By Kamilla

Translating the Traveling Salesman Problem (TSP) into a Genetic Algorithm (GA) involves encoding the solution to the problem as a chromosome and then defining operations such as selection, crossover, and mutation that mimic the process of natural evolution.

Chromosome Representation

In the context of the TSP, a chromosome can be represented as a sequence of numbers, where each number represents a city. The order of the numbers defines the order in which the cities are visited. For example, a chromosome [2, 4, 3, 1] for a TSP with four cities indicates that the path goes from city 2 to city 4 to city 3, ends in city 1, and then returns to city 2.

Selection Techniques

Selection determines which individuals (solutions) from the current generation are used to create the next generation. Two common techniques are:

Roulette Wheel Selection (Fitness Proportionate Selection): Individuals are selected based on their fitness levels. The better the fitness, the higher the chance of being selected. However, this method can lead to premature convergence if high-fitness individuals dominate early.

Tournament Selection: A set number of individuals are chosen at random, and the best among these is selected. This process is repeated until the desired number of individuals is selected. It helps maintain diversity but might require careful tuning of the tournament size.

Crossover Techniques

Crossover combines two parents to create offspring for the next generation. Two common techniques are:

Ordered Crossover (OX1): A subsection of one parent's route is copied to the child, and the remaining cities are filled in the order they appear in the other parent, avoiding duplicates. It's good for preserving relative order.

Partially Mapped Crossover (PMX): Two crossover points are selected, and the sections between them are swapped between the parents. Mapping is used to resolve conflicts, ensuring that each city appears only once. This helps preserve the absolute position of certain cities.

Mutation Techniques

Mutation introduces small random changes in the offspring to maintain diversity. Two common techniques are:

Swap Mutation: Two cities in the route are chosen randomly and swapped. It's simple and effective, but large-scale mutations can disrupt good solutions.

Inversion Mutation: A subsection of the route is selected and its order is reversed. This can be more disruptive than swap mutation but might help escape local optima.

Elitism

Elitism involves copying a small proportion of the fittest individuals into the next generation unchanged. It ensures that the fitness of the best individual never decreases but can lead to premature convergence if overused.

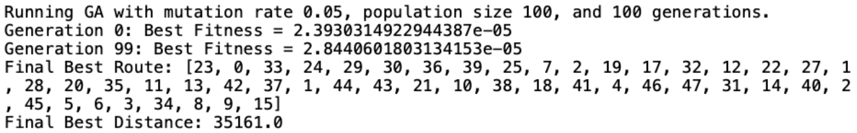
Population Size

The population size should be large enough to provide diversity but not so large as to make the computation infeasible. A common practice is to experiment with different sizes and choose one that balances the exploration of the search space (diversity) and the exploitation of good solutions (convergence speed).

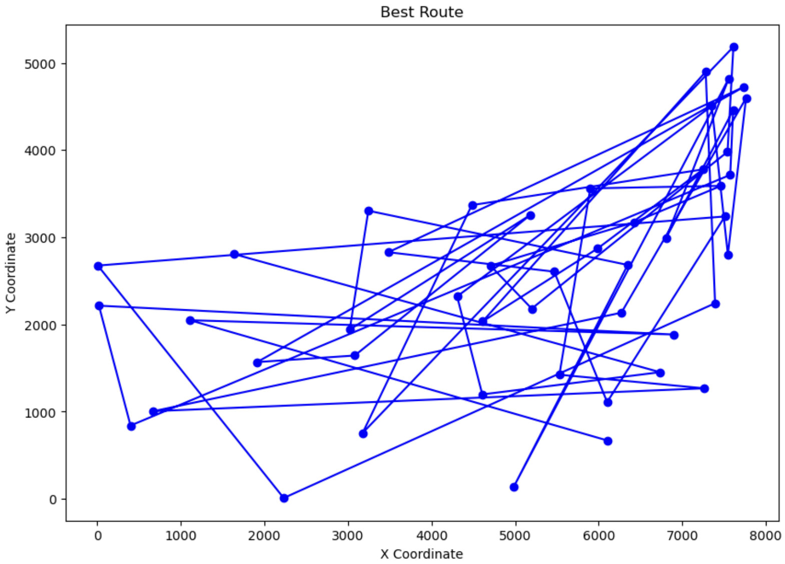
The TSPLIB95 library provides a comprehensive collection of TSP instances, both small and large, designed for testing and benchmarking TSP-solving algorithms. Dataset has taken from give website https://people.sc.fsu.edu/~jburkardt/datasets/tsp/tsp.html

