TSP Genetic Algorithm Report

Name: David Shaulov

ID: 317005403

Chromosome representation:

- I chose to represent the route as a list of 48 cities, for example $[3,42, \dots, 8]$, where the route is $3 \to 42 \to \cdots \to 8 \to 3$

Fitness function implementation:

- The fitness function measures the total distance traveled of the route, and returns $\frac{1}{total\_distance}$, so that the lower the distance the higher the fitness.

Selection type:

- I have implemented both roulette and elitism selection.

Crossover types:

- Since regular crossover does not work well for the representation I have chosen, I implemented a different kind of crossover:
  - Two random points are chosen $point_A, point_B$
  - $parent_A[point_A : point_B]$ is copied to the child.
  - Every city that was not copied from $parent_A$ is taken from $parent_B$ in order.
  - Same for the second child but in reverse.
- This crossover ensures that cities don't repeat themselves while maintaining some information from parent chromosomes.

Mutation implementation:

- Given mutation rate p, with probability p the mutation function chooses a random city and inserts it at a random index, for example for $[3, 21, 5, 10, 11]$, a mutation might choose 11 and insert it at index 1 and the result will be $[3, 11, 21, 5, 10]$.

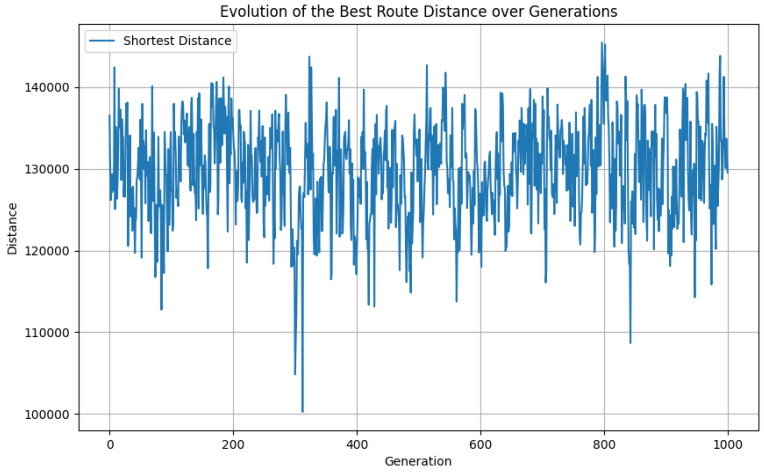
Experiments:

- I have separated the experiments into 2 main categories: roulette selection and elitism selection since they achieve very different results.

Roulette Selection Experiments:

Population Size	Number of Generations	Mutation Rate
50	1000	0.05

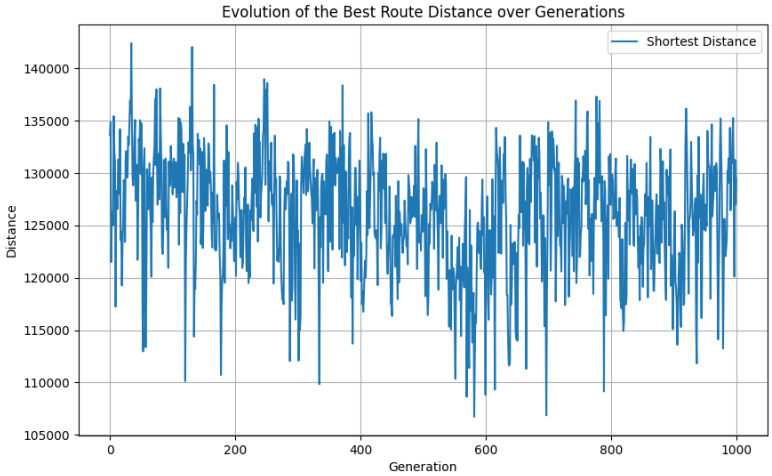
- Result:



  - Best performance is 100,243 at generation 280.

