**Traveling Salesman Project Report**

Author: Edwin Piątek   
Index number: 97253  
Group: SD3

16/01/2025

### Table of contents:

[**Table of contents: 2**](#_a5cllaumd2k0)

[**Description: 2**](#_yu7uy86bgcsz)

[**Parameter Testing 3**](#_hl2hmuj0nn41)

[Test 1 3](#_w8bbzvky1bds)

[Test 1 Result 3](#_q2mmwpbpol8q)

[Test 2 3](#_9l6dltlb2hvz)

[Test 2 Result 3](#_dtx2j8nxg9df)

[Test 3 4](#_o784x8beraye)

[Test 3 Result 4](#_3dlpswdutjj8)

[Test Conclusion 4](#_urmco6moyvw9)

[**Algorithm Comparison 5**](#_a4xf99q3oclv)

[**Best Solutions 5**](#_4snfd7c1c05n)

[**Conclusion 6**](#_px0n7snnm1z)

### Description:

Algorithm used in my code is using tournament selection, inverted mutation and ordered crossover to solve the Traveling Salesman Project (TSP). All of those operations are based on initial population that is determined by generating “n” (represents the size of population specified by the user) amount of random solutions and adding them to the chosen way of storing solutions throughout the whole program which is python list.

### 

### Parameter Testing

#### Test 1

* Size of initial population: 10
* Mutation probability: 3%
* Selection type: Tournament
* Tournament size: 3
* Greedy status: Off

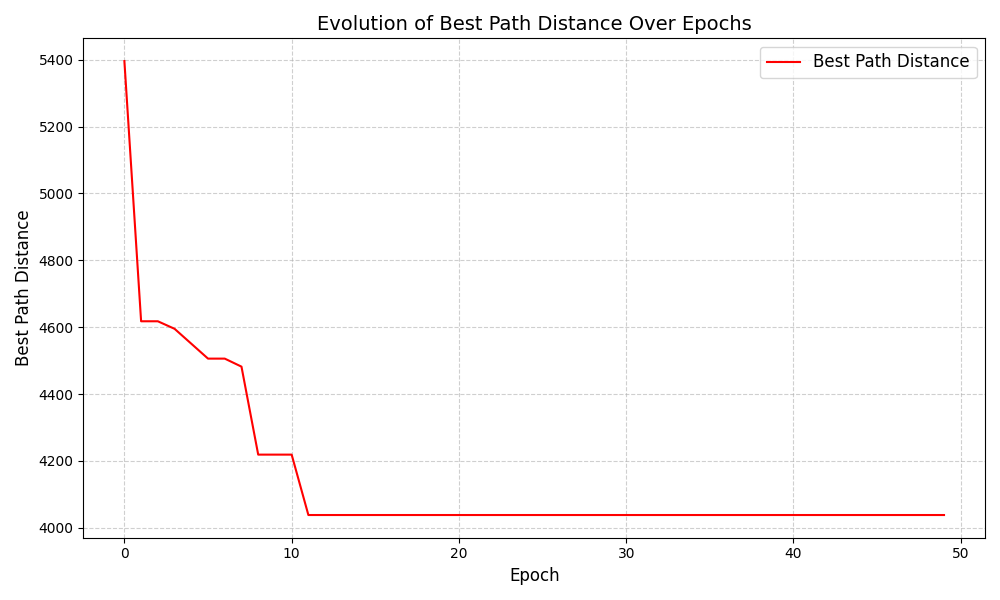
##### Test 1 Result

##### 

#### Test 2

* Size of initial population: 20
* Mutation probability: 4%
* Selection type: Tournament
* Tournament size: 4
* Greedy status: Off

