COMP5511 Artificial Intelligence Concepts
- Project 1 -

The Traveling Salesman Problem

21030816g Chiu Pit Ho

In this project, I use python to solve the problem, the code is referenced by **MorvanZhou** (**https://github.com/MorvanZhou/Evolutionary-Algorithm/blob/master/tutorial-contents/Genetic%20Algorithm/Travel%20Sales%20Person.py**).

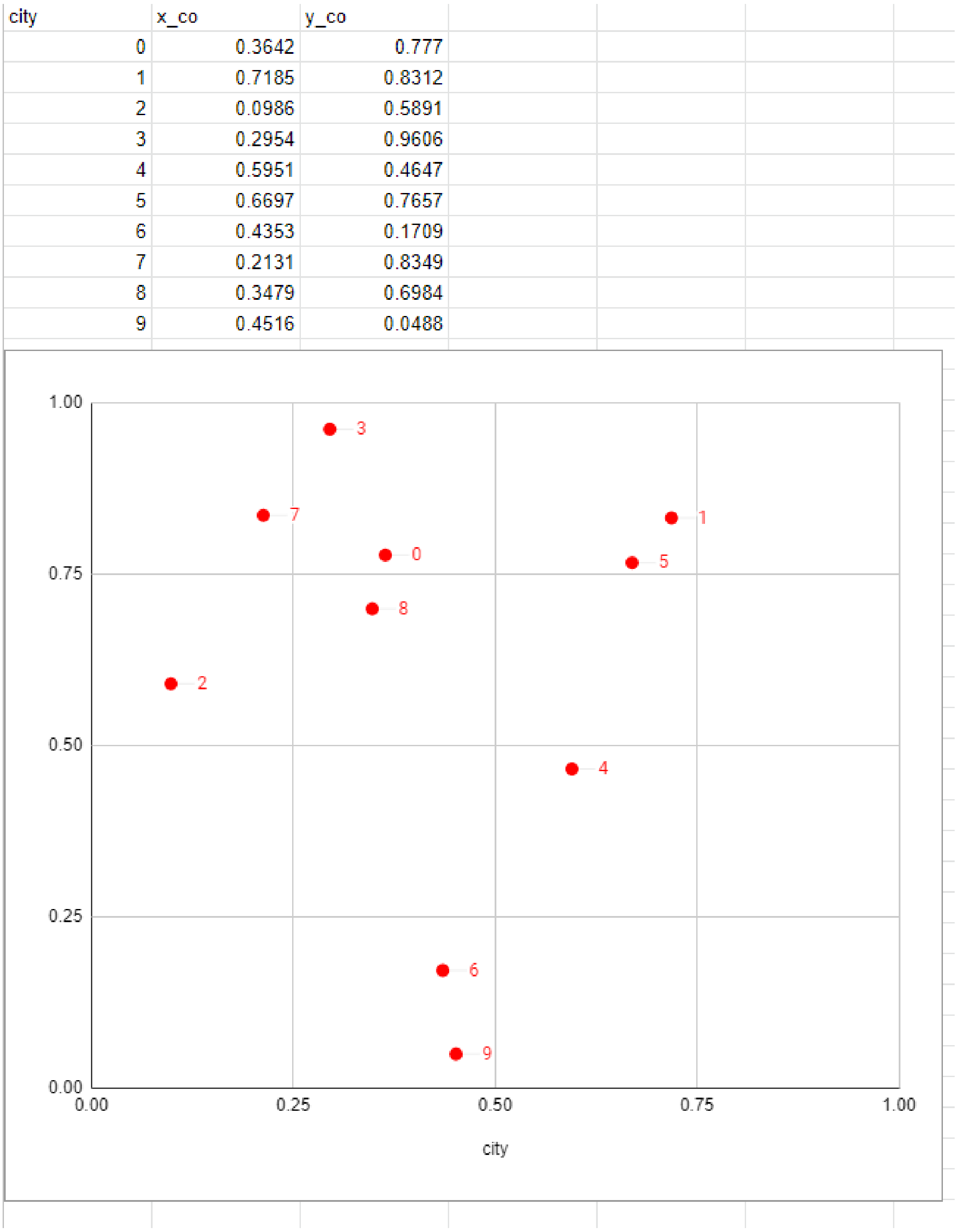
By referencing his code, I modify few parts:

1. Make the code can load our dataset (10 city locations)
2. modify the **get_fitness function to perform rank weighting selection.**

The code is in the file - GA.html

Result

After running the program several times, I found out that the shortest distance is **1.94084**. There are many ways to achieve the shortest distance, such as [2 7 3 0 8 1 5 4 9 6] , [2 7 1 5 6 3 0 4 9 8], ect.



city	x_co	y_co
0	0.3642	0.777
1	0.7185	0.8312
2	0.0986	0.5891
3	0.2954	0.9606
4	0.5951	0.4647
5	0.6697	0.7657
6	0.4353	0.1709
7	0.2131	0.8349
8	0.3479	0.6984
9	0.4516	0.0488

Finding

After trying different parameters, I find out some insight:

- POP_SIZE (population size)

Since each dna comes from random generation , if the population size is larger, the dna is more diverse. If the population size value is not large enough, the DNA diversity is low, and the ability to rely on the evolved offspring will also be insufficient.

