AI332
Computational Cognitive Science

Task2 Report

Written By:

Abdulrhman Ahmed [ID: 20210566]
Adham Waleed Gaber [ID: 20210058]
Noha Mohamed [ID: 20210425]

Ant Colony Optimization Simulation Report

Ant Colony Optimization (ACO) is a swarm optimization algorithm, it is a probabilistic technique for finding optimal paths.

In the TSP, the challenge is to find the shortest possible route that allows a salesman to visit each city once and return to the starting city, minimizing the total travel distance.

ACO Algorithm for TSP

1. **Initialize the parameters: as the number of Ants M, the pheromone evaporation rate ρ, initial pheromone level on the edges τxy .**
2. **Distance Matrix: This matrix is a fundamental component that quantifies the distance between each pair of cities. It remains constant throughout the process.**
3. **Pheromone Update:**

- **Evaporation: After each tour or set of tours, some amount of pheromone evaporates from all paths, which prevents infinite accumulation and encourages exploration of new paths. This is usually controlled by a parameter called the evaporation rate.**
- **Deposition: Ants deposit pheromones on the paths they traversed based on the quality of their tour. Typically, the shorter the tour, the more pheromones are deposited, which reinforces successful paths for future ants to follow.**

4. **calculate_probabilities: The probability of moving to the next city depends on the amount of pheromone on the connecting edge and the visibility (inverse of the distance). This rule is often defined as:**
5. **Iteration:**
6. **The process repeats over multiple iterations. Each iteration represents a complete set of tours generated by all ants.**
7. **Ants are reset to construct new solutions using the updated pheromone levels.**
8. **Termination:**
9. **The algorithm can be set to terminate after a fixed number of iterations or when there is no improvement in the solution for a specified number of iterations.**
10. **The best solution found across all iterations is reported as the optimal or near-optimal tour.**

2_ Display of Distances Between Cities

Distances Table for 10 and 20 Cities:

Present depend on the generated distance matrices for each city configuration (10 and 20 cities) which will be constant through all experiments for a particular number of cities.

3. Development of Pheromone Map and Current Optimal Path

Pheromone Map Visualization:
For iterations 10, 20, 30, 40, and 50, display the pheromone levels on each path (i, j) using a table or graphical representation to show the evolution of pheromone concentrations.

Optimal Path Updates:
For the same set of iterations, detail the current optimal path and its length.

4.Iteration: Steps 2 and 3 are repeated for a number of iterations. Over time, efficient paths accumulate more pheromone and hence are more likely to be selected by ants.

5.Convergence: The algorithm continues until a stopping criterion is met, which could be a set number of iterations, or when the solution quality (path length) ceases to improve.

Results:

Simulation for 10 Cities:

The distance matrix for the 10 cities is as follows:

```
[[ 7 15  5  9 15 17  4 13  9  3]
 [19 10  3 14 11  7 15 16 13 19]
 [13 12  5  4 10 10 16 14 19  5]
 [11 11  5 11  5  6  8 10 18  9]
 [12  4  5 17 10 10 19  5 14 14]
 [18  3 16  3 17 15 14 13  9  7]
 [10  3 12  9  7 17  9 13  5 10]
 [11  5  3 18 14  3  3  6 18 15]
```

```
 [ 4 10 11  7  4 19 18 16 13 17]
 [ 6  3 10 16  7 14  9  4  4 19]]
```

Outcomes:

With 1 ant:
Best path: [2, 1, 8, 0, 5, 3, 4, 7, 6, 9, 2] with length: 78
Worst path: [2, 1, 8, 0, 5, 3, 4, 7, 6, 9, 2] with length: 78

With 5 ants:
Best Path: [3, 4, 2, 9, 7, 6, 5, 8, 0, 1, 3] Length: 55
Worst Path: [3, 1, 5, 0, 7, 9, 4, 2, 6, 8, 3] Length: 115

With 10 ants:
Best Path: [5, 4, 8, 7, 3, 1, 0, 2, 6, 9, 5] Length: 53
Worst Path: [4, 9, 1, 3, 7, 5, 6, 2, 8, 0, 4] Length: 112

With 20 ants:
Best Path: [4, 6, 7, 9, 8, 5, 3, 1, 2, 0, 4] Length: 46
Worst Path: [8, 4, 6, 2, 1, 0, 9, 3, 5, 7, 8] Length: 133

Simulation for 20 Cities:

