UNIVERSIDAD DISTRITAL FRANCISCO JOSE DE CALDAS

SYSTEMS ANALYSIS

JUAN DIEGO LOZADA GONZALEZ

20222020014

CARLOS ANDRES VIRGUEZ

WORKSHOP II

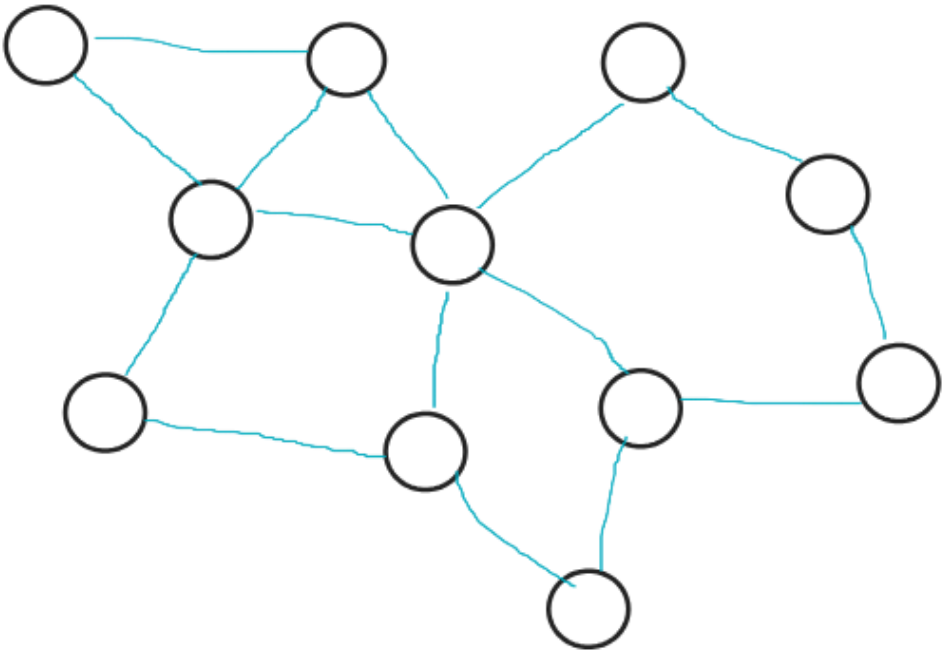
BOGOTA D.C

2024

WORKSHOP II

The workshop Swarm Intelligence and Sinergy: Ant Colony for the Traveling Salesman Problem is a way to solve problem related to logistic.

