	MÜHENDİSLİK FAKÜLTESİ, Bilgisayar Mühendisliği Bölümü 2021– 2022 **"Bilgisayar Mühendisliği / BLM19447E - Introduction to Artificial Intelligence "**

BLM19447E - Introduction to Artificial Intelligence

HW#2

(Due to 2-12-2021)

(Report)

Name	Aycha yahia
ID	1621221034

Abstract:

Travelling Salesman Problem (TSP), which aims to find a minimum-distance path between a set of cities that a salesman might follow on a business trip. The best path is the path that has the shortest total round-trip distance--from an arbitrary starting city, though all other cities, and back to the starting city.

Genetic algorithm:

Population size =100 path initialized randomly using the cityName list was extracted from the database.

Fitness Function: It is defined as the total distance of travel represented by the

order of cities. To calculate the distance between the cities I used the Euclidean distance formula, which gives the distance between two points (or) the straight line distance. Let us assume that (x1,y1)(x1,y1) and (x2,y2)(x2,y2) are two points in a two-dimensional plane as shown in **figure 1**:[1]

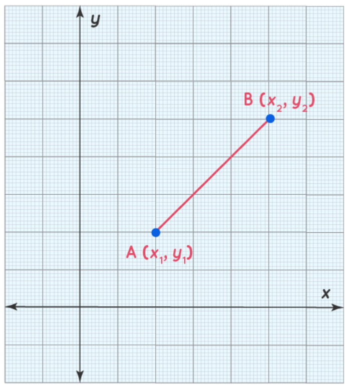

Figure 1 two points in a two-dimensional plane

The Euclidean distance formula says:

$$d = \sqrt{(x_{22} - x_{11}) + (y_{22} - x_{11})} \text{ where,} \qquad \textbf{(eq 1)}$$

- (x_{11}, y_{11}) are the coordinates of one point.
- (x_{22}, y_{22}) are the coordinates of the other point.

[1] https://www.cuemath.com/euclidean-distance-formula/

- d is the distance between (x_{11}, y_{11}) and (x_{22}, y_{22})

Crossover Operator: list slicing was applied as cross over. After selecting 2 parents randomly start and end indices are selected randomly and slicing is applied over the both parent genes. As shown in **figure 2**. After the cross over repeated cities are found and replaced with missing cities in every path.

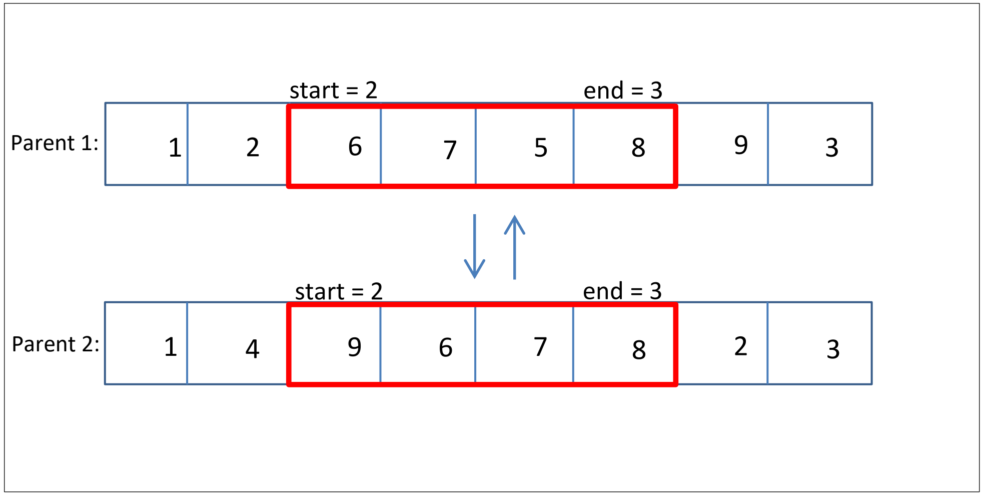

Figure 2

Mutation Operator: The inversion mutation was applied over non child genes. Inversion mutation takes randomly selected path from the population. Then slice with randomly start and end indices selected to be inverted. As shown in **figure 3**:

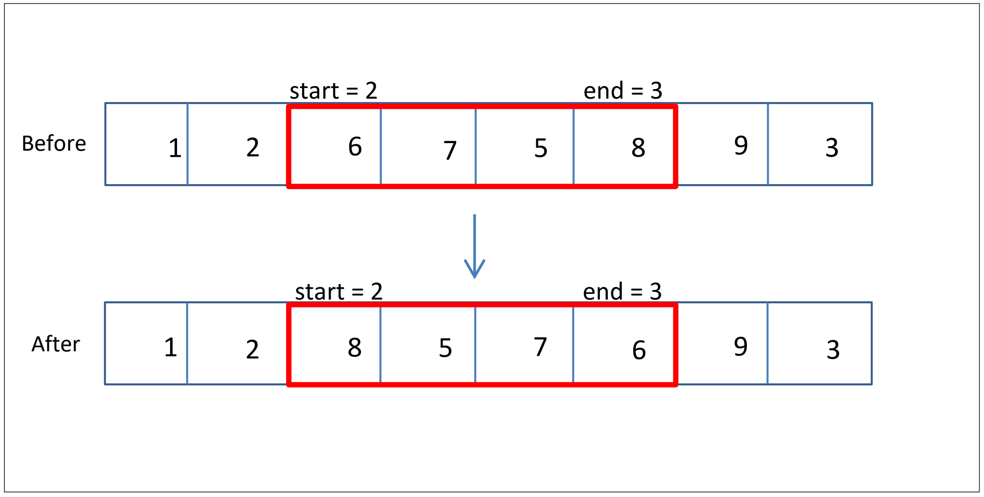

Figure 3

Results:

Figure 4 shows the min cost over generations. The genetic algorithm shows unstable performance between generation 0-1000 with min cost value = 2760.

