Ant Colony Optimization for Travelling Salesman Problem

Calculating Probability

calculate_probability(current_city, next_city):

- Function to calculate the probability of moving from current city to the given next city.
- Initially the probability is the same for all cities.
- Calculate probability using this formula:

$$P_{xy}^{k} = \frac{\tau_{xy}^{\alpha} \cdot \eta_{xy}^{\beta}}{\sum_{z \in J_k} \tau_{xz}^{\alpha} \cdot \eta_{xz}^{\beta}}$$

Choosing the next city

choosing_next_city(path):

- Function to calculate the probability of moving to each of the unvisited cities.

- Choose a random choice of the unvisited cities using the roulette wheel according to the probabilities

Generating the paths

generate_paths(num_ants):

- Function to generate the paths according to the number of ants.

- Each ant have a path generated by using the choose_next_city function.

Updating the pheromones

update_pheromones(paths):

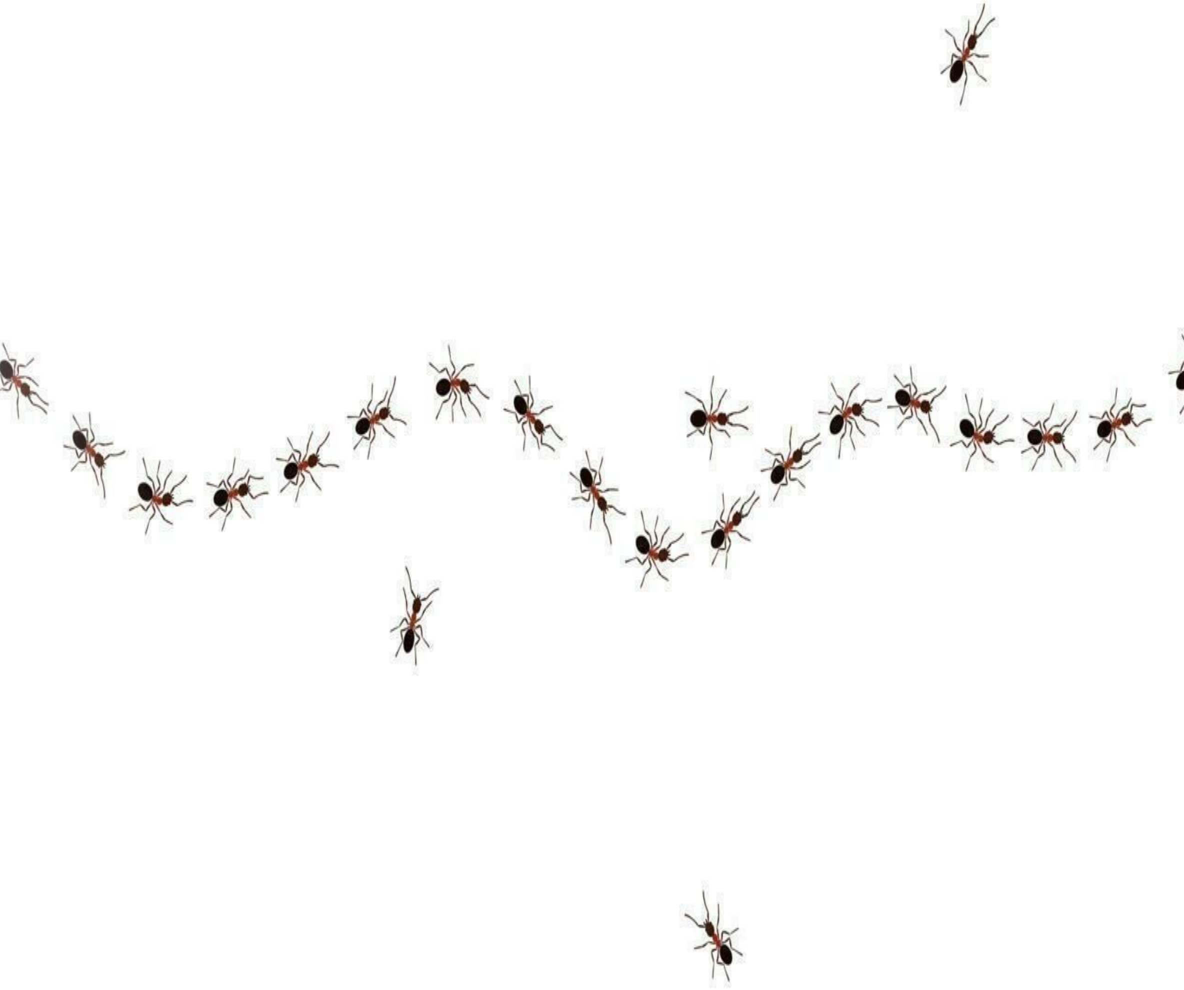
- Function to update the pheromones on the edges between cities.