1. Create some diagrams and explanations to represent/understand the problem



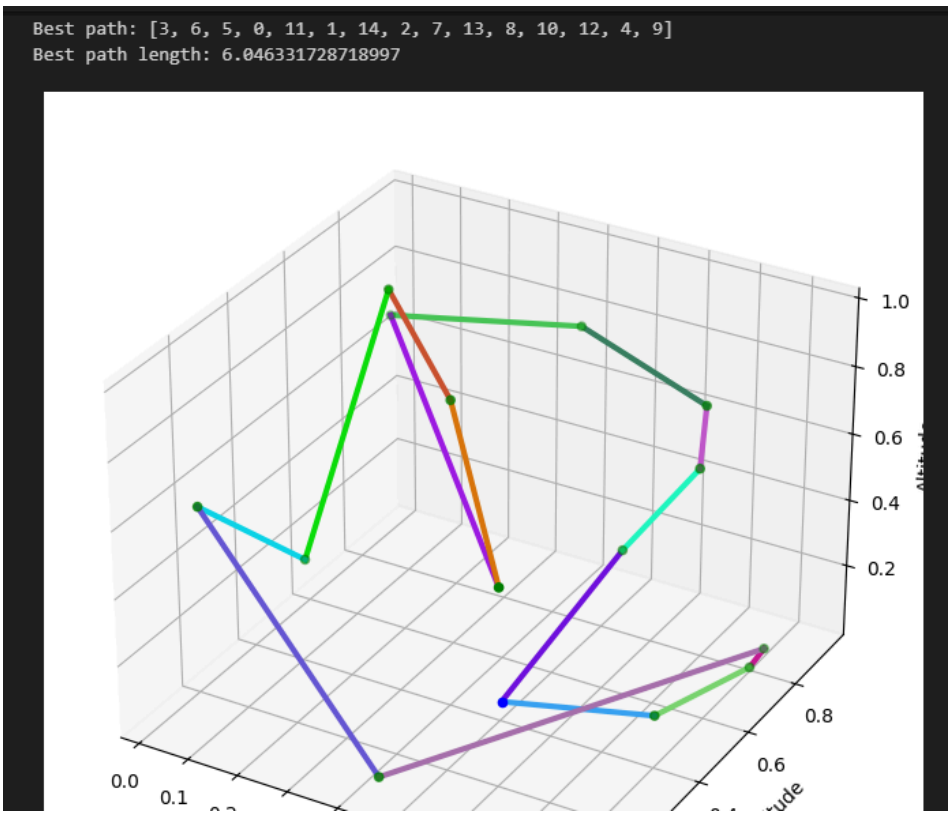
The problem could be represented by graphs, where each arist is a city, but, the idea is visit all the cities in the least time possible, for that reason is an NP problem, it will cost a lot of time if we try do it in a manual way, for that, we use ants with the purpose to visit all the cities in all the possible ways but taking account the pheromone, the pheromone serves to show to short path between each city.

6. After the code we can realize that:

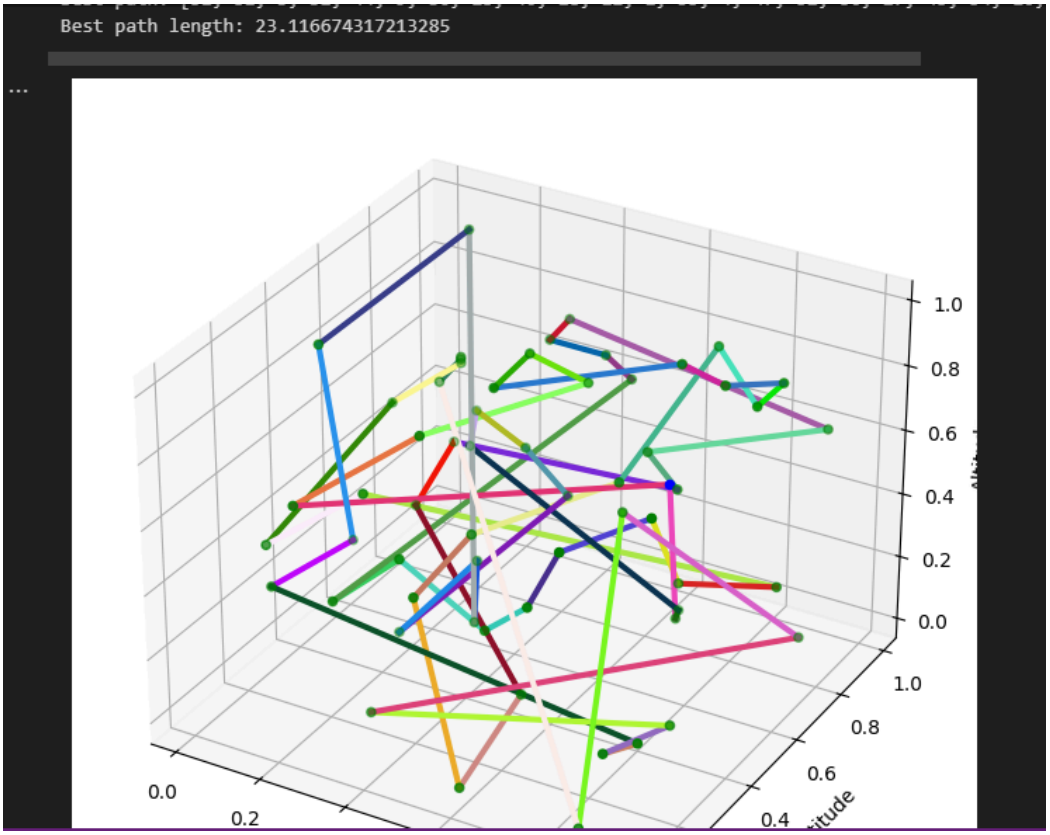
First, the base test i took, was

```
# model parameters
number_cities = 15
number_ants = 10
number_iterations = 10
alpha = 1
beta = 1
evaporation_rate = 0.5
Q = 1

# HERE create list of cities
cities = generate_cities(number_cities)

# HERE call ant_colony_optimization function
best_path, best_path_length = ant_colony_optimization(cities,number_ants,number_iterations,alpha,beta,evaporation_rate,Q)
```

