Ant Colony Optimization for the Traveling Salesman Problem

1. Introduction

The Traveling Salesman Problem (TSP) is a classic optimization problem where the goal is to find the shortest possible route that visits a set of cities and returns to the origin city. Ant Colony Optimization (ACO) is a metaheuristic inspired by the foraging behavior of ants, which is effective for solving combinatorial optimization problems like the TSP.

2. How ACO Works in This Problem

In this project, ACO is used to solve the TSP by simulating a colony of artificial ants that iteratively build solutions and update pheromone trails. Each ant constructs a tour by probabilistically choosing the next city based on pheromone intensity and heuristic information (inverse of distance). After all ants complete their tours, pheromone trails are updated to reinforce shorter paths. This process is repeated for a fixed number of iterations, and the best solution found is recorded.

3. Problem Configuration

- Two sets of cities were used: 10 and 20 cities.
- Distances between cities were randomly generated as integers between 3 and 50, and the same distance matrix was used for all runs for a given city set.
- For each city set, ACO was run with 1, 5, 10, and 20 ants, each for 50 iterations.

4. Distance Matrices

Distance Matrix for 10 Cities:

```
[ 0 25 14 23 11 19 14 23 12 23]
[25  0 27 36 18 17 20 41 31 16]
[14 27  0 24 16  7 14 25 27 32]
[23 36 24  0 31 18 16  8 19 16]
[11 18 16 31  0 28 30  8 21 29]
[19 17  7 18 28  0 39 30 30 39]
[14 20 14 16 30 39  0 26 18 28]
[23 41 25  8  8 30 26  0 24 33]
[12 31 27 19 21 30 18 24  0 15]
[23 16 32 16 29 39 28 33 15  0]
```

Distance Matrix for 20 Cities:

```
[ 0 12 48 23 17 32 30 24 26 17 29 33 11 31 47 37 38 34 27 30]
[12  0 11 45 43 42 27 40 27 22 24 22 26 34 17 36 34 27 14 19]
[48 11  0 32 27 25 49 15 46 30 38 25 17 20 27 27 24 35 26 31]
[23 45 32  0 16 14 12 33 34 27 27  8 24 20 20 35 19  5 36 14]
[17 43 27 16  0 34 18 35 32 41 22 18 36 34 18 36 28 24 34 32]
[32 42 25 14 34  0 19 35 18 25 25 41 44 34 37 28 29 20 27 40]
[30 27 49 12 18 19  0 21 35 39 41 41 43 44 32 15 29 17 26 43]
[24 40 15 33 35 35 21  0 28 35 22 10 44 19 32 48 22 26 44 16]
[26 27 46 34 32 18 35 28  0 31 32 37 23 25 30 14 46 29 29 29]
[17 22 30 27 41 25 39 35 31  0 15 16 12 39 25 27 25 31 29  5]
[29 24 38 27 22 25 41 22 32 15  0 27 39 38 25 38 28 36 17 25]
[33 22 25  8 18 41 41 10 37 16 27  0 31 18 29 31 32 29 24 28]
[11 26 17 24 36 44 43 44 23 12 39 31  0 45 38 28  5 36 36 14]
[31 34 20 20 34 34 44 19 25 39 38 18 45  0 22 26 27 26 13 10]
[47 17 27 20 18 37 32 32 30 25 25 29 38 22  0 24  3 16 23 18]
[37 36 27 35 36 28 15 48 14 27 38 31 28 26 24  0 31 29 35 36]
[38 34 24 19 28 29 29 22 46 25 28 32   5  27   3  31 0   8  11 32]
[34 27 35  5 24 20 17 26 29 31 36 29 36 26 16 29  8  0 32 28]
[27 14 26 36 34 27 26 44 29 29 17 24 36 13 23 35 11 32  0 39]
[30 19 31 14 32 40 43 16 29  5 25 28 14 10 18 36 32 28 39  0]
```

5. Results

10 Cities

Best Path (1 ant): [3, 7, 4, 0, 6, 2, 5, 1, 9, 8, 3]

Best Cost (1 ant): 129

Best Path (5 ants): [6, 2, 5, 1, 9, 8, 0, 4, 7, 3, 6]

Best Cost (5 ants): 124

Best Path (10 ants): [6, 3, 7, 4, 0, 8, 9, 1, 5, 2, 6]