- N_GENERATIONS (number of generations)

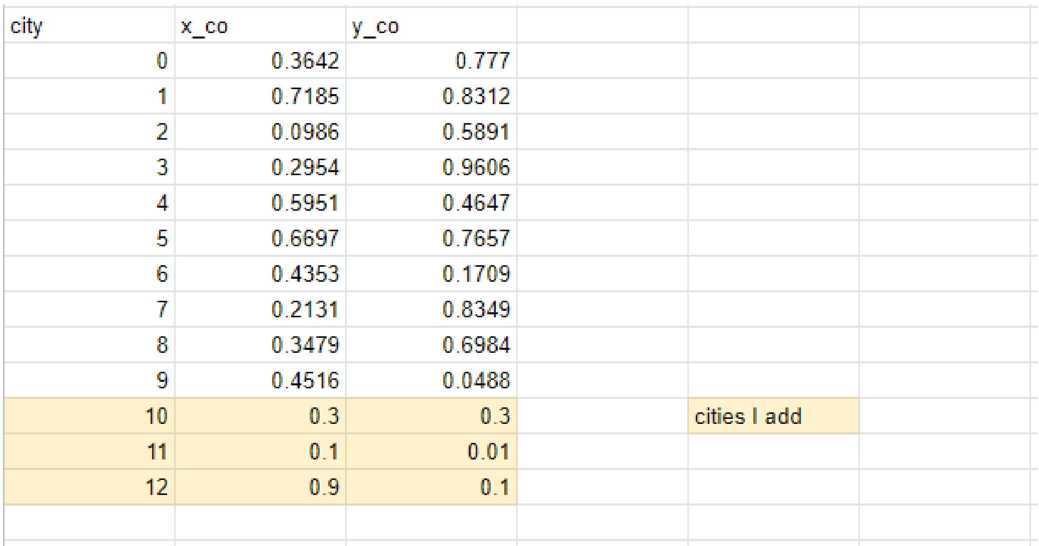
Number of generations represents the number of times of evolution, the more the number of times, the improvement of the ability of dna.If this value is insufficient, the outcome will not converge, because DNA has not undergone a sufficient number of selections, crossovers and mutations.

- CROSS_RATE (crossover rate) and MUTATE_RATE (mutation rate)

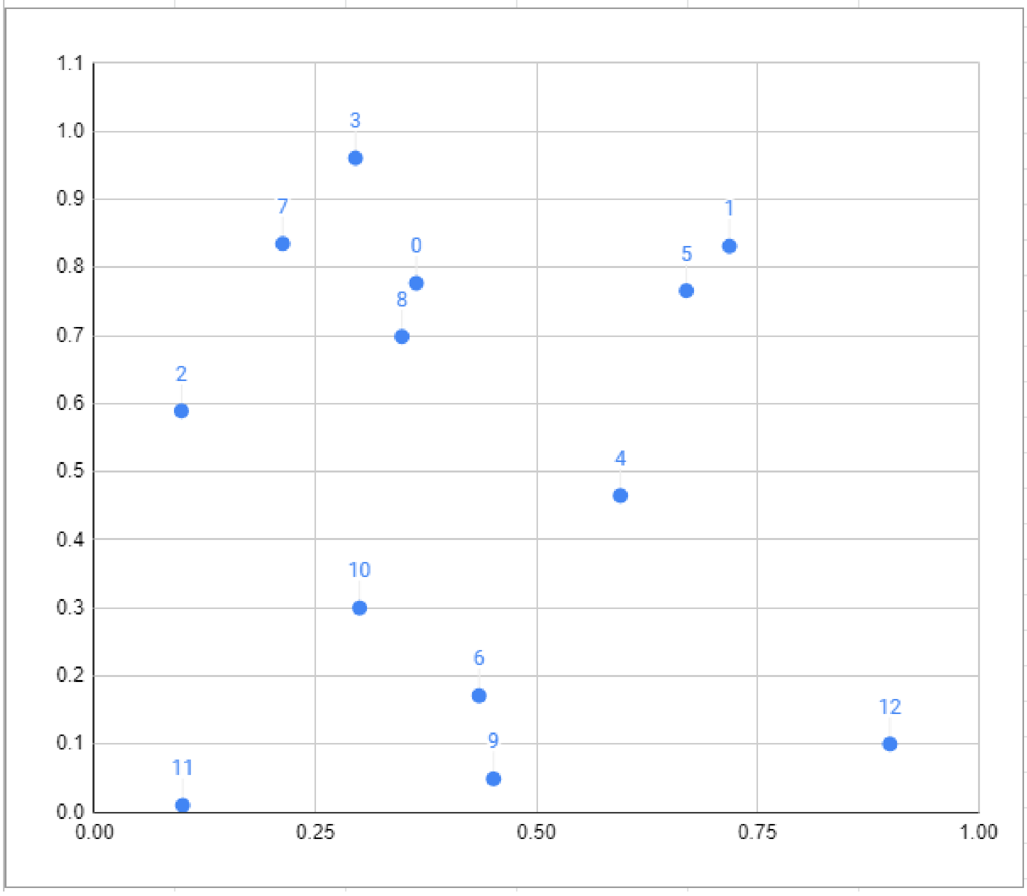
The role of both is to generate new DNA during evolution. If the values of both are too large, this will make the resulting ability unable to converge, and if they are too small, new DNA will not be generated, reducing diversity.

