Domain 1

A population of 20 cities was added to the initial population. The coordinates (x, y) of the cities were as followed: (20,20), (20,40), (20,160), (40, 120), (60,20), (60,80), (60,200), (80,180), (100,40), (100, 120), (100,160), (120,80), (140, 140), (140,180), (160,80), (180, 60), (180,100), (180,200), (200,40), (200,160).

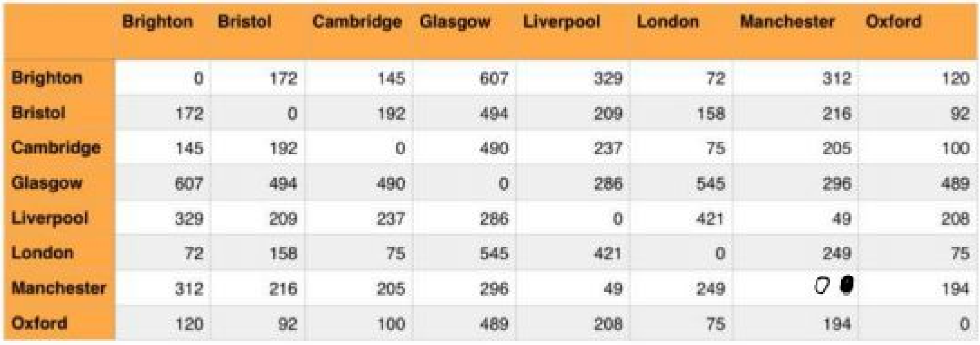


Graphical Solution:



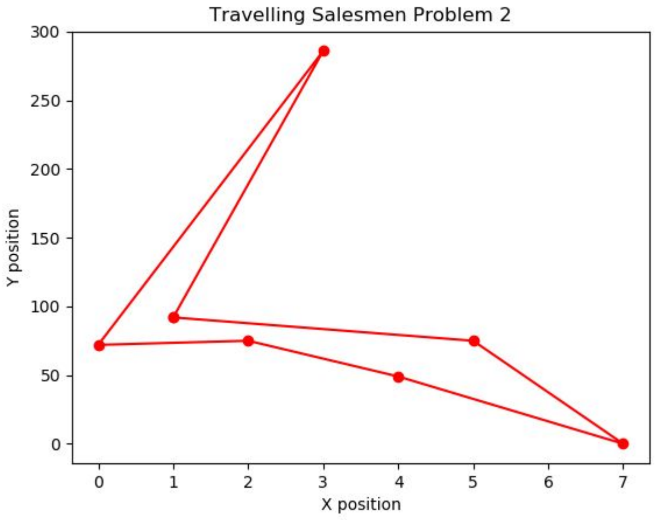
Problem Domain 2

A population of 8 cities was added to the initial population. The distances between each of the cities are as followed:



	Brighton	Bristol	Cambridge	Glasgow	Liverpool	London	Manchester	Oxford
Brighton	0	172	145	607	329	72	312	120
Bristol	172	0	192	494	209	158	216	92
Cambridge	145	192	0	490	237	75	205	100
Glasgow	607	494	490	0	286	545	296	489
Liverpool	329	209	237	286	0	421	49	208
London	72	158	75	545	421	0	249	75
Manchester	312	216	205	296	49	249	0	194
Oxford	120	92	100	489	208	75	194	0

Graphical Solution:



Comparison of Results

Total Distance Cost

Mutation Rate	Problem Domain 1	Problem Domain 2
0.01	2153	622
0.05	2251	869
0.1	2146	902
0.25	2397	670
0.5	2190	810
0.75	2432	776
0.9	2534	882

Work Done

Everyone in our group worked on the code together to figure out the assignment because working it on seperate made it harder for us to finish the assignment. However we did split up certain parts. For example Abinash was responsible for the initialization and evaluation, while Saijeeshan worked on the selection component of the algorithm. Rohil worked on the crossover and mutation components. We later came together and combined our parts and successfully completed the genetic algorithm. We all worked on the report portion, as well.