Population Size	Number of Generations	Mutation Rate
100	1000	0.05

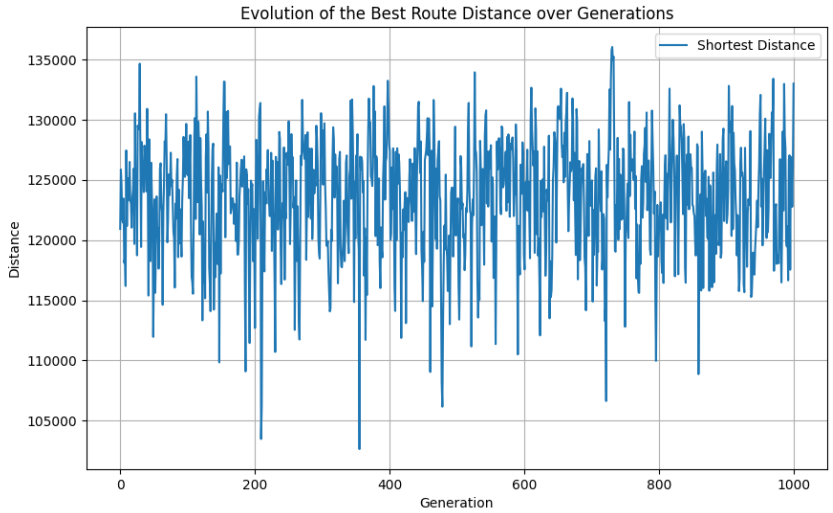
- Result:



  - Best performance is 106,720, at generation 580.

Population Size	Number of Generations	Mutation Rate
200	1000	0.05

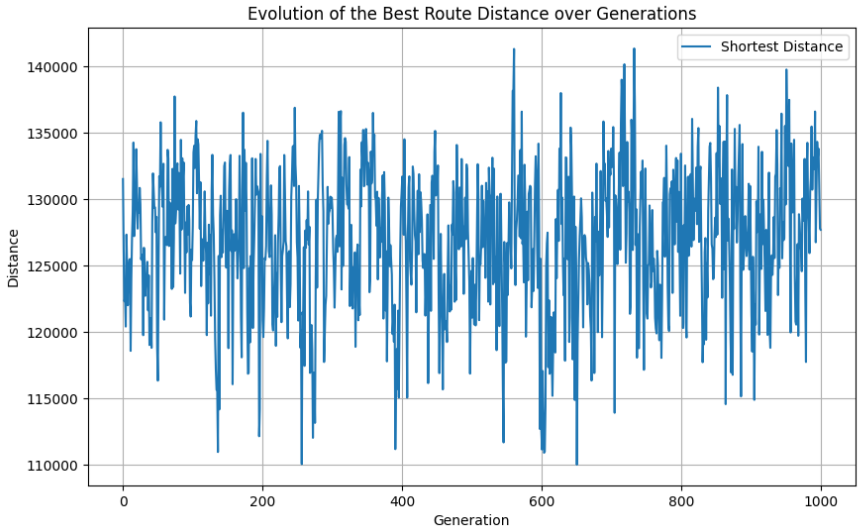
- Result:



  - Best performance is 102,611, at generation 357.

Population Size	Number of Generations	Mutation Rate
100	1000	0.1

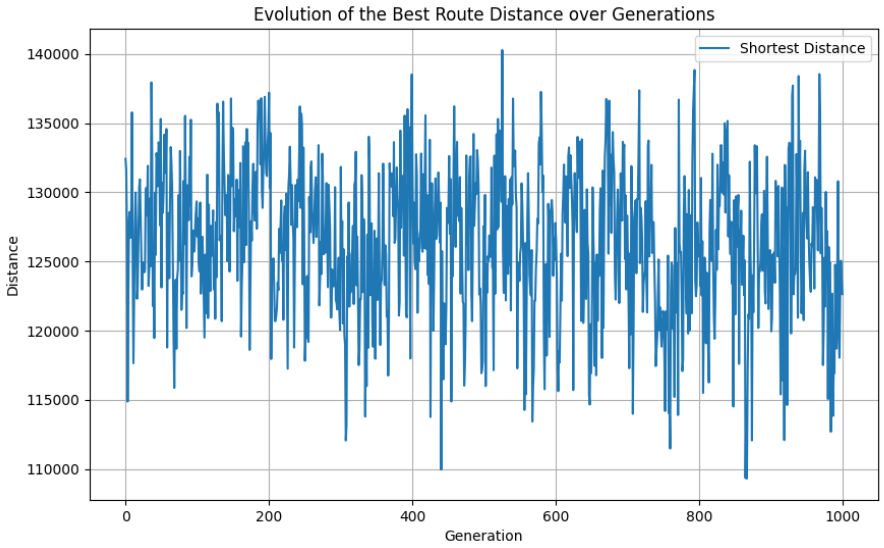
- Result:



  - Best performance is 110,012, at generation 247.