Best Cost (10 ants): 124

Best Path (20 ants): [8, 0, 4, 7, 3, 6, 2, 5, 1, 9, 8]

Best Cost (20 ants): 124

20 Cities

Best Path (1 ant): [14, 16, 12, 0, 9, 19, 13, 18, 1, 2, 7, 11, 3, 17, 6, 15, 8, 5, 10, 4, 14]

Best Cost (1 ant): 256

Best Path (5 ants): [10, 18, 13, 19, 9, 12, 16, 14, 4, 0, 1, 2, 7, 11, 3, 17, 6, 15, 8, 5, 10]

Best Cost (5 ants): 250

Best Path (10 ants): [0, 12, 16, 14, 17, 3, 11, 7, 2, 1, 18, 13, 19, 9, 10, 5, 8, 15, 6, 4, 0]

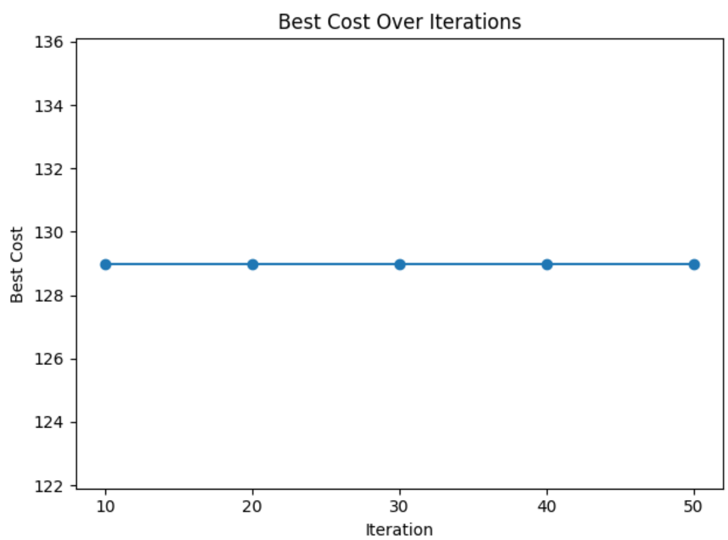
Best Cost (10 ants): 248

Best Path (20 ants): [4, 0, 12, 16, 14, 17, 3, 11, 7, 2, 1, 18, 13, 19, 9, 10, 5, 8, 15, 6, 4]