Introduce more cities

- 3 cities are introduced



city	x_co	y_co			
0	0.3642	0.777			
1	0.7185	0.8312			
2	0.0986	0.5891			
3	0.2954	0.9606			
4	0.5951	0.4647			
5	0.6697	0.7657			
6	0.4353	0.1709			
7	0.2131	0.8349			
8	0.3479	0.6984			
9	0.4516	0.0488			
10	0.3	0.3		cities I add	
11	0.1	0.01			
12	0.9	0.1			

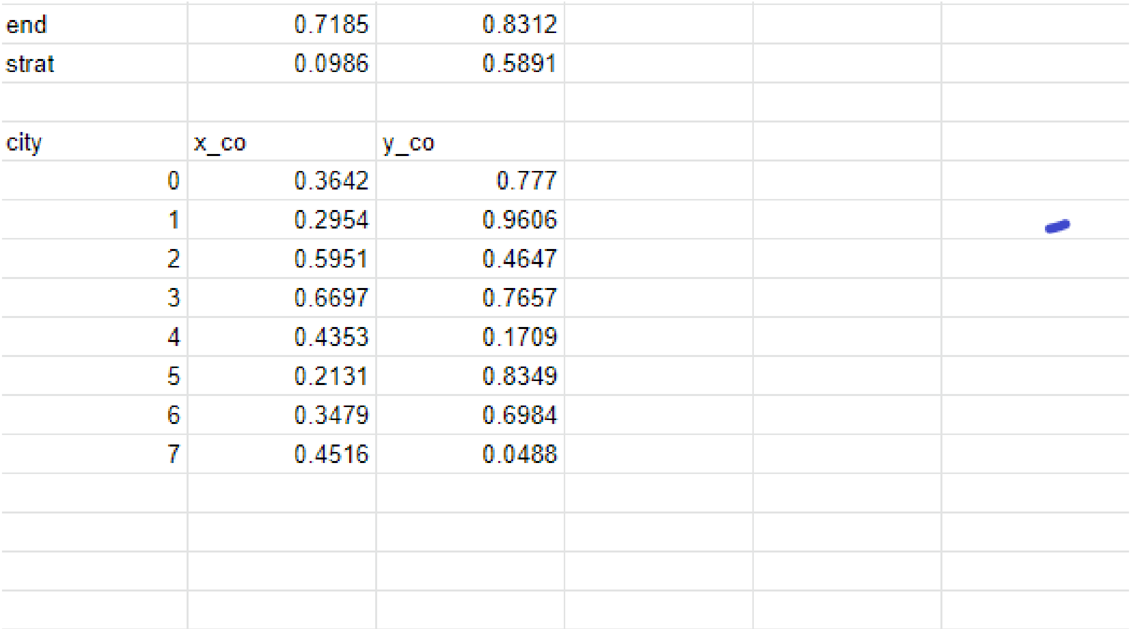


After running the program several times, I found out that the shortest distance is **2.97484** with paths [12 4 5 1 8 10 3 7 2 0 6 9 11] etc.