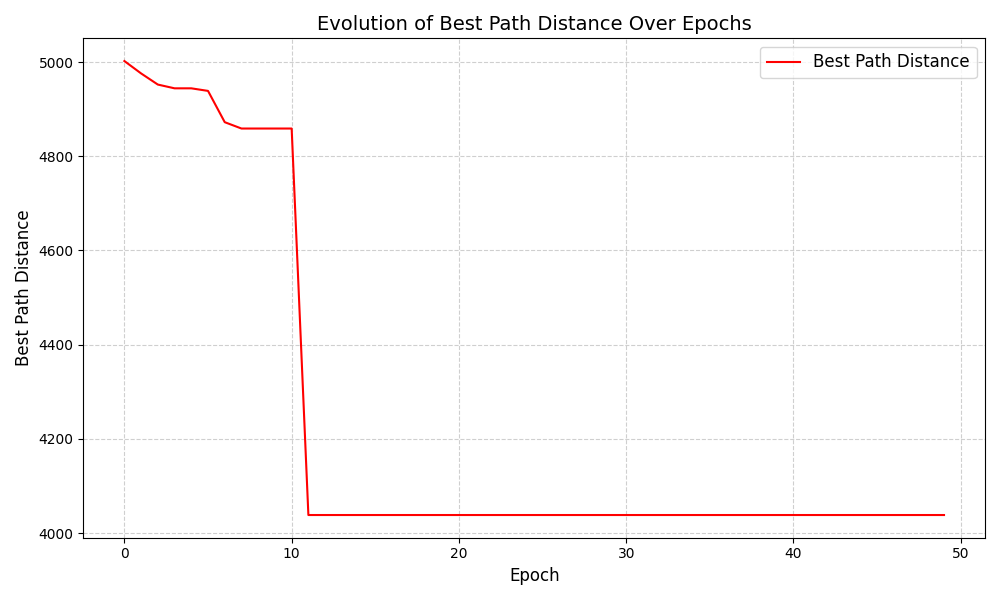
##### Test 2 Result



#### Test 3

* Size of initial population: 30
* Mutation probability: 5%
* Selection type: Tournament
* Tournament size: 7
* Greedy status: Off

##### Test 3 Result



#### Test Conclusion

**Initial Population Size:**

* Increasing the population size leads to a more diverse search space, resulting in better convergence and lower final distances.
* For example, a population size of 30 achieved the best convergence, as seen in Test 3, but it may require more computational time.

**Mutation Probability:**

* A mutation probability of 0.03 (as in Test 1) resulted in a good balance between diversity and convergence.
* Higher mutation probabilities (e.g., 0.05 in Test 3) maintain diversity but might disrupt convergence if set too high.