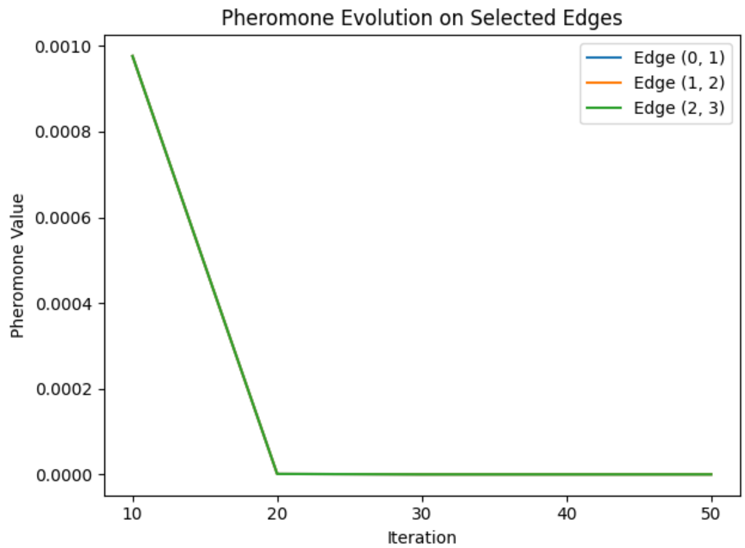
Best Cost (20 ants): 248

6. Pheromone Map and Best Cost Graphs

10 Cities, 1 Ant

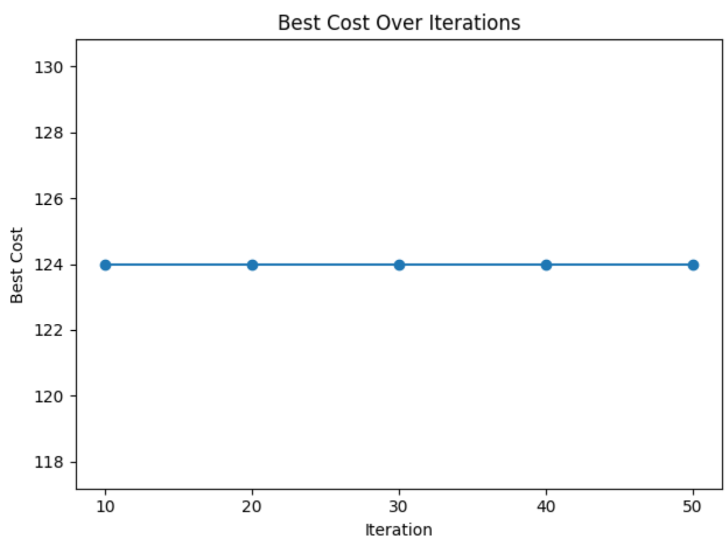


This plot shows that the best cost found by the ant stayed at 129 for all iterations. The solution was found early and did not improve further.

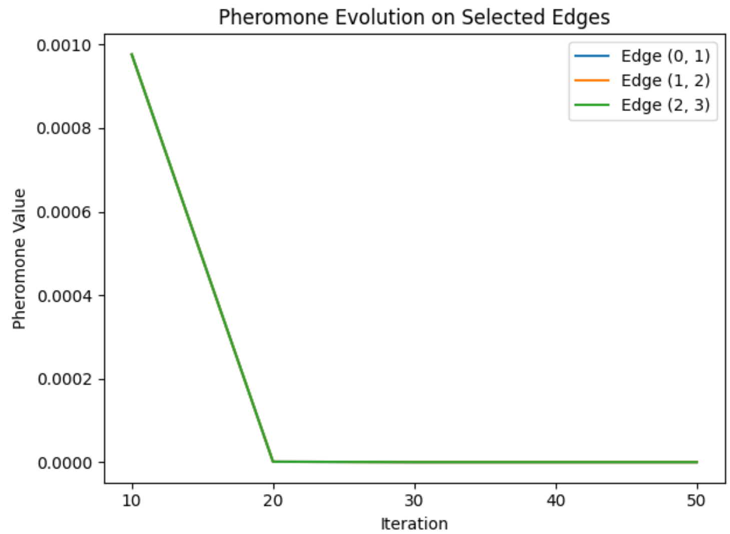


All selected edges quickly lost their pheromone, dropping to zero and staying there. This means these edges were not used in the best paths.

10 Cities, 5 Ants

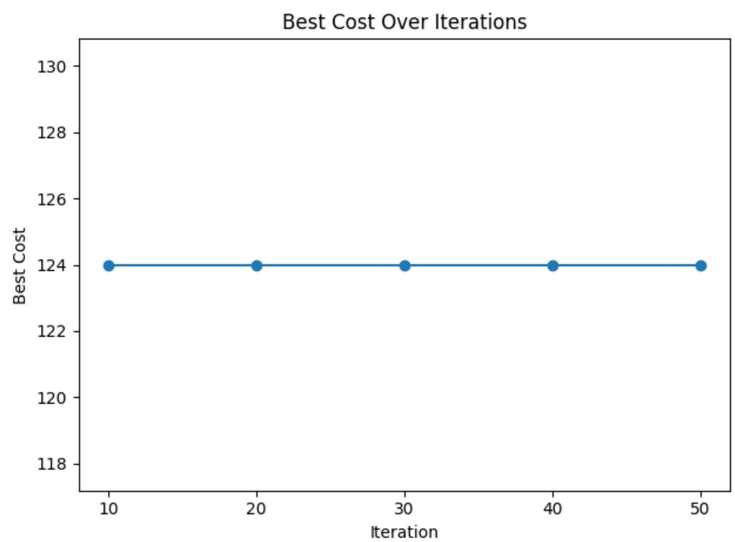


This plot shows that the best cost found by the ant stayed at 124 for all iterations. The solution was found early and did not improve further.