There are taken two datasets of 17 cities and 48 cities.

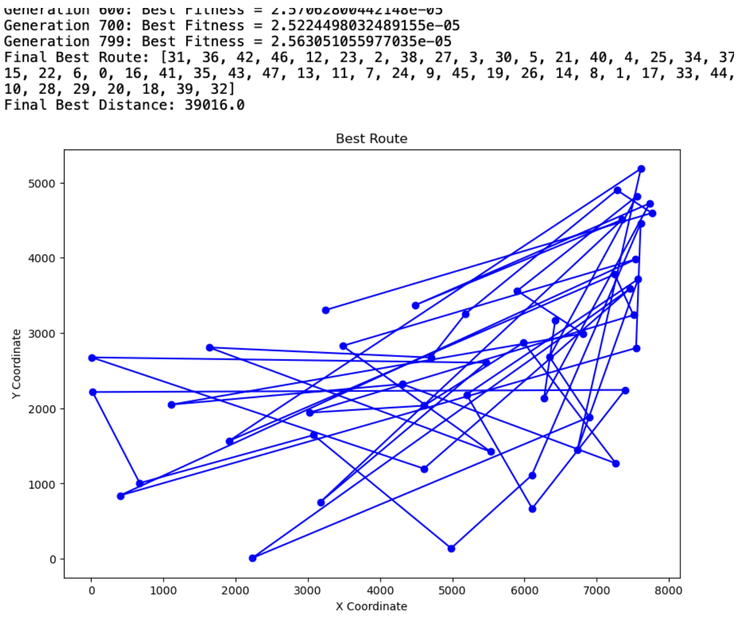
Combination 1: att48.tsp
Selection: Roulette Wheel Selection
Crossover: Ordered Crossover
Mutation: Swap Mutation
Mutation Rate: 0.05
population_size = 100
num_generations = 100



```
Running GA with mutation rate 0.05, population size 100, and 100 generations.
Generation 0: Best Fitness = 2.3930314922944387e-05
Generation 99: Best Fitness = 2.8440601803134153e-05
Final Best Route: [23, 0, 33, 24, 29, 30, 36, 39, 25, 7, 2, 19, 17, 32, 12, 22, 27, 1
, 28, 20, 35, 11, 13, 42, 37, 1, 44, 43, 21, 10, 38, 18, 41, 4, 46, 47, 31, 14, 40, 2
, 45, 5, 6, 3, 34, 8, 9, 15]
Final Best Distance: 35161.0
```