and the output was the next one

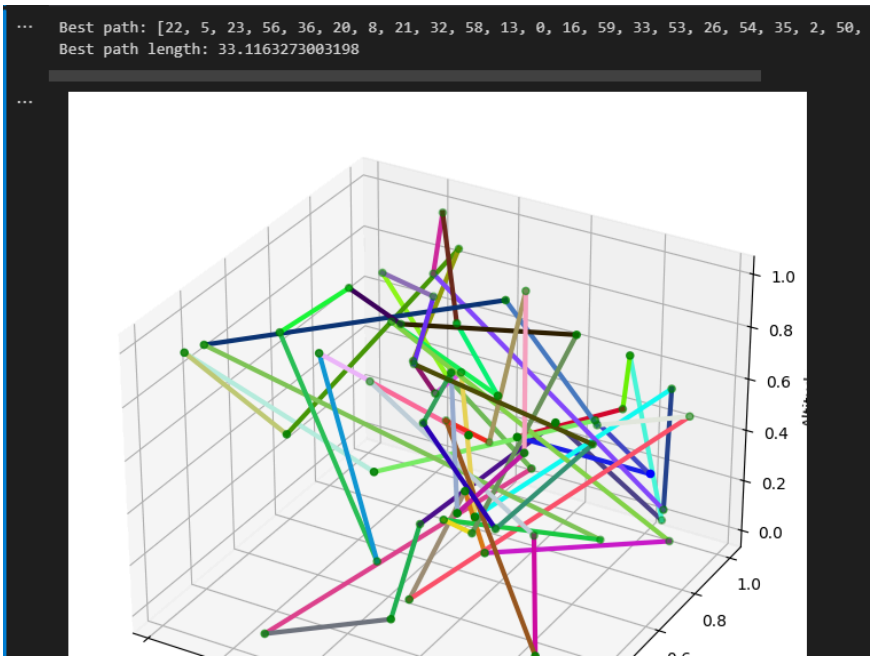


Between we put more cities the path length will increase

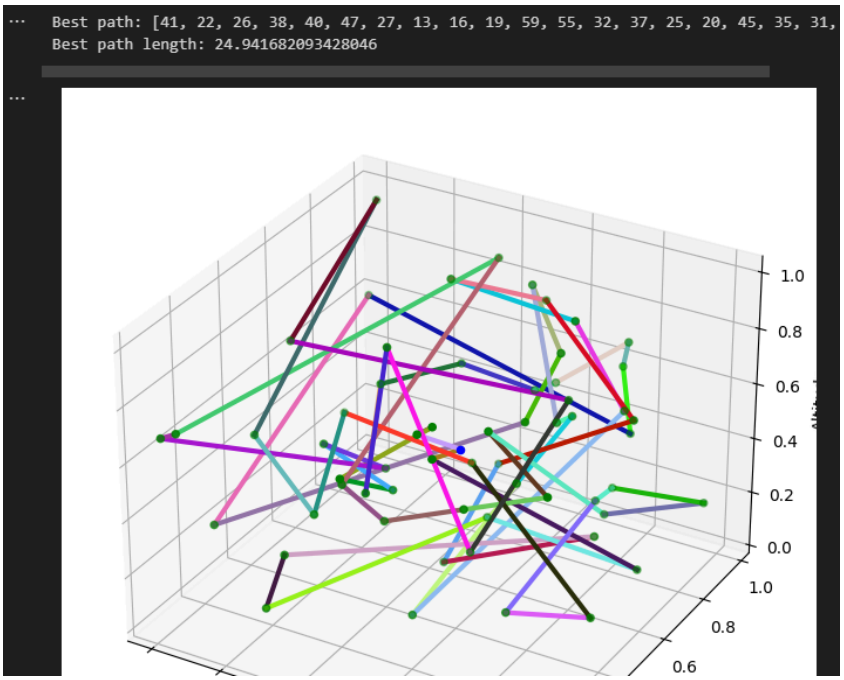


But if we put more ants to visit the cities, the path length will be shorter, but the duration of compiling will be more, the problem is when we have a high number of cities and a lot of ants, the compute will be delay