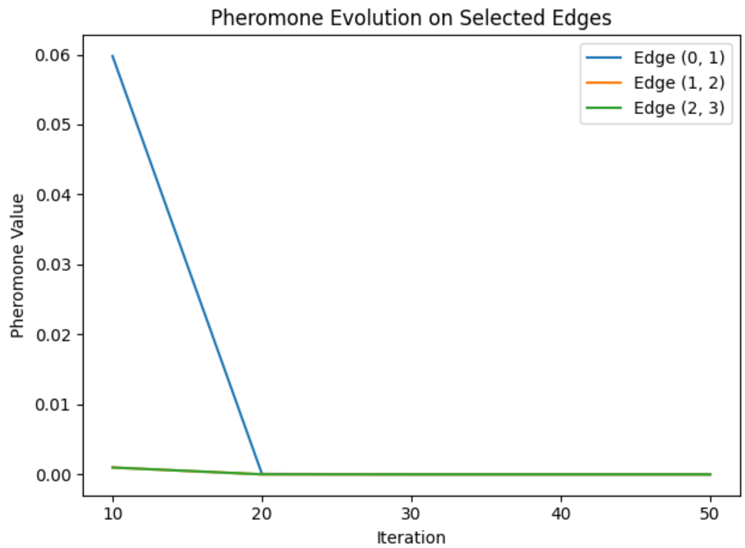


All selected edges quickly lost their pheromone, dropping to zero and staying there. This means these edges were not used in the best paths.

10 Cities, 10 Ants

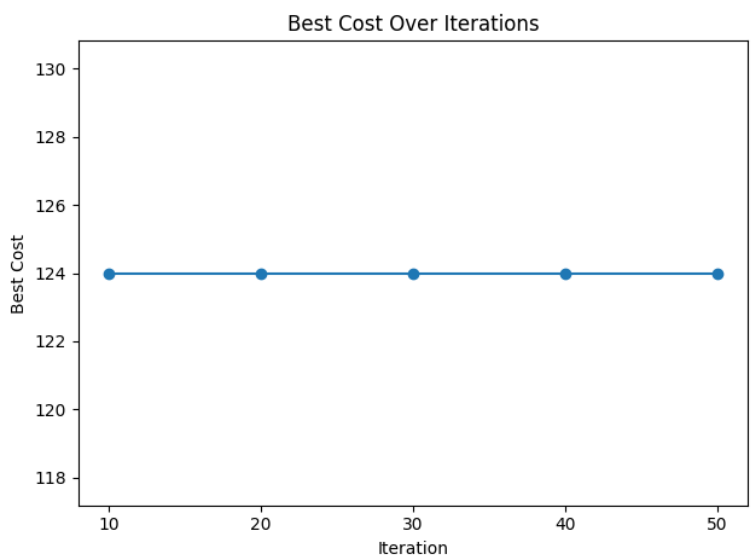


This plot shows that the best cost found by the ant stayed at 124 for all iterations. The solution was found early and did not improve any further.

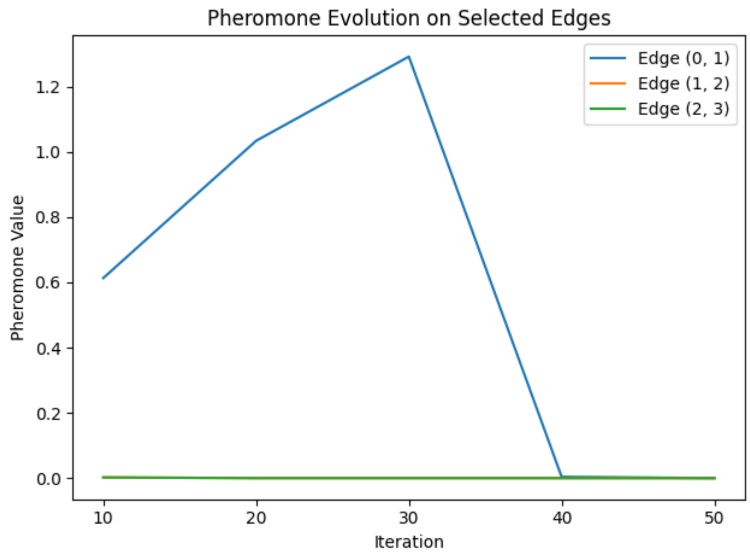


The pheromone on edge (0, 1) and (2, 3) quickly dropped to zero and stayed there, showing these edges were not used in the best solutions. Edge (1, 2) was never used.

