Activity 4

Optimization with Genetic Algorithms

- **Git Repository**
  https://github.com/YoussefEzz/Genetic-Algorithm

- **Genetic Algorithm Parameters :**
  - **Description of the chromosome**

We have **G** cities(nodes) indexed in TSPLIB as **0,1,2,…,G-1** , each two cities are connected with some distance(edge weights) represented as a matrix **G * G.**

One of the goals in TSP(Travel salesman problem) is to traverse all cities under the condition of starting and ending up in the same city.

Assume we have 5 cities (0 to 4) and the salesman starts at city 0 and takes the path: 0 -> 1 -> 2 -> 3 -> 4 -> 0 then the search space is all permutations of the cities that start and end with the same start city and the chromosome is a subset of that search space

So for G cities and start **city 0** the chromosome sample is and it's length is **G+1**:

0	1	2	…	G - 1	0

  - **Fitness Function**

The goal is to find the minimum path that traverses all cities.so, to evaluate how close the chromosome is to this goal the fitness function is to calculate the sum of distances between every two successive cities in the chromosome

0	1	2	…	G - 1	0

Cost = Weight[0][1] + Weight[1][2] + … + Weight[G-1][0]

The smaller the cost the more fit is the chromosome so **fitness = 100 / cost**

- ### Partially Mapped Crossover Operator(PMX)[1]

After choosing two random cut points on parents to build offspring, the portion between cut points, one parent's string is mapped onto the other parent's string and the remaining information is exchanged.

Consider, for example, the two parents' tours with randomly one cut point between 3rd and 4th bits and other cut point between 6th and 7th bits are as follows (the two cut points marked with "|")

Parent 1	3	4	8	2	7	1	6	5
Parent 2	4	2	5	1	6	8	3	7

The mapping sections are between the cut points. In this example, the mapping systems are:
2 <-> 1, 7 <-> 6 and 1 <-> 8. Now two mapping sections are copied with each other to make offspring as follows:

Child 1	X	X	X	1	6	8	X	X
Child 2	X	X	X	2	7	1	X	X

Then we can fill further bits (from the original parents), for those which have no conflict as follows:

Child 1	3	4	X	1	6	8	X	5
Child 2	4	X	5	2	7	1	3	X

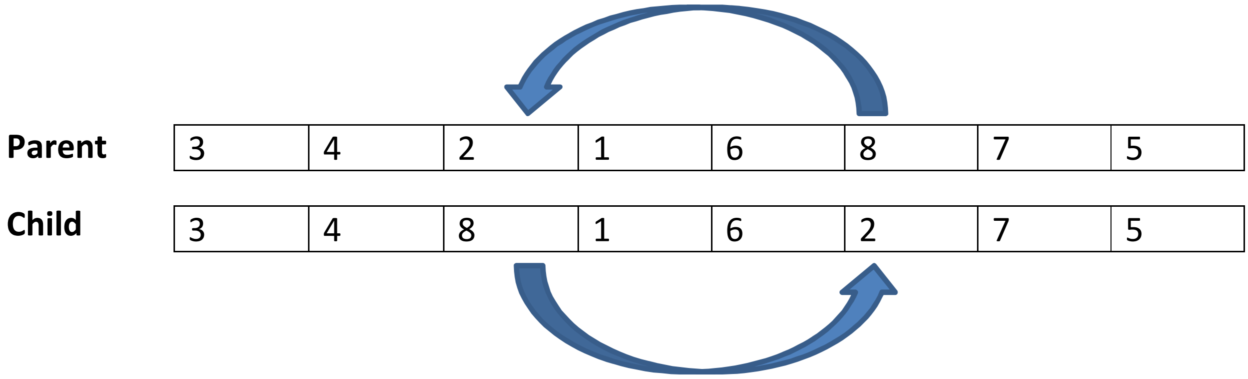
Hence, the first in the first offspring is 8 which comes from first parent but 8 is already in this offspring, so we check mapping 1 ←→ 8 and see again 1 existing in this offspring, again check mapping 2 ←→ 1 , so 2 occupies at first ×. Similarly, the second × in first offspring is 6 which comes from first parent but 6 exists in this offspring; we check mapping 7 ←→ 6 as well, so 7 occupies at second . Thus the offspring 1 is