Population Size	Number of Generations	Mutation Rate
100	1000	0.2

- Result:

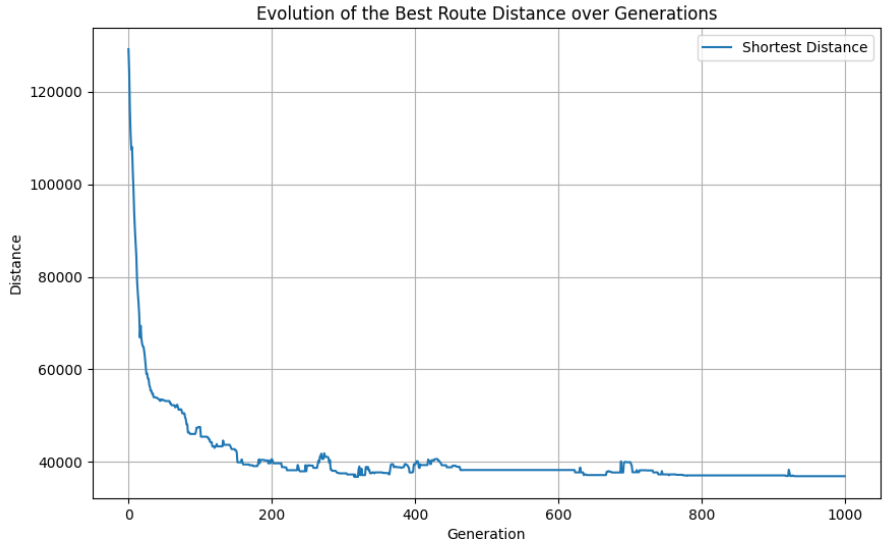


  - Best performance is 109,305, at generation 432.

<u>Elitism Selection Experiments:</u>

Population Size	Number of Generations	Mutation Rate	Elitism Size
100	1000	0.05	5

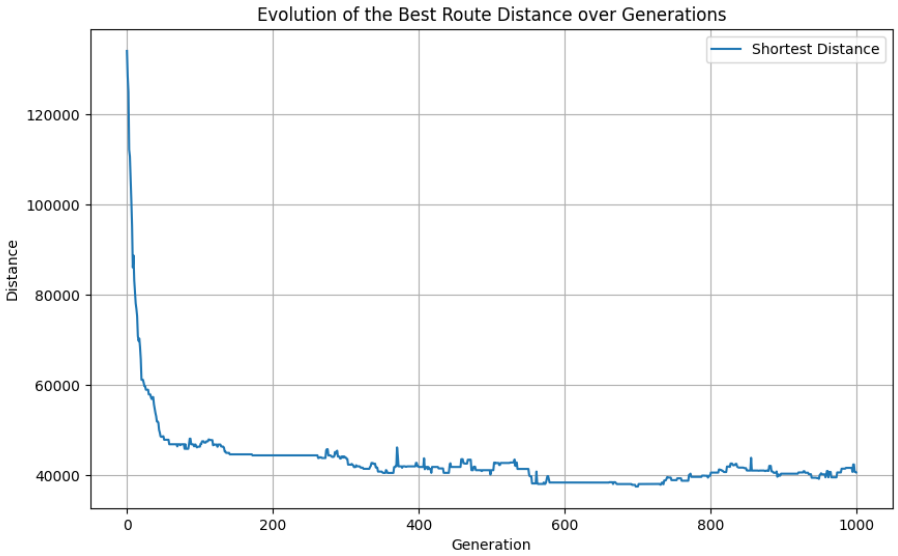
- Result:



  - Best performance is 36,737, at generation 317.

