Middlesex University London

CST3170

Artificial Intelligence

Travelling Salesman Problem

COURSEWORK 1 – Final Report

Maaz Chowdhry

M00910300

Algorithm Used: Genetic Algorithm with 3 OPT

In addressing the Traveling Salesman Problem (TSP), a hybrid approach using a Genetic Algorithm enhanced with the 3-OPT mutation strategy was selected and applied.

1. Why was this algorithm selected?

Genetic Algorithm:

- Efficiency over Brute Force: Initially, a brute force method was employed, but it proved inefficient as the number of cities increased. Genetic Algorithm was chosen for its greater efficiency in handling larger datasets.
- Mimicking Natural Processes: Genetic Algorithm's approach, inspired by natural selection and evolution, provides a dynamic and adaptive way to explore large solution spaces.
- Avoidance of Local Minima: Traditional methods can easily get stuck in local minima. Genetic Algorithm, with its population-based approach, offers a better chance of escaping such traps and finding near-optimal solutions.

Enhancement with 3-OPT:

- Local Optimization: While the Genetic Algorithm excels at exploring a wide range of possibilities, the 3-OPT algorithm enhances this by fine-tuning solutions focusing on local improvements.
- Reduced Risk of Stagnation: The incorporation of 3-OPT into the mutation phase of the Genetic Algorithm helps avoid premature convergence and ensures continuous improvement of solutions.

2. Application of the Algorithm

Setup and Initialization:

- Multiple Populations: Created five distinct populations, each with 150 unique paths, by randomly shuffling cities in each path. This diversity aids in exploring various solution pathways.
- Initial Strategy: Started with brute force to understand the problem's complexity, then shifted to Genetic Algorithm for a more scalable solution.

Algorithm:

- Iteration and Evolution: The algorithm iteratively evolved each population. In each generation, the top-performing paths were selected for further mutations using 3-OPT.
- Convergence Criterion: The process continued until all five populations converged on a single path, ensuring that the solution was consistently recognised as optimal across different populations.
- Mutation Mechanism: During the mutation phase, the 3-OPT strategy was applied. This involved making local changes by swapping three edges to find shorter and more efficient paths.
- Balanced Approach: This combination of the Genetic Algorithm's global search capabilities and 3-OPT's local optimisation provided a balanced approach to solving TSP.

Outcome:

This methodology efficiently navigated the complex solution space of TSP, effectively balancing between exploring new solutions and refining existing ones.
The algorithm proved to be more scalable and robust compared to brute force, especially for larger numbers of cities.

Additional Implementation Details

Termination Condition:

- Maximum Generation Limit: To avoid indefinite running, a maximum generation limit of 1 million was set, acting as a fail-safe if convergence was not achieved.

Final Test Configuration:

- Reduced Populations and Generations: For the final test, the number of populations was reduced to two, and the maximum generations were capped at 150. This configuration aimed to find a "good enough" path quickly, balancing efficiency with solution quality.

Self-Marking Sheet (100/100)

Opinion	Points	Area
10	10	Self Marking Sheet
10	10	Solve First Training Problem
10	10	Get Optimal Results for All Three Training Problems.
10	10	Describe Algorithm(s) Used
10	10	Quality of Code
20	20	Get Optimal Results for the First Three Tests
20	20	Get Optimal Results for the First Three Tests in under a minute.
10	10	Best system on Fourth Test (Path length squared times time.)

Results Screenshots

Train 1 -> 24.293 ✅ Optimal Result

```
Best Path:
1 3 2 4 1
Best Distance: 24.293
Generations: 1
Time: 0 min, 0 sec, 43 ms, 935250 ns
Time: 43935250 ns
Time x Shortest Path^2 :2.592838115327725E10
```

Train 2 -> 65.654 ✅ Optimal Result

```
Best Path:
1 3 2 4 6 5 8 7 1
Best Distance: 65.654
Generations: 5
Time: 0 min, 0 sec, 46 ms, 262667 ns
Time: 46262667 ns
Time x Shortest Path^2 :1.9941280730621857E11
```

Train 3 -> 229.5092 ✅ Optimal Result

```
Best Path:
1 2 9 6 8 3 4 5 7 1
Best Distance: 229.5092
Generations: 5
Time: 0 min, 0 sec, 46 ms, 174792 ns
Time: 46174792 ns
Time x Shortest Path^2 :2.4322328291578916E12
```

Test 1 -> 560.8263 ✅ Optimal Result ✅ Under a minute

```
Best Path:
1 4 3 2 5 8 12 7 6 11 9 10 1
Best Distance: 560.8263
Generations: 5007
Time: 0 min, 0 sec, 786 ms, 773917 ns
Time: 786773917 ns
Time x Shortest Path^2 :2.4746096220028806E14
```

Test 2 -> 718.1851 ✅ Optimal Result ✅ Under a minute

```
Best Path:
1 8 10 4 9 12 5 11 15 7 3 14 13 6 2 1
Best Distance: 718.1851
Generations: 18020
Time: 0 min, 2 sec, 742 ms, 796833 ns
Time: 2742796833 ns
Time x Shortest Path^2 :1.4147067337815045E15
```

Test 3 -> 118524.0861 ✅ Optimal Result ✅ Under a minute

```
Best Path:
1 13 18 2 3 9 16 7 15 14 5 8 17 11 4 6 12 19 10 1
Best Distance: 118524.0861
Generations: 25035
Time: 0 min, 3 sec, 845 ms, 747708 ns
Time: 3845747708 ns
Time x Shortest Path^2 :5.402490607187301E19
```

Test 4 -> $Time\ *\ (ShortestPath)^2 = 3.50E18$

```
Stopped because of generation limit of 150 reached for city lenght:39.
Best Path:
1 9 35 17 30 18 7 23 12 10 3 2 24 4 37 34 25 15 20 28 19 5 8 21 31 39 22 6 14 13 38 36 26 27 16 33 29 32 11 1
Best Distance: 230326.6307
Generations: 151
Time: 0 min, 0 sec, 65 ms, 990334 ns
Time: 65990334 ns
Time x Shortest Path^2 :3.5008107646856146E18
```