Report homework nr.3 by Mangu Andrei-Daniel and Putanu Alexandru – group E2

1. Problem description

The travelling salesman problem (TSP) asks the following question: "Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city and returns to the origin city?"

2. Algorithm description

1.1 Representation

We chose to respresent the list of cities on an XoY matrix as pair of coordinates, for example ‘4 8616.6667 105016.6667’ corresponds to: city 4 can be found at the coordinates x=8616.6667 and y=105016.6667.

To represent the agents, we chose inversion sequence vectors as in the Lehmer code, such that the first element from the sequence seq corresponds to the number obtained by direct counting in the original set 1,2,3,..,n and for each consecutive element, we eliminate the number from the set, updating it this way.

1.2 Selection

For the selection we chose the fortune wheel, this way the agents with smaller paths have a greater chance to be selected. The fitness function is f(x)=distance(x)/max_distance for each generation.

1.3 Operators used
Crossover and mutation are the operators used in the algorithm. With the chance of 17.5 %, parents are selected from the current generation for crossover. Crossover is with a single cutting point, randomly chosen.

For the mutation, each gene can be changed with a probabilty of 0.3%.

2. Influence of the parameters' values

The parameters we are going to talk about are mutation and crossover probability, number of generations and number of agents from the population.

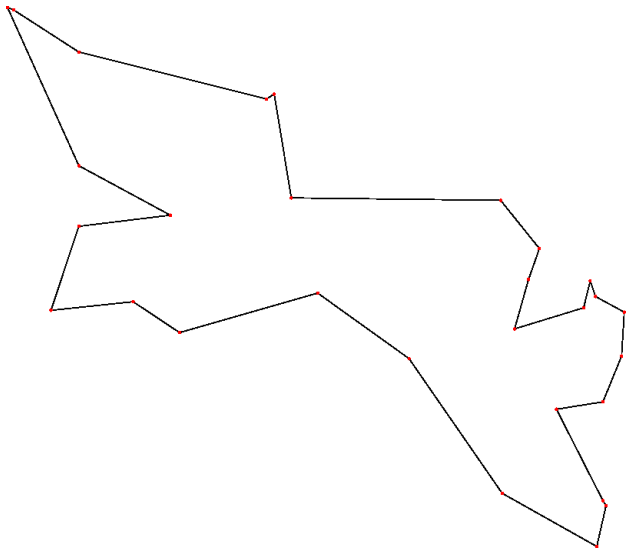
About the mutation probability, we got to the number 0.003 by trying different values and reached the conclusion that for our algorithm, 0.003 is just enough to assure variety in the solutions and small enough not to destroy the results of selection.

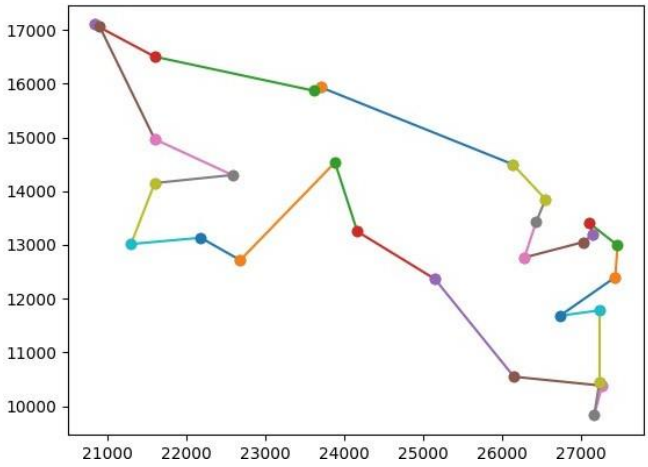
Crossover probability is 0.175 for the same reason, to assure just enough variety in the population. And as for the number of generations and agents, it's very clear that the larger the pool of possible solutions, the better for the result.