Population Size	Number of Generations	Mutation Rate	Elitism Size
100	1000	0.1	5

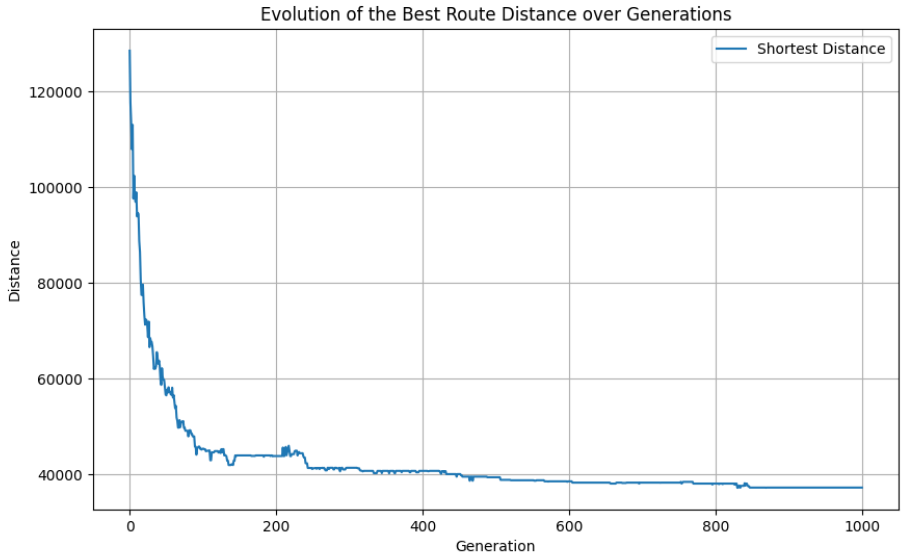
- Result:



  - Best performance is 37,541, at generation 697.

Population Size	Number of Generations	Mutation Rate	Elitism Size
100	1000	0.05	20

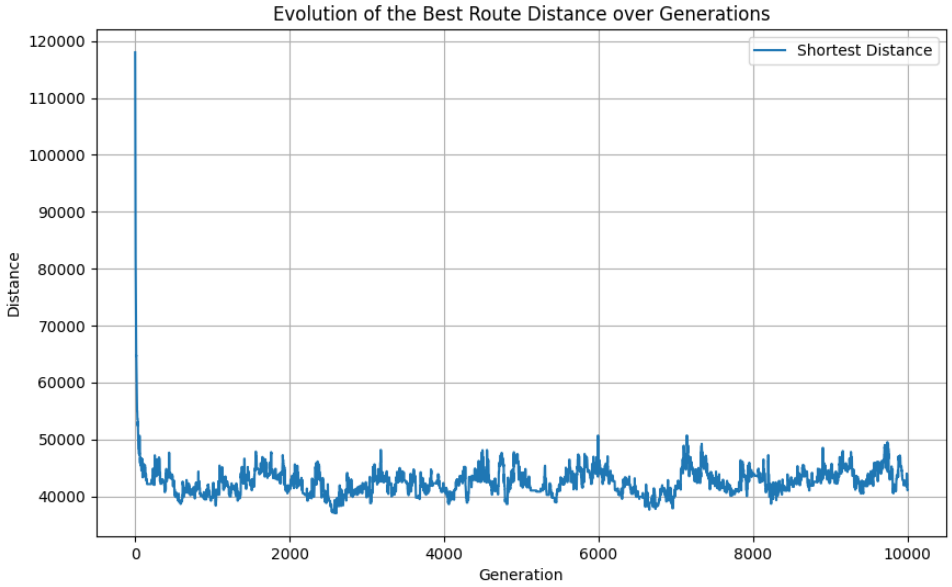
- Result:



  - Best performance is 37,278, at generation 829.

Population Size	Number of Generations	Mutation Rate	Elitism Size
100	10000	0.05	5