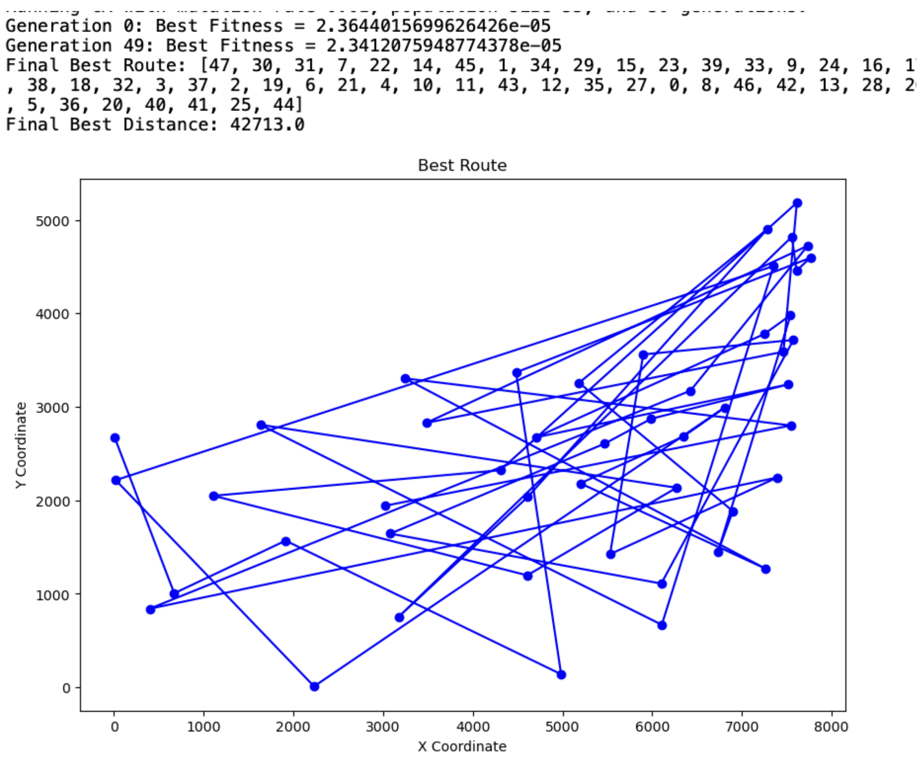


Combination 2: att48.tsp
Selection: Roulette Wheel Selection
Crossover: Ordered Crossover
Mutation: Swap Mutation
Mutation Rate: 0.02
population_size = 700
num_generations = 800



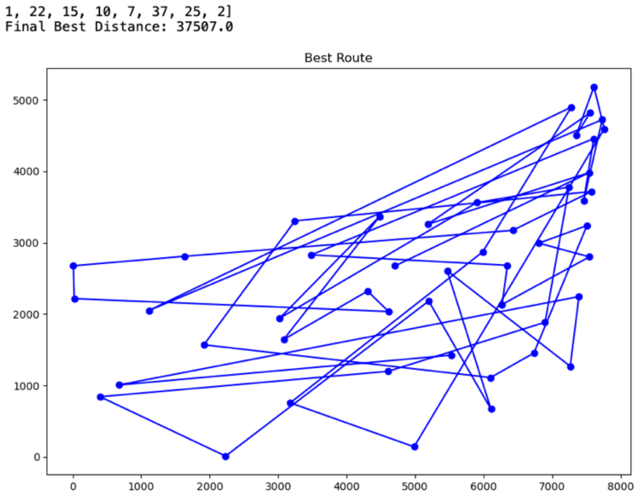
```
Generation 600: Best Fitness = 2.57062800442148e-05
Generation 700: Best Fitness = 2.5224498032489155e-05
Generation 799: Best Fitness = 2.563051055977035e-05
Final Best Route: [31, 36, 42, 46, 12, 23, 2, 38, 27, 3, 30, 5, 21, 40, 4, 25, 34, 37
15, 22, 6, 0, 16, 41, 35, 43, 47, 13, 11, 7, 24, 9, 45, 19, 26, 14, 8, 1, 17, 33, 44,
10, 28, 29, 20, 18, 39, 32]
Final Best Distance: 39016.0
```

Combination 3: att48.tsp
Selection: Roulette Wheel Selection
Crossover: Ordered Crossover
Mutation: Swap Mutation
Mutation Rate: 0.01
population_size = 55
num_generations = 50