[1] https://www.hindawi.com/journals/cin/2017/7430125/

Child 1	3	4	2	1	6	8	7	5
Child 2	4	8	5	2	7	1	3	6

- **Swap mutation operator**

  Two random cutpoints are selected were the cities are swapped



Parent	3	4	2	1	6	8	7	5
Child	3	4	8	1	6	2	7	5

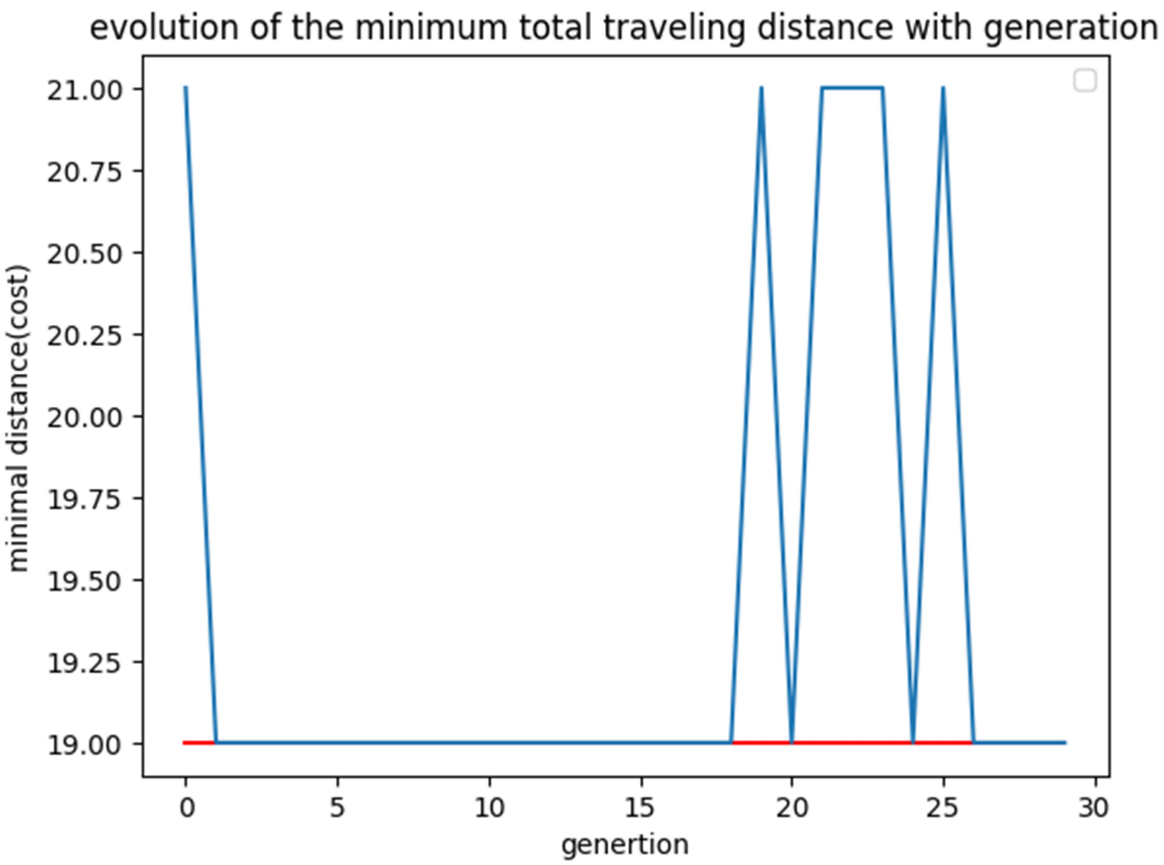
- **The results of executing the code for 3 problems of different sizes**
  - **problem with less than 10 cities** :

    Dataset : **five_d.txt** is a set of 5 cities**.** The True minimal tour has length 19.