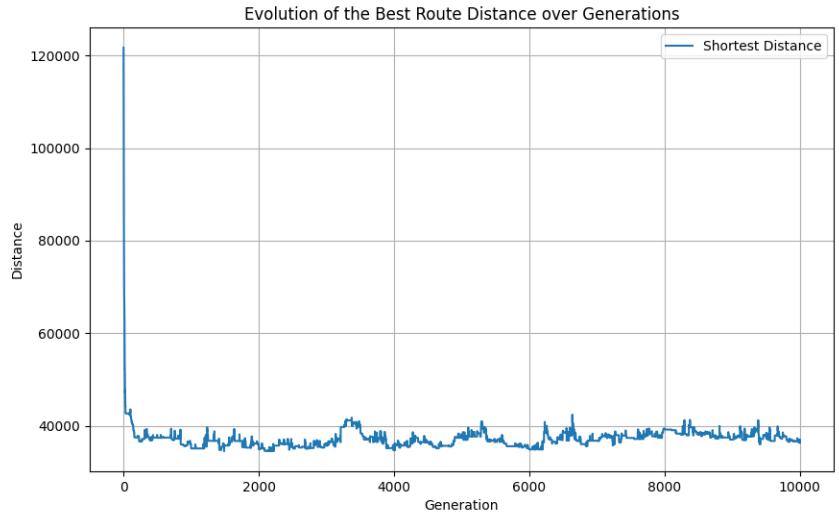
- Result:



  - Best performance is 37,039, at generation 2600.

Population Size	Number of Generations	Mutation Rate	Elitism Size
200	10000	0.05	5

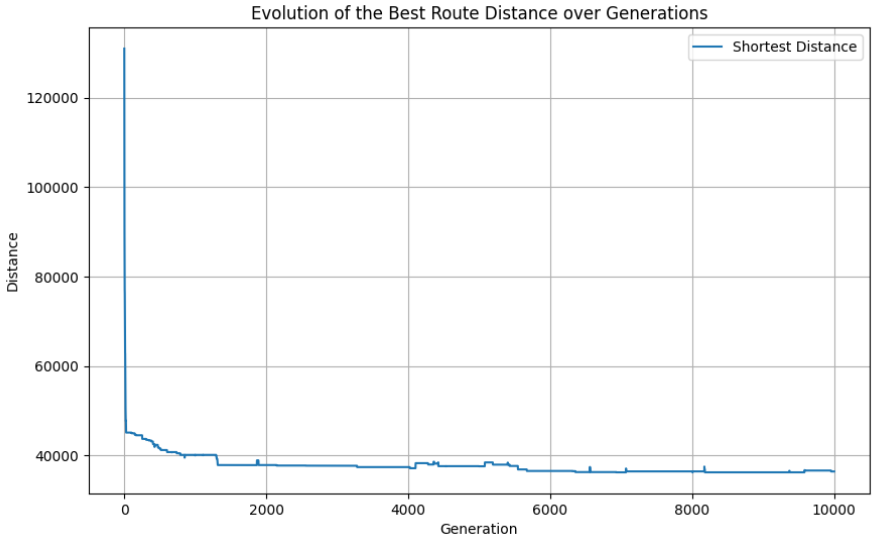
- Result:



  - Best performance is 34,565, at generation 1487.

Population Size	Number of Generations	Mutation Rate	Elitism Size
400	10000	0.05	5

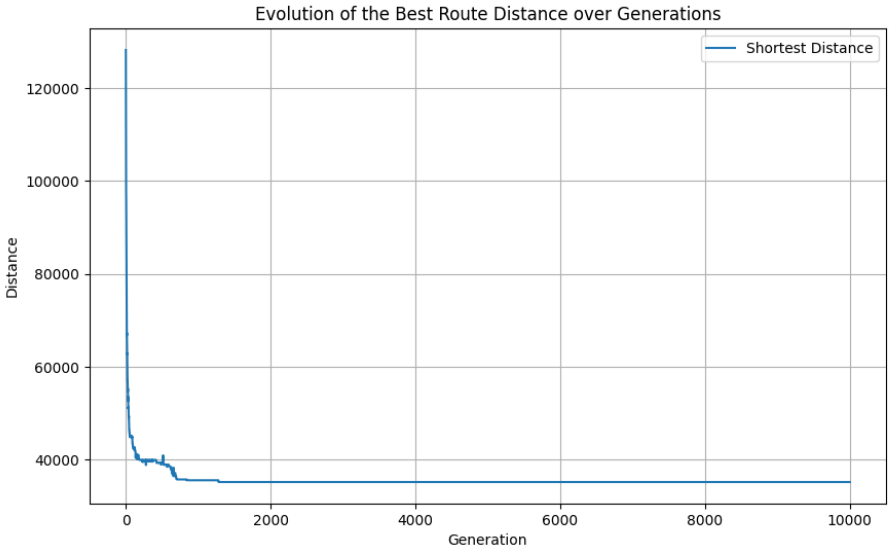
- Result:



  - Best performance is 36,235, at generation 8279.