10 Cities, 20 Ants

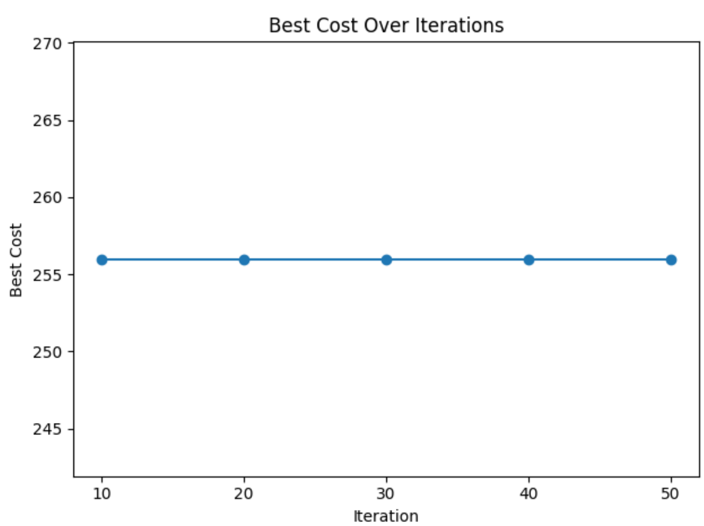


This plot shows that the best cost found by the ant stayed at 124 for all iterations. The solution was found early and did not improve any further.

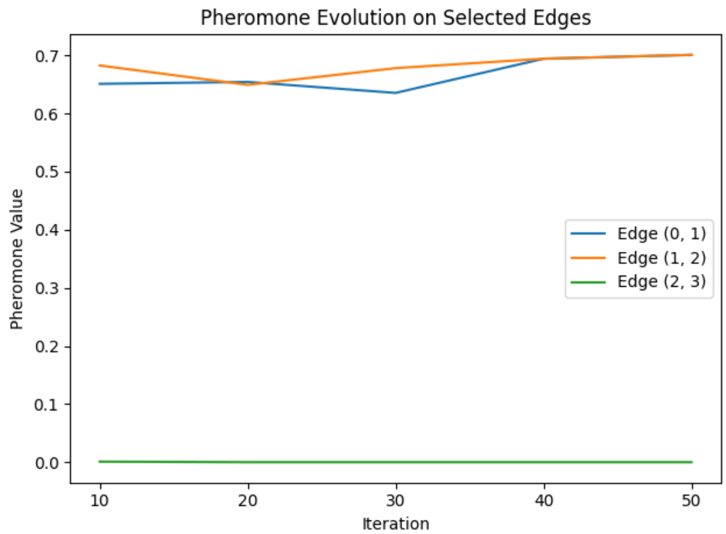


The pheromone on edge (0, 1) increased at first, then dropped to zero, meaning it was used in good paths early but not later. Edges (1, 2) and (2, 3) were not used.

<u>20 Cities, 1 Ant</u>

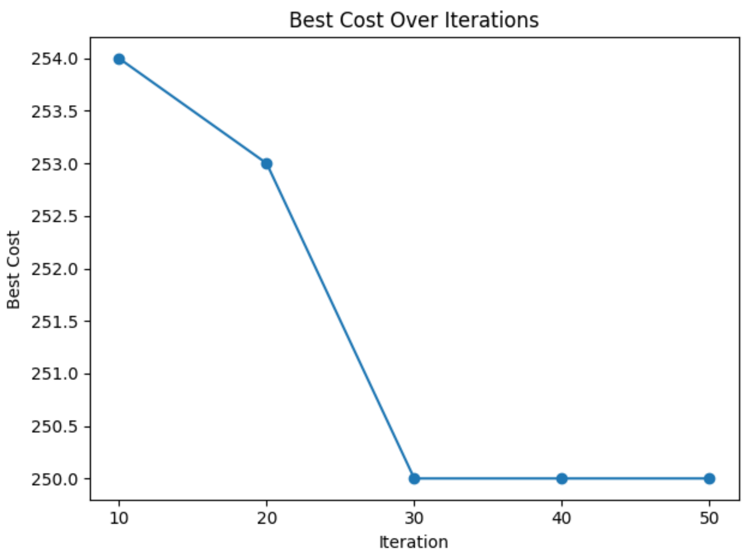


This plot shows that the best cost found by the ant stayed at 256 for all iterations. The solution was found early and did not improve further