3. Algorithm behavior

For these tests, the parameters were: 800 agents, 2000 generations, probability of mutation: 0.003%, probability of crossover: 17,5% .
Here are the results after 30 iterations:

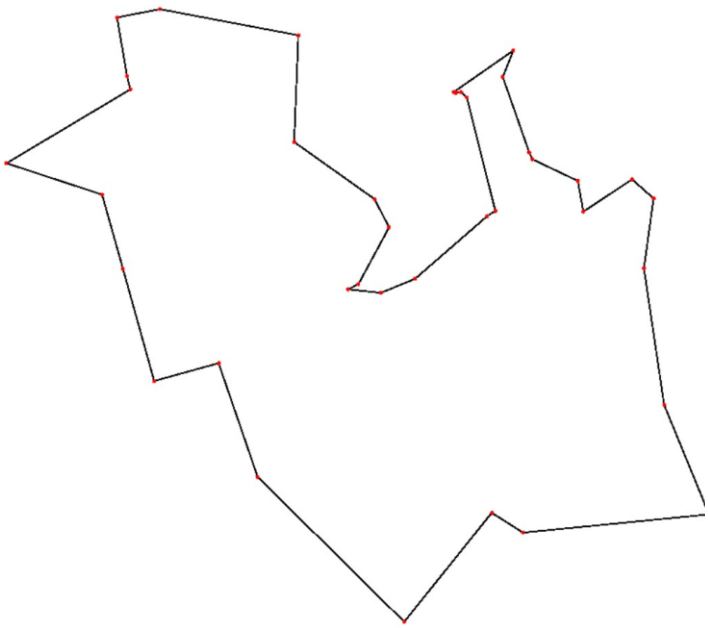
West Sahara:

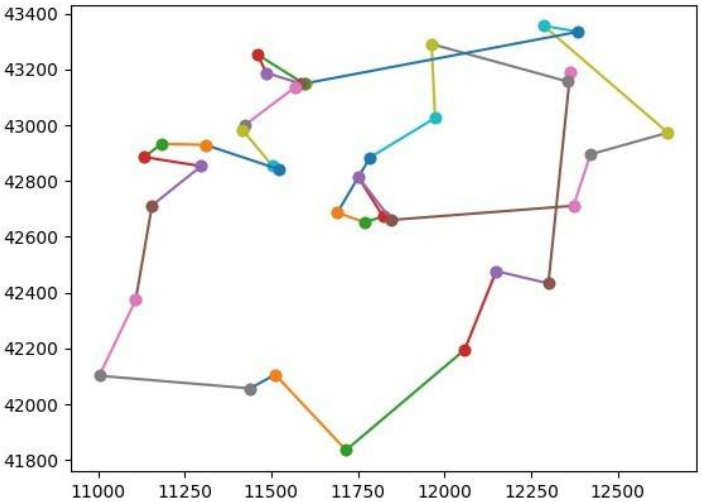




Maximum: 85363
Average: 37915
Best distance: 29499
Number of cities: 29
Optimal value: 27603

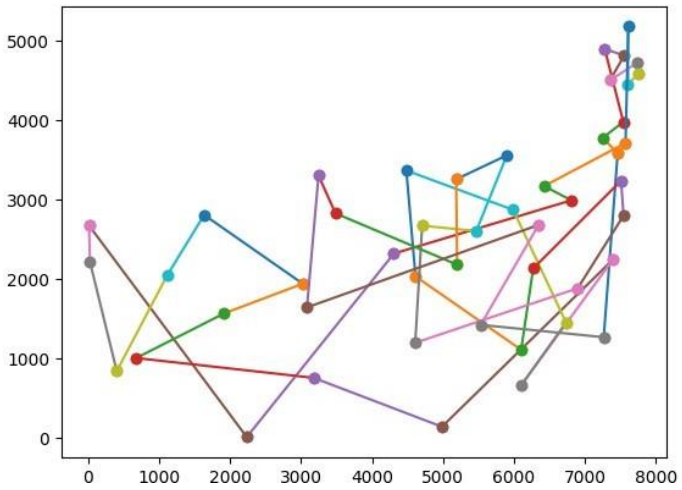
Djibouti:





Maximum : 23052
Average: 11699
Best distance: 8799
Number of cities: 38
Optimal value: 6656

American capitals:



Maximum: 127747
Average : 83686
Best distance: 63170
Number of cities: 48
Optimal value: 33522

4.Conclusion

For a small number of cities, the algorithm succedes in finding a path with a distance close to the optimal, but as we increase the number of cities, the difference between the result and the optimal value increases. Even if the results are not the best, the implementation suits the problem and uses the genetic algorithm well.