Population Size	Number of Generations	Mutation Rate	Elitism Size
200	10000	0.05	20

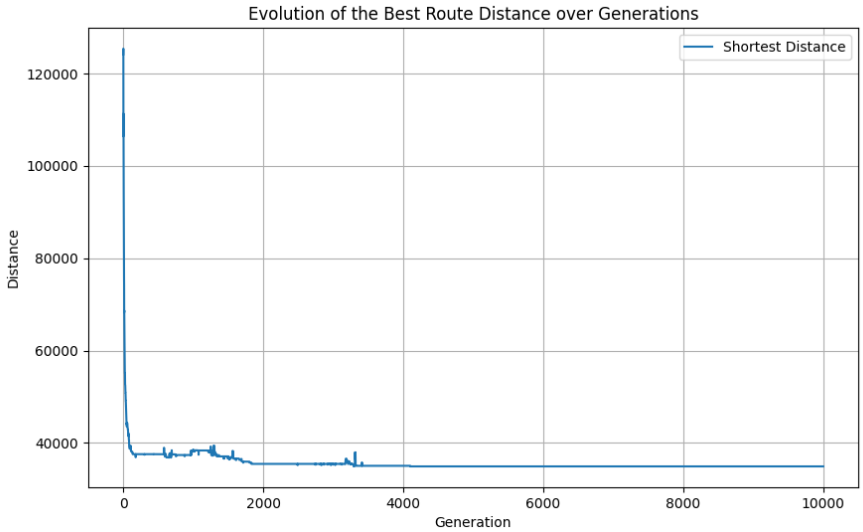
- Result:



  - Best performance is 35,210, at generation 1284.

Population Size	Number of Generations	Mutation Rate	Elitism Size
200	10000	0.1	20

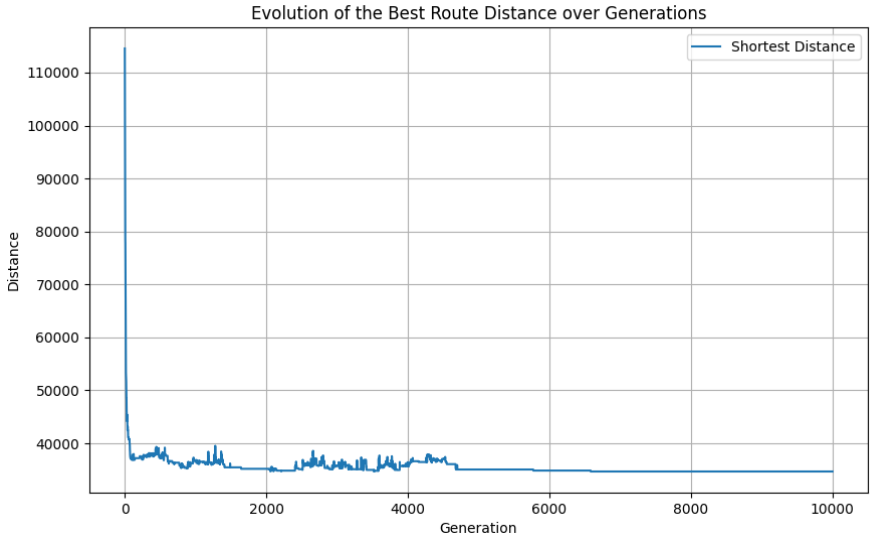
- Result:



  - Best performance is 34,933, at generation 3293.

Population Size	Number of Generations	Mutation Rate	Elitism Size
200	10000	0.075	10

- Result:



  - Best performance is 34,213, at generation 3864.

Comparison with brute force:

- The brute force algorithm achieves a distance of $40{,}526$ –the genetic algorithm outperforms it, but only by $\sim 6{,}000$ units.

Runtimes:

- Experiments with $1{,}000$ generations took about 10 seconds on average.

- Experiments with $10{,}000$ generations took about 180 seconds on average.

Conclusion:

- Elitism selection achieves vastly better results.
- Mutation higher than 0.05 makes it harder to find better routes.
- Routes don't improve much after generation 2000.