When alpha (It determines how much the ants are influenced by the pheromone trails left by other ants) and beta (It determines how much the ants are influenced by the distance to the next city) are lower, the path start to be more mix, and also de the path is more large

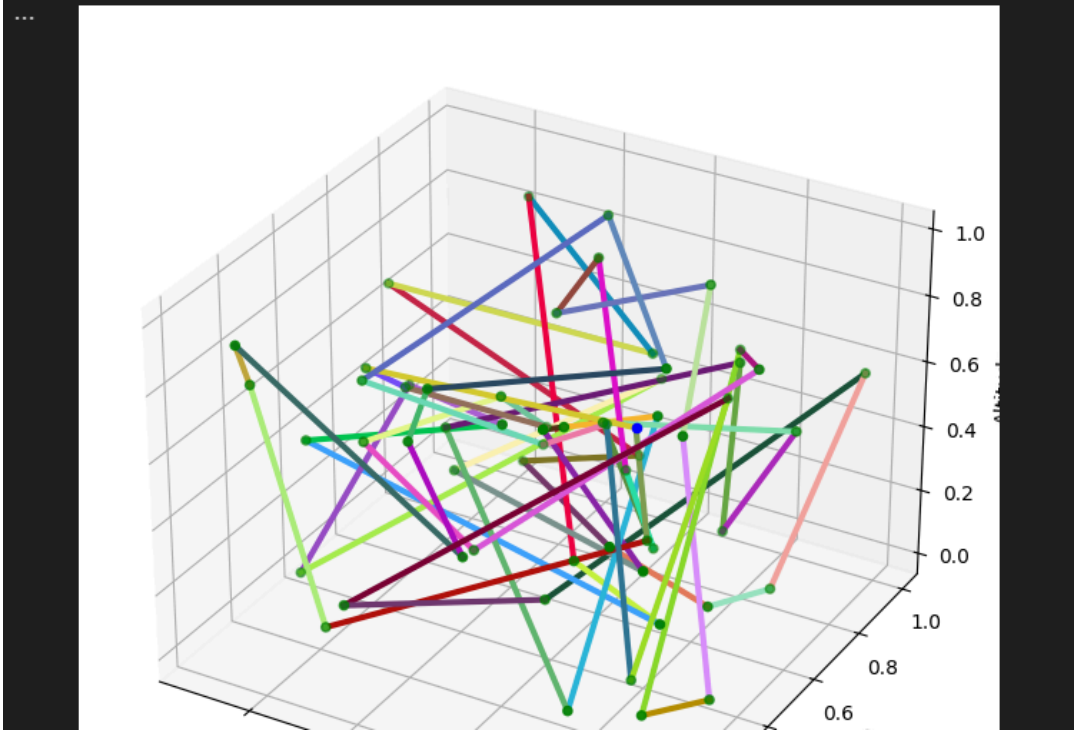


When beta is 1 and alpha 0.2, the path length start to reduce due to the ants are influenced by the distance to the cities,