- Update the best path by finding the shortest path distance / longest path distance.

$$\sum_{k}^{m} \Delta\tau_{xy}^{k}$$

- Apply evaporation to all edges using this formula:

$$\tau_{xy} \leftarrow (1-\rho)\tau_{xy}$$



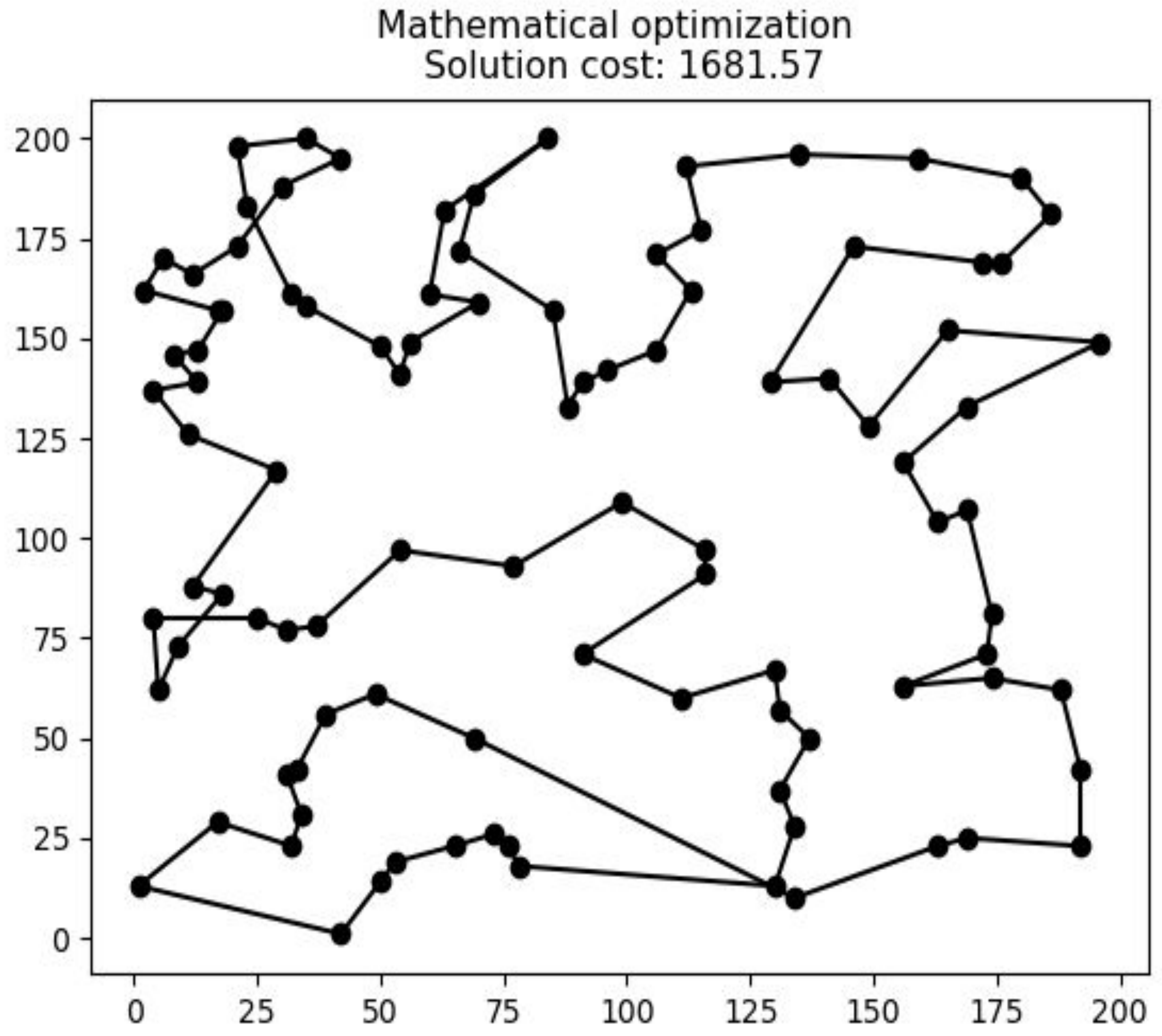
Calculating the distance

calculate_distance(path)

- Function to calculate the total distance of the path.