    URL : https://people.sc.fsu.edu/~jburkardt/datasets/tsp/five_d.txt

Population Size	**Number of generations**	**True minimal cost**	**Actual minimal cost**
4	4	19	19
4	10		19
8	14		19
8	20		19
16	24		19
16	30		19

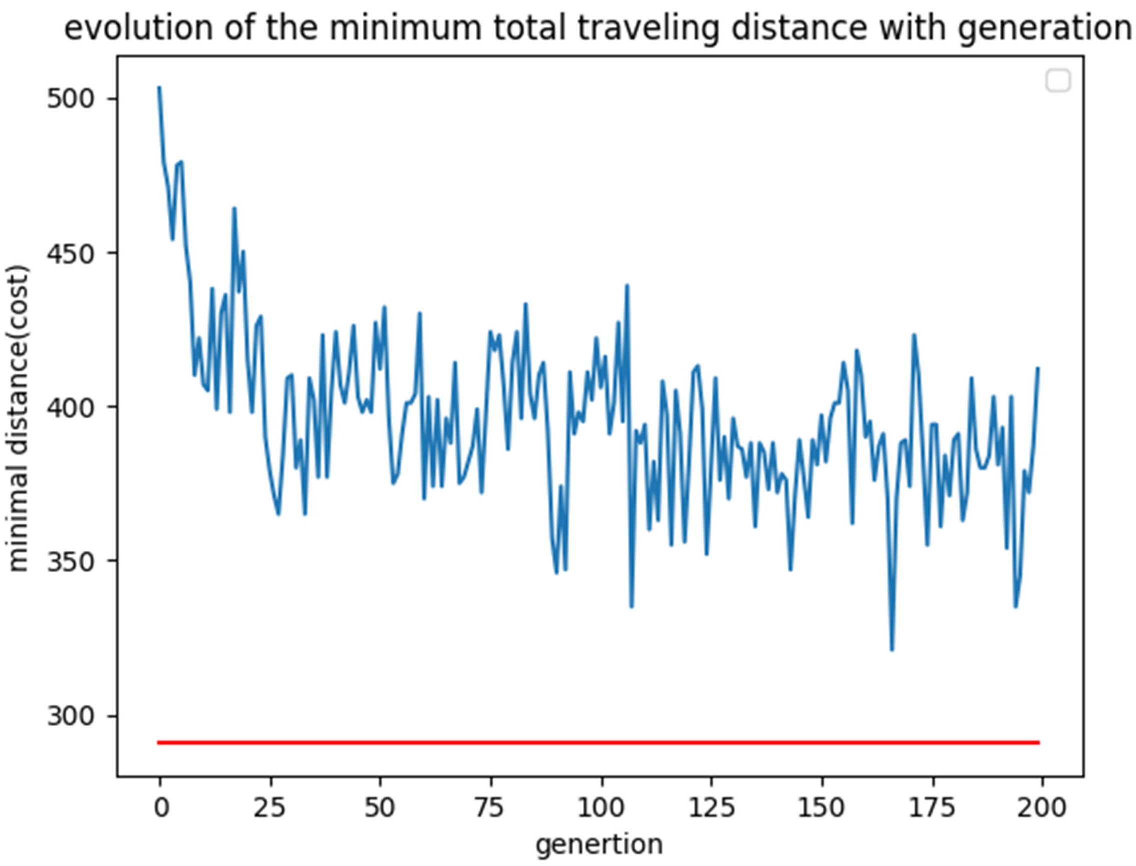
The resulting actual minimal cost is the same as the true for all tested population sizes and number of generations and the system reaches a stationary state early



- **problem with 10-30 cities**:
  Dataset : **p01.tsp** is a set of 15 cities. It is NOT from TSPLIB. The minimal cost is 291.
  URL : https://people.sc.fsu.edu/~jburkardt/datasets/tsp/p01.tsp

Population Size	**Number of generations**	**True minimal cost**	**Range of Actual minimal cost**
10	10	291	364 - 512
50	50		433 - 444
100	100		406 - 419
100	150		384 - 433
150	200		369 - 391
200	200		380 - 385