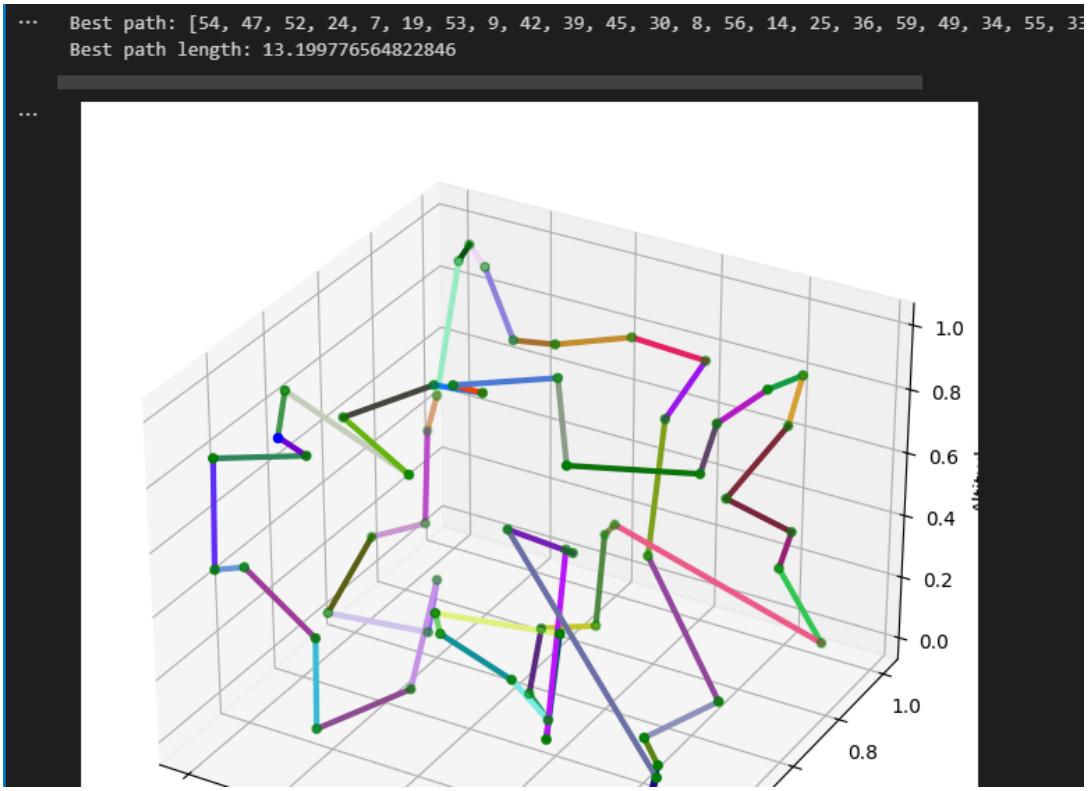
When viceverse,

```
Best path: [34, 57, 22, 3, 59, 42, 33, 17, 43, 29, 39, 8, 53, 48, 10, 13, 6, 28, 32, 54, 35,
Best path length: 32.05695594856159
```