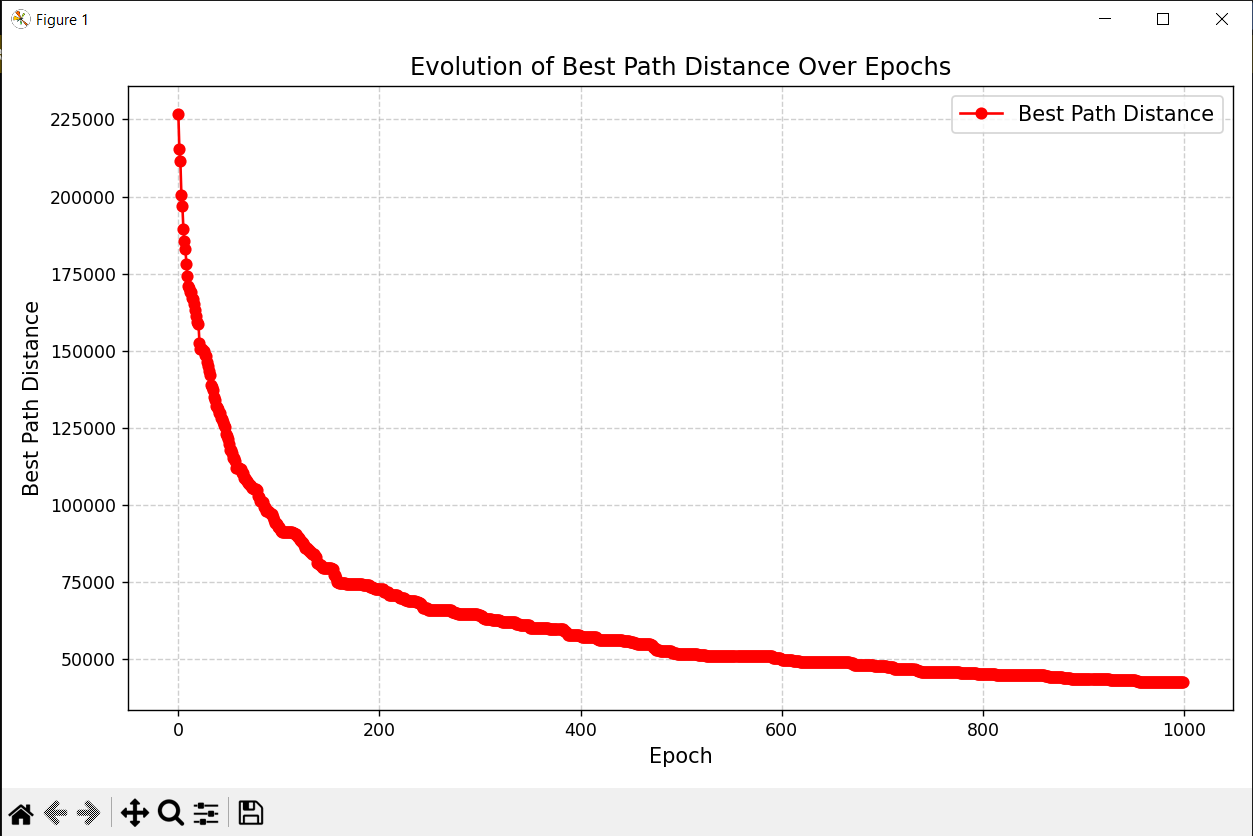
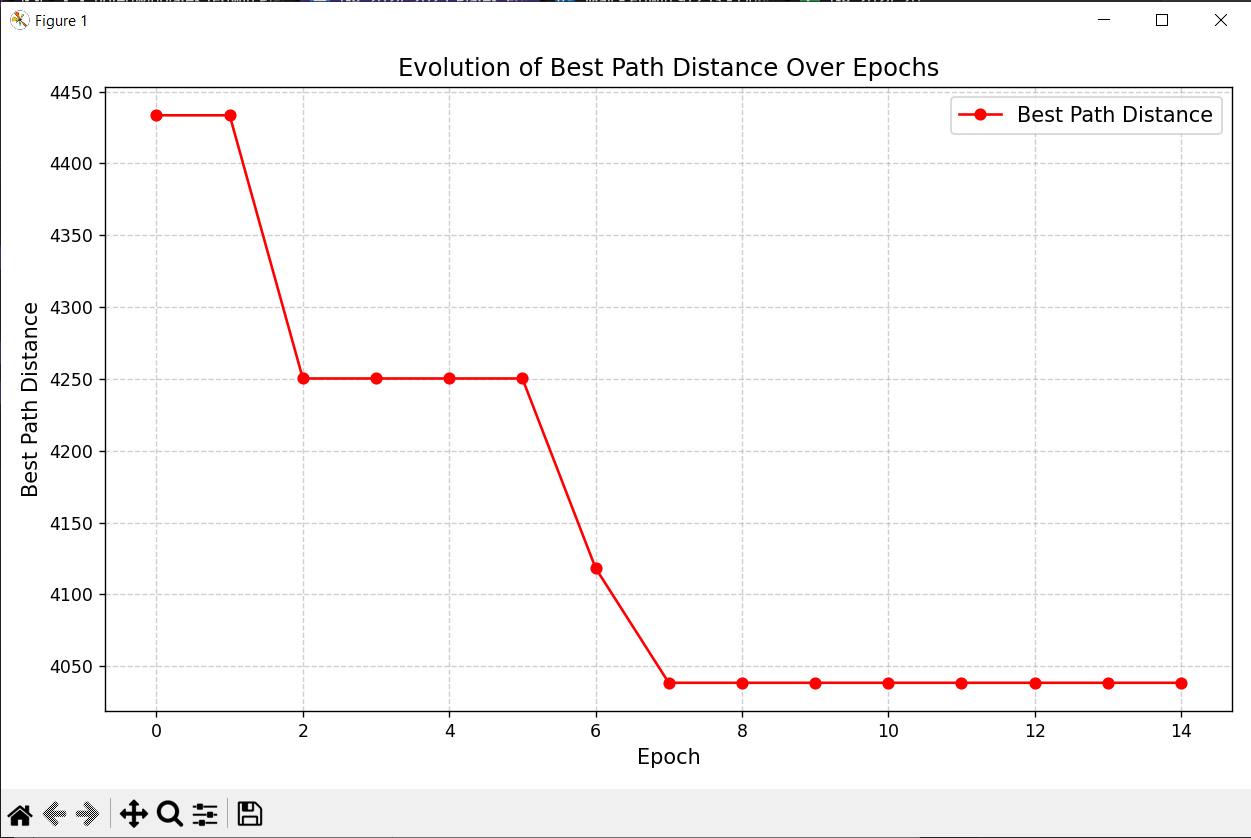
**Tournament Size:**