The resulting actual minimal cost is close but not equal to true minimal cost for all tested population sizes and number of generations and the system reaches a stationary state as population size approaches 200

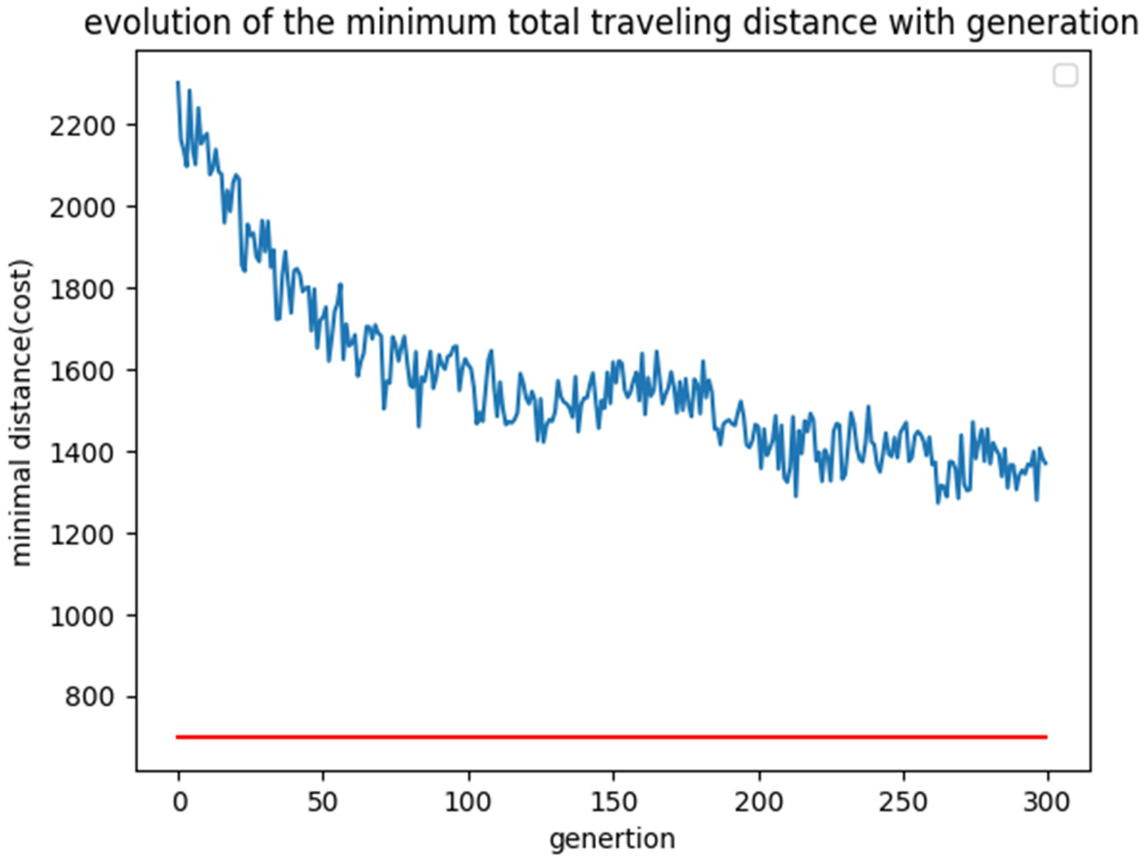


- **problem with more than 30 cities**:

  Dataset : DANTZIG42 is a set of 42 cities, from TSPLIB. The minimal tour has length 699.

  URL : https://people.sc.fsu.edu/~jburkardt/datasets/tsp/dantzig42_d.txt

Population Size	**Number of generations**	**True minimal cost**	**Range of Actual minimal cost**
50	50	699	1703 - 1730
100	100		1464 - 1589
150	150		1523 - 1581
200	200		1511 - 1628
250	250		1478 - 1526
400	300		1373 - 1449


evolution of the minimum total traveling distance with generation
minimal distance(cost)
2200
2000
1800
1600
1400
1200
1000
800
0
50
100
150
200
250
300
genertion