The distance matrix for the 20 cities is as follows

```
[[11 15 17 19 11 12 10 12 14 18 14 12 19 12 11  4  3  5  5  4]
 [ 6 16  9  8 16 14 13  6  7 17 19 18 11 17 18  8  8 10 16 13]
 [10  6 10 15 10  9 18  3  4 18 14 14 11  4 10 16 14 19 10  4]
 [12 14 12  9 16  9 15 19 15  7 14  9  6 18 12 13 17 19  7 12]
 [ 5 13  5 19 16 13 18 17  8 19 17 14 13 13 17 11  5  3 19 15]
 [11  6  7 12  4  3 19 18 19 18 19 13  3  9  5 18 12  8  4 17]
 [17  4  9 15 15 14 16  7 19 19 13  8 10  7 11 15  8  6  8 12]
 [17  5  7 15  9 15 17  7 10  4  9 18  9  7 14  7  6 17 17  6]
 [16 10  3 14 14 11 18  6 15 18  9  6 11 14  3  5 18  3 11 13]
 [17 14 14  8 18  5 12  6  7  8 18  5 13  3  7  7  7  6 17 16]
 [17 15 17 15 10 13 16  6 17 19 19  6  5 12  5 15 16 17  5 13]
```

```
 [ 4  5 10 12  4  5  5 10 17 10  8  3 11  5  5  8 10  6  9 14]
 [18 13 12 17  3  4 17  8 10 11 17  4 14  3 19 19 10 16  7  8]
 [ 4  5 11 16 16 14  9 17  6  9 17 18  4  9 18  3 12  7 19  4]
 [19 19  7  7 17 18 13  3  4  5  8 16 14 15 17 10  3 17 19  3]
 [10  5 18 16 12  7  3  5  9  7  5 12  9 13 14 17 10 15  4  3]
 [ 8 11  6  3  8  6 18 17 12 18 19  4 14  5  7 15 13 15 14 17]
 [ 6  7 11  7 18 18 19 18 10  7  5 17  5 19 12 13 16 10 18  6]
 [12 16 12  9 18  5 17 19  4  8  6 14 12  6  8 10  8  6 11 10]
 [ 5  3 18  7 11 14  5 14  7  9 14 17  6 18 14 15  8  4  7 15]]
```

With 1 ant:

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

Length: 106

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

Length: 184

With 5 ants:

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

Length: 90

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

Length: 175

With 10 ants:

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

Length: 83

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

Length: 162

With 20 ants:

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

Length: 78

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

Length: 170

Differences in ACO Performance: 10 Cities vs. 20 Cities

Overview

Comparing the outcomes of the Ant Colony Optimization (ACO) algorithm when applied to two different scales—10 cities and 20 cities—with the same number of ant agents provides insight into the algorithm's scalability and performance under varying complexities.

Results Comparison:

Path Lengths:

In all scenarios, regardless of the number of ants, path lengths are generally longer for 20 cities compared to 10 cities. This is an expected result due to the increased number of possible paths and the greater complexity in finding the optimal tour.

Best and Worst Path Lengths:

The difference between the best and worst path lengths tends to increase with more cities. For instance, with 1 ant, the best path length in the 10-city setup was 78, and the worst was 95. For 20 cities, these lengths were 106 and 184, respectively. This indicates that the complexity added by more cities can lead to greater disparity in path quality, especially with fewer ants.

Improvement with More Ants:

The improvement from increasing the number of ants is more pronounced in the 20-city setup. For example, the reduction in path length from the worst to the best scenario is more substantial when the number of ants is increased in the 20-city setup compared to the 10-city setup. This suggests that more ants contribute more significantly to exploring and optimizing paths in more complex scenarios.

Impact of City Set Size:

With more cities, there is a greater need for exploration to effectively cover the expanded solution space. This need is partially met by increasing the number of ants, which allows for more extensive parallel exploration and reduces the chance of converging prematurely on suboptimal paths.

Stochastic Variability:

The variability in path lengths observed (difference between the best and worst paths) is indicative of the stochastic nature of ACO. This variability tends to be more pronounced in larger city sets, suggesting that randomness in path selection and pheromone deposition has a more significant impact when the problem space is larger.

Performance Scaling:

The algorithm's performance does not degrade linearly with the increase in city number, showing reasonable scaling. However, optimal performance in larger city sets clearly benefits from a greater number of agents, emphasizing the need for sufficient computational resources to handle larger problems.

Conclusion:

When using the same amount of ant agents on a larger set of cities, there are noticeable differences in path lengths, the degree of variability between the best and worst paths, and the impact of increasing ant numbers. The results highlight the challenges of scaling up the ACO algorithm and underscore the importance of tuning the number of ants and other parameters to suit the complexity of the task. Overall, ACO remains a robust method for tackling larger instances of the Traveling Salesman Problem, provided that adjustments are made to accommodate the increased complexity.