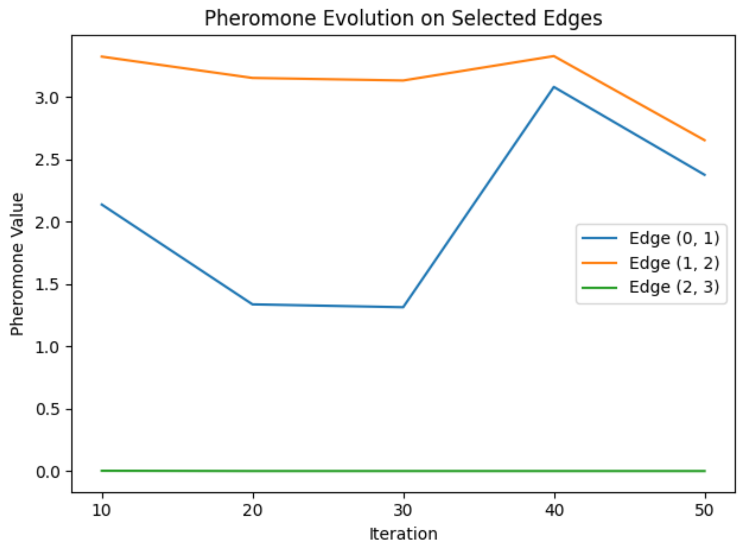


The pheromone values for edges (0, 1) and (1, 2) stayed moderate and increased slightly, showing these edges were used in good solutions. Edge (2, 3) was not used.

20 Cities, 5 Ants

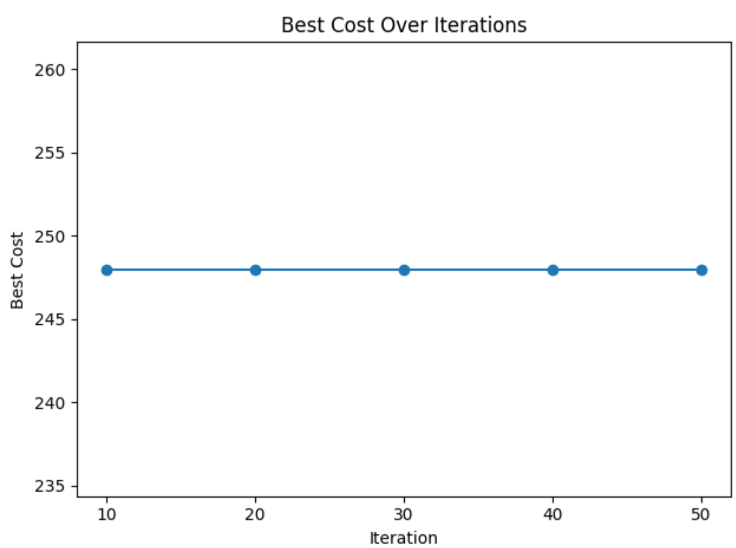


The best cost improved over time, dropping from 254 to 250 by an iteration of 30, and then stayed the same. This shows the algorithm found a better solution as it progressed

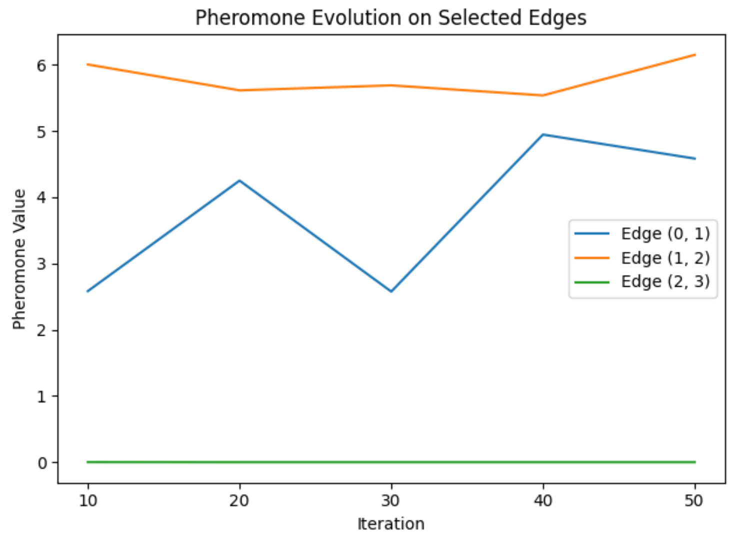


The pheromone values for edges (0, 1) and (1, 2) fluctuated but stayed high, showing these edges were often used in good solutions. Edge (2, 3) was not used.