After 1000 generation:

```
best path:  ['52', '72', '74', '97', '58', '8', '41', '59', '64', '60', '29', '1', '43', '71', '101', '96', '93', '38', '11', '7', '62', '32', '10', '65', '77', '26', '48', '47', '31', '20', '66', '9', '21', '44', '89', '12', '73', '27', '45', '17', '86', '57', '61', '91', '5', '69', '40', '54', '24', '23', '83', '14', '53', '94', '42', '75', '18', '25', '90', '63', '46', '16', '33', '78', '49', '82', '35', '3', '68', '92', '95', '100', '6', '36', '19', '76', '50', '2', '22', '67', '34', '99', '56', '15', '81', '98', '88', '30', '4', '87', '80', '13', '70', '79', '51', '39', '84', '85', '37', '28', '55']  cost:  2760
```

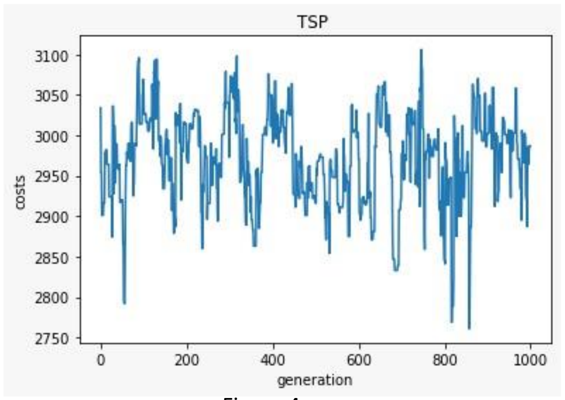

Figure 4

Figure 5 shows the min cost over generations. The genetic algorithm continues showing unstable performance between generations 0-2000 with min cost value 2698.

After 2000 generation:

```
best path:  ['52', '72', '74', '97', '58', '8', '41', '59', '64', '60', '29', '1', '43', '71', '101', '96', '93', '38', '11', '7', '62', '32', '10', '65', '77', '26', '48', '47', '31', '20', '66', '9', '21', '44', '89', '12', '73', '27', '45', '17', '86', '57', '61', '91', '5', '69', '40', '54', '24', '23', '83', '14', '53', '94', '42', '75', '18', '25', '90', '63', '46', '16', '33', '78', '49', '82', '35', '3', '68', '92', '95', '100', '6', '36', '19', '76', '50', '2', '22', '67', '34', '99', '56', '15', '81', '98', '88', '30', '4', '87', '80', '13', '70', '79', '51', '39', '84', '85', '37', '28', '55']  cost:  2698
```

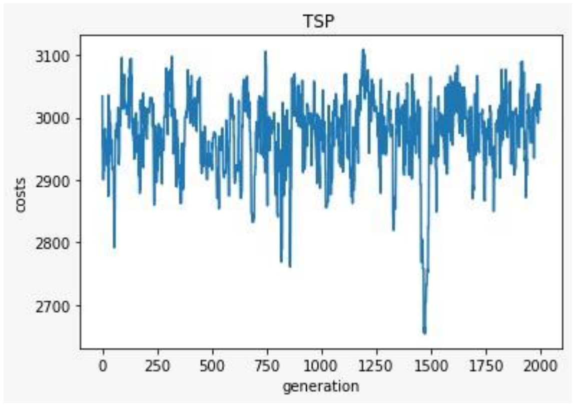

Figure 5

After 2000 until 5000 generations no change noticed on the behavior of the algorithm. As shown on the figure the min cost appears on generation 3800 with min cost value = 2698 equals to value found in generation 1500 as shown in **figure 6**. I recommend the reason of unstable behavior shown on the graphs is because of the randomness of the genetic algorithm so we cannot grantee continually increasing or decreasing performance curve.

best path: ['52', '72', '74', '97', '58', '8', '41', '59', '64', '60', '29', '1', '43', '71', '101', '96', '93', '38', '11', '7', '62', '32', '10', '65', '77', '26', '48', '47', '31', '20', '66', '9', '21', '44', '89', '12', '73', '27', '45', '17', '86', '57', '61', '91', '5', '69', '40', '54', '24', '23', '83', '14', '53', '94', '42', '75', '18', '25', '90', '63', '46', '16', '33', '78', '49', '82', '35', '3', '68', '92', '95', '100', '6', '36', '19', '76', '50', '2', '22', '67', '34', '99', '56', '15', '81', '98', '88', '30', '4', '87', '80', '13', '70', '79', '51', '39', '84', '85', '37', '28', '55'] cost: 2698

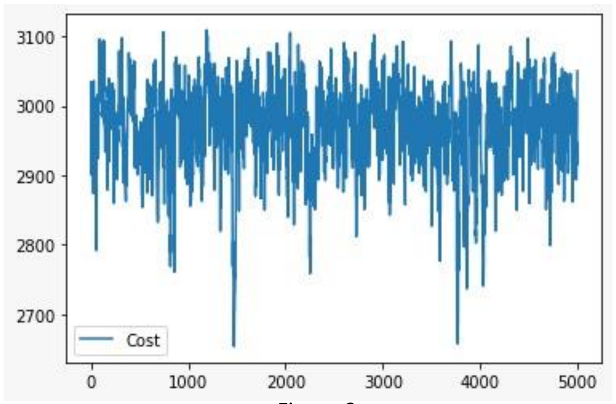

Figure 6