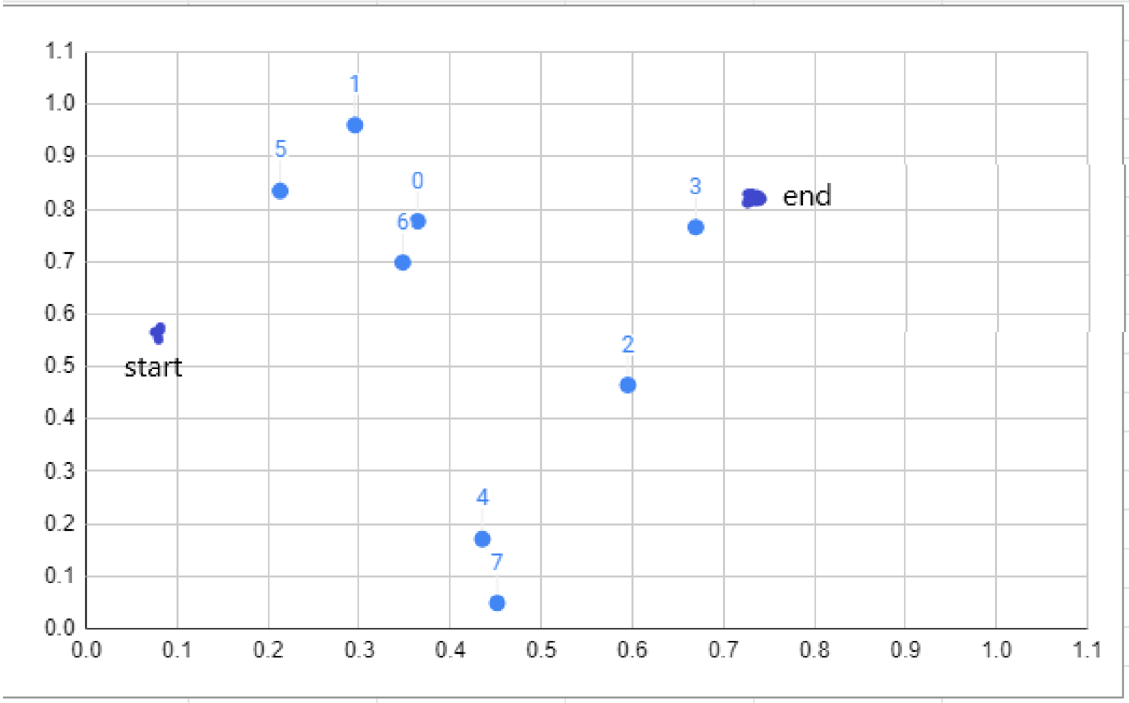
restrictions on the start and end point

In this part, I pick city 2 as the start, city 1 as the end. In the GA program, I only pass in the city [0,3,4,5,6,7,8,9] to create population and dna. Also, in the get_distance function, I need to calculate the distance between start and the first city, last and the end city. Finally and in the total_distance.

- **The code is in the file - restriction_S_E.html**



end	0.7185	0.8312
strat	0.0986	0.5891
city	x_co	y_co
0	0.3642	0.777
1	0.2954	0.9606
2	0.5951	0.4647
3	0.6697	0.7657
4	0.4353	0.1709
5	0.2131	0.8349
6	0.3479	0.6984
7	0.4516	0.0488



After running the program several times, I found out that the shortest distance is **2.18737** with paths [start 5 1 0 6 4 7 2 3 end] etc.

Extension:

In this part, writing the program and solving a problem is quite difficult for me.
I try to write down some opinions or general directions, to help me solve the problem.

(1) Asymmetric traveling salesman problem (ATSP)
The TSP problem is considered as an asymmetric one, i.e For any two cities A and B, the distance from A to B is different from that from B to A.
We can set up different distance calculation methods for different directions and cities.
For example, we have 10 cities. The max number of distances we need to calculate is 10p2 = 90.

(2) Sequential ordering problem (SOP)
Sequence constraints are required in real world problems. The salesman is asked to visit certain cities in a required sequence. A particular city has to be visited before some other cities.
Break it into different smaller problems? First we define the sequence, for each stage in the sequence, we can do the GA individually. We can get the shortest path for each stage.
Finally combine them to get the whole path.
(Dynamic programming?)

(3) Time window is considered in the problem. The salesman is required to visit certain cities within a certain time window. A data set with 100 cities is given (‘TSPTW_dataset.txt’).
Same answer with (3)

(4) For large-scale data, the cities can be divided into several regions (decided by students), the salesman must finish visiting all the cities within the region before traveling to any other cities in other regions. Using clustering techniques can decompose the large-scale data into several small-scale data sets by its relativity. The dataset with 50 points is provided (‘Cluster_dataset.txt’).
Also break it into different smaller problems, for each small region, we do the GA individually.
Finally do a big GA to get the whole path.
(Dynamic programming?)