20 Cities, 10 Ants

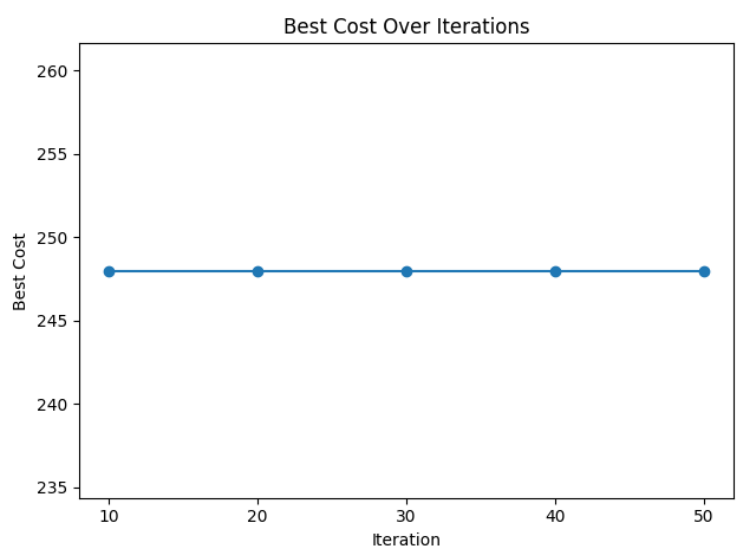


This plot shows that the best cost found by the ant stayed at 248 for all iterations. The solution was found early and did not improve any further.

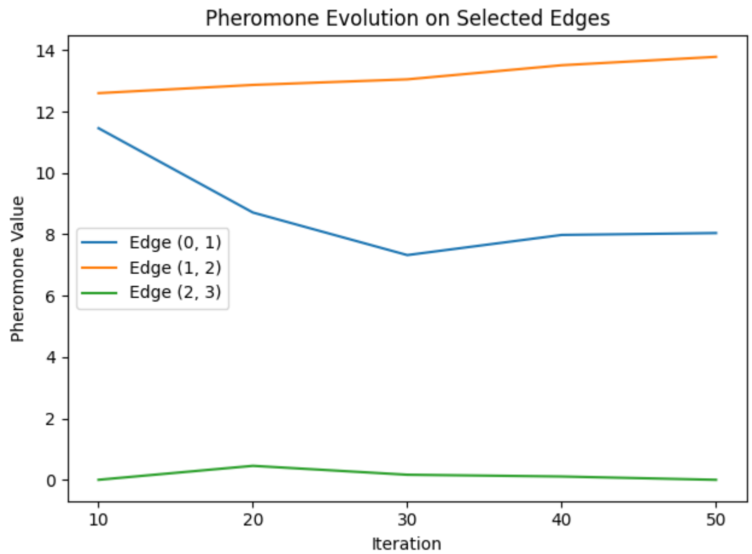


The pheromone values for edges (0, 1) and (1, 2) stayed high and varied, showing these edges were frequently used in good solutions. Edge (2, 3) was not used.

<u>20 Cities, 20 Ants</u>



This plot shows that the best cost found by the ant stayed at 248 for all iterations. The solution was found early and did not improve any further.



The pheromone values for edges (0, 1) and (1, 2) stayed high and increased, showing these edges were consistently used in the best solutions. Edge (2, 3) had low values, meaning it was rarely used.

7. Analysis

The 'Best Cost Over Iterations' plots show that the algorithm often finds the best solution early, as the best cost stabilizes quickly. The 'Pheromone Evolution on Selected Edges' plots illustrate how the algorithm reinforces certain edges that are part of good solutions. Edges with rising pheromone values are frequently used in the best paths, while those with flat or zero values are not. For larger city sets and more ants, the pheromone values and best cost may fluctuate more before stabilizing, indicating more exploration before convergence.

8. Conclusion

ACO is effective for the TSP, especially with a moderate number of ants. The algorithm quickly finds good solutions, and the pheromone mechanism helps reinforce optimal or near-optimal paths. Larger problems benefit more from increased colony size, but the best cost may not always improve with more ants beyond a certain point.