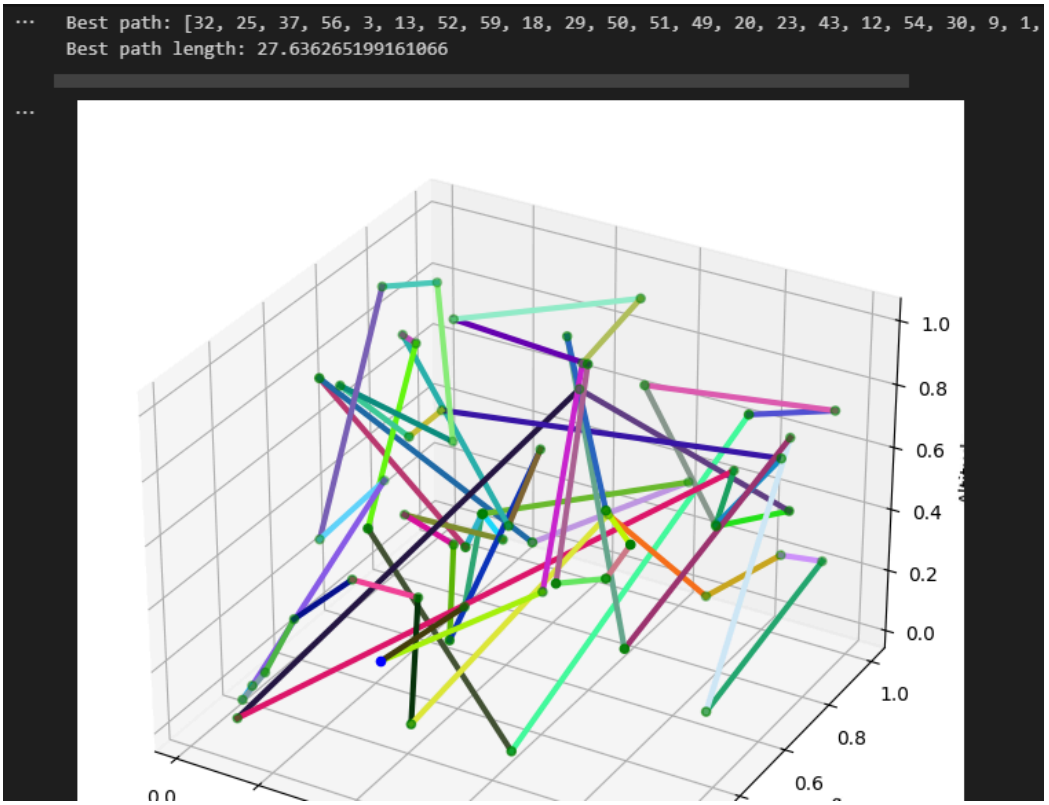


We see that the length between the cities is more extend, someone can see that there are cities near to others but the ants decide to choose other option, due to pheromone. But i someone wants to see a big difference with the reduction of the path length is necessary to increase the iterations, in my case i put 100 iterations the result was more efficient

```
Best path: [54, 47, 52, 24, 7, 19, 53, 9, 42, 39, 45, 30, 8, 56, 14, 25, 36, 59, 49, 34, 55, 33
Best path length: 13.199776564822846
```



When the evaporation rate is higher, the paths start to be less efficient, due to the evaporation is the responsable to forget past solutions, so the solutions start to be delayed



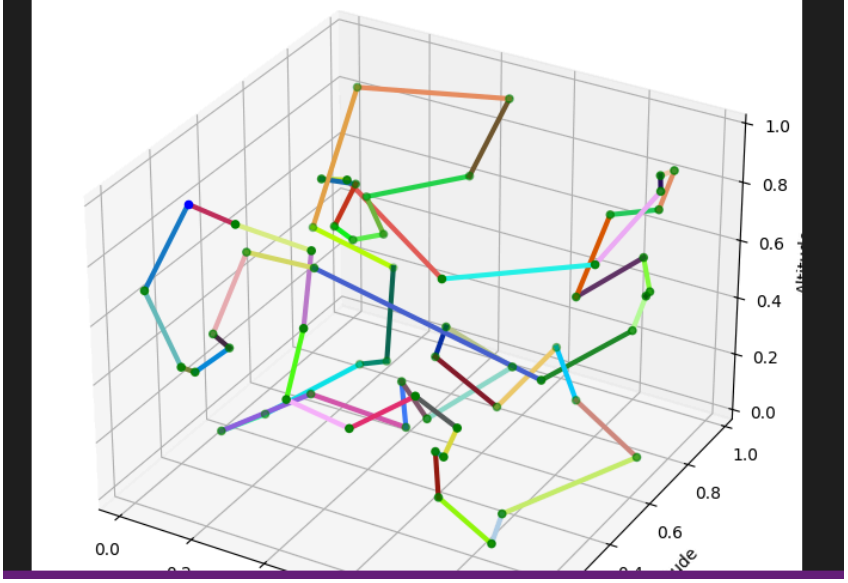
And for last, the Q, with indicate the intensity of the pheromone, between more bigger, it's better