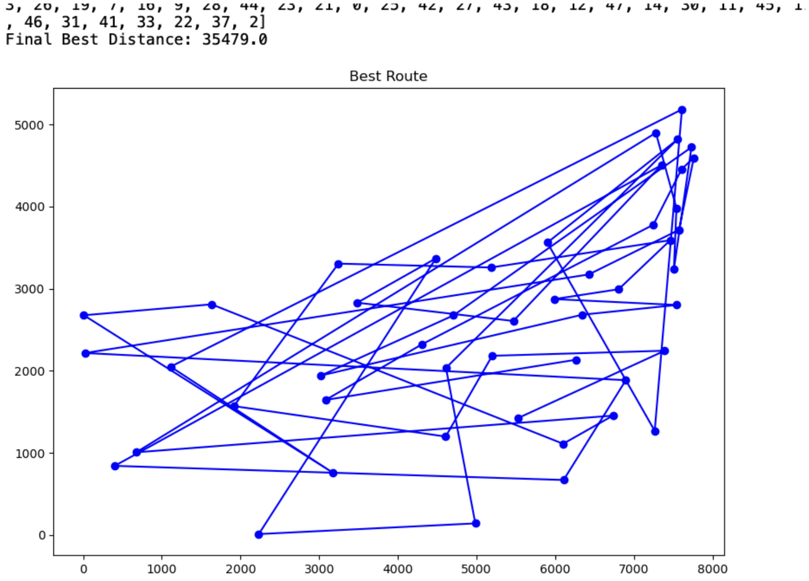
```
Running GA with mutation rate 0.01, population size 55, and 50 generations:
Generation 0: Best Fitness = 2.3644015699626426e-05
Generation 49: Best Fitness = 2.3412075948774378e-05
Final Best Route: [47, 30, 31, 7, 22, 14, 45, 1, 34, 29, 15, 23, 39, 33, 9, 24, 16, 1
, 38, 18, 32, 3, 37, 2, 19, 6, 21, 4, 10, 11, 43, 12, 35, 27, 0, 8, 46, 42, 13, 28, 2
, 5, 36, 20, 40, 41, 25, 44]
Final Best Distance: 42713.0
```

Combination 4: att4.tsp
Selection: Tournament Selection
Crossover: PMX
Mutation: Inversion Mutation
Mutation Rate: 0.01
population_size = 400
num_generations = 600



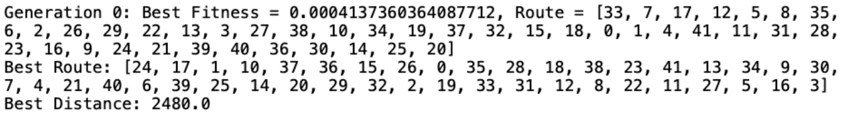
```
1, 22, 15, 10, 7, 37, 25, 2]
Final Best Distance: 37507.0
```

Combination 5: att48.tsp
Selection: Tournament Selection
Crossover: PMX
Mutation: Inversion Mutation
num_cities = len(distance_matrix)
population_size = 55
num_generations = 50
mutation_rate = 0.1



```
3, 26, 19, 7, 16, 9, 28, 44, 23, 21, 0, 25, 42, 27, 43, 18, 12, 47, 14, 30, 11, 45, 1
, 46, 31, 41, 33, 22, 37, 2]
Final Best Distance: 35479.0
```

Combination 6: gr17.tsp
Selection: Tournament Selection
Crossover: PMX
Mutation: Inversion Mutation
mutation_rate = 0.01
population_size = 55
num_generations = 50



```
Generation 0: Best Fitness = 0.0004137360364087712, Route = [33, 7, 17, 12, 5, 8, 35,
6, 2, 26, 29, 22, 13, 3, 27, 38, 10, 34, 19, 37, 32, 15, 18, 0, 1, 4, 41, 11, 31, 28,
23, 16, 9, 24, 21, 39, 40, 36, 30, 14, 25, 20]
Best Route: [24, 17, 1, 10, 37, 36, 15, 26, 0, 35, 28, 18, 38, 23, 41, 13, 34, 9, 30,
7, 4, 21, 40, 6, 39, 25, 14, 20, 29, 32, 2, 19, 33, 31, 12, 8, 22, 11, 27, 5, 16, 3]
Best Distance: 2480.0
```