* Larger tournament sizes (e.g., 7 in Test 3) focus more on selecting the best-performing individuals, which speeds up convergence but risks reducing diversity.
* Smaller tournament sizes (e.g., 3 in Test 1) promote diversity but may slow convergence.

### Algorithm Comparison

| **Algorithm** | **Run results** | **Best Result** | **Average** | **Standard Deviation** | **Variance** |
| --- | --- | --- | --- | --- | --- |
| Genetic Algorithm | 9225.29, 8979.02, 9427.59, 8732.17, 9390.40, 9440.41, 9747.11, 9487.10, 9283.50, 8953.57 | 8732.17 | 9266.62 | 286.23 | 81926.18 |
| Random Algorithm | 28466.08, 30484.49, 32040.67, 32461.15, 31230.04, 29866.03, 26804.54, 29334.89, 30358.30, 30542.15 | 26804.54 | 30158.84 | 1586.30 | 2516355.17 |
| Greedy Algorithm | 10094.39, 9252.40, 9575.16, 9575.16, 9506.05, 10299.37, 9396.88, 9140.13, 9140.13, 9253.75, 9260.70 | 9140.13 | 9535.40 | 362.00 | 131045.97 |

### Best Solutions



File name: Berlin 11 File name: kroA150

Population size: 100 Population size: 100

Number of epochs: 15 Number of epochs: 1000

Mutation probability: 3% Mutation probability: 6%

Tournament size: 5 Tournament size: 50

Final score: 4038.44 Final score: 42609.66

Run time: 00m:02s:52ms Run time: 05m:19s:46ms

### 

### Conclusion

In this project, I investigated the efficacy of the genetic algorithm in optimizing solutions for the Traveling Salesman Problem (TSP). By systematically analyzing various parameters, including population size, mutation probability, and tournament size, I observed that increasing the population size improved solution diversity, while larger tournament sizes facilitated faster convergence. However, it became apparent that maintaining a balance between these factors is essential to avoid overemphasizing either diversity or convergence.

A comparative analysis of the genetic algorithm against random and greedy methods revealed its superior performance, yielding the lowest average distances while maintaining a manageable runtime. This supports the algorithm’s applicability in solving complex optimization problems. Based on the findings, I concluded that a mutation probability of 3% and moderate tournament sizes provided the optimal trade-off between computational performance and cost.