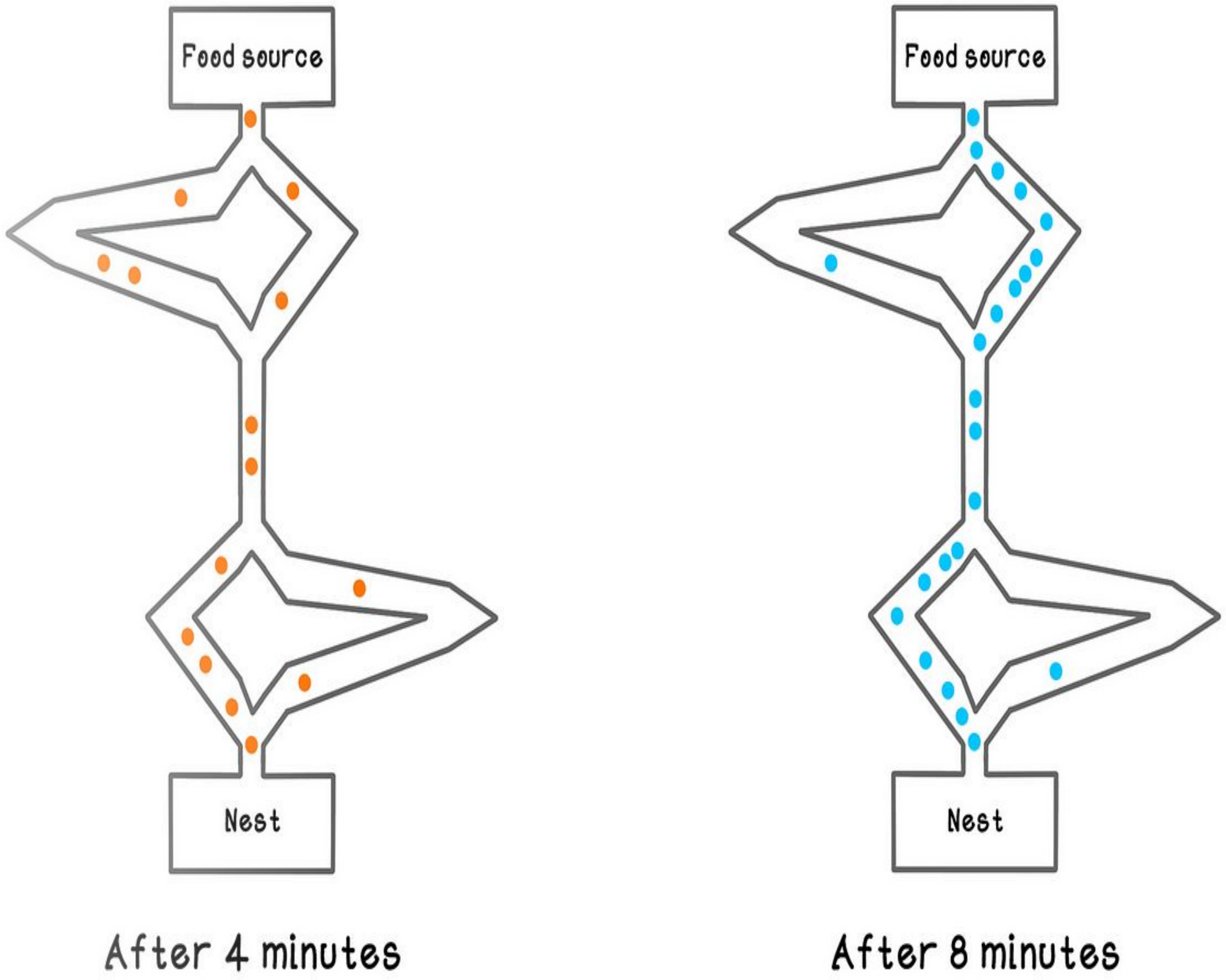
Main Program

- Generation of random distances between cities
- Loop over the number of iterations.
- Construct paths according to the number of ants.
- Update pheromones on the edges.
- Store shortest path
- Logging and Saving Results



Conclusion

- Ant colony optimization presents a promising approach to tackling the challenging Travelling Salesman Problem.
- Through the collective behavior of artificial ants, this algorithm efficiently explores the solution space, finding near-optimal solutions that rival traditional optimization techniques.
- While further optimization may be done, ant colony optimization stands as a compelling tool for addressing combinatorial optimization problems like the Travelling Salesman Problem.



Grokking Artificial Intelligence Algorithms