Taking account all this analysis, the best path is when alpha,beta, iterations and Q are a high number and when evaporation rate is low.

```
# model parameters
number_cities = 60
number_ants = 60
number_iterations = 100
alpha = 1.5
beta = 1.5
evaporation_rate = 0.2
Q = 1.5
```

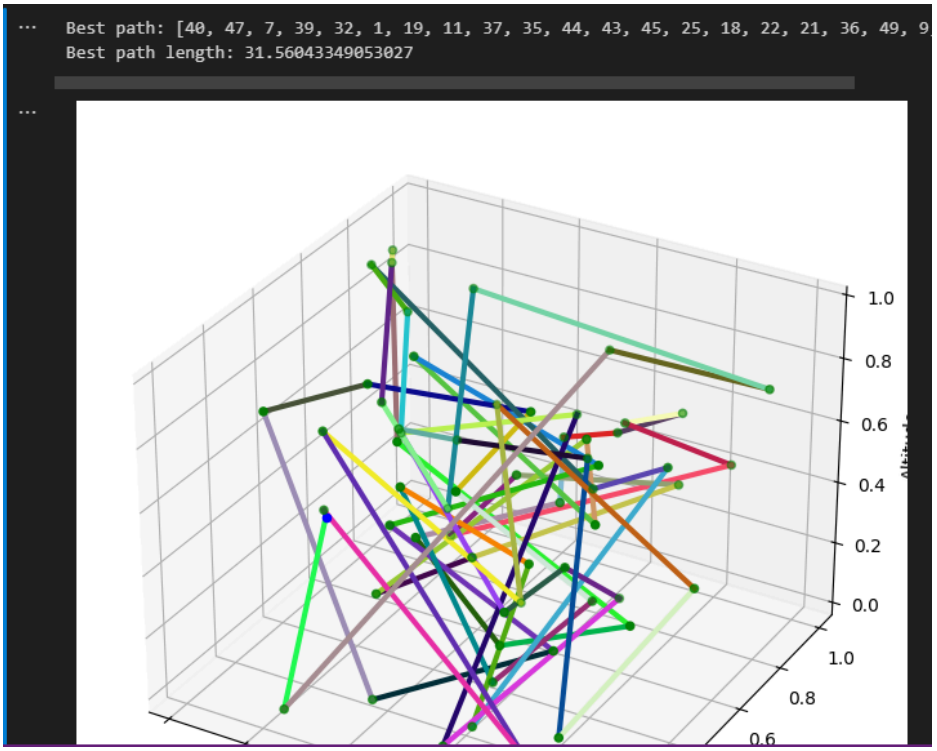
I use those parameters and the result was the next one.

```
Best path: [17, 13, 4, 22, 52, 5, 18, 21, 59, 24, 40, 12, 48, 55, 46, 53, 43, 35, 44, 47,
Best path length: 12.156037850771828
```



The bad thing is the duration compiling, it last 2m and 5seg
And if we do the opposite the result is the next one

```
# model parameters
number_cities = 60
number_ants = 60
number_iterations = 10
alpha = 0.2
beta = 0.2
evaporation_rate = 1
Q = 0.5
```



With those parameters last just 12s, but it is less efficient than the other.
I will show another example of an algorythm efficient