Research Summary

Students: Wenting Zheng (200123899)

Mentor: Dr. Steffen Heber

Semester: 2019 Spring

# Abstract

This project focuses on assessing the performance of Ant Colony Optimization for solving the Travelling Salesman Problem. The implementation is modified either based on research or considerable tests. The final implementation structure is Max-Min Ant System due to its stable solution with good performance to TSP. Our goal is to explore the relationship between the algorithm’s performance and the parameters.

# Problem Statement

Ant Colony Optimization (ACO) algorithm is inspired by the real ants’ collective intelligence through pheromone-based social interacts. It can generate a near-optimal solution to TSP in a limited time and is applied in many combinatorial optimization problems, such as Traveling Salesman Problem (TSP).

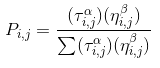
This paper will focus on evaluating the performance of ACO in TSP, including some implementation choices and parameter effects. Since TSP’s type is not the main purpose of this research, we will use the simplest TSP, which measures the Euclidean distance as the cost between cities.

# Literature Summary

Ant System was initially introduced by Marco Dorigo in 1992, which mimics ants’ behavior to find the path between nest and food source. Other scholars later extended the ant colony algorithm family based on this Ant System. For example, Max-Min Ant System was developed by Hoos and Stützle in 1996 and Ant Colony Optimization was developed by Gambardella Dorigo in 1997. The most intuitive application of ACO is planning and scheduling to minimize costs, such as job scheduling, transportation choice, and vehicle routing. It can also be applied in electrical and biomedical fields, such as image processing and DNA sequencing.

There are three main parts in ACO, edge selection, pheromone update, and termination criteria. For each iteration, all the ants first construct the solutions and calculate the length. Then the pheromone will be updated based on the performance of the constructed solutions. Finally, the program will terminate based on certain criteria.

The ants use the following formula to calculate the probability of moving to an unvisited city.



In the formula, 𝒯 represents the amount of pheromone on edge between city i and j. ƞ usually is because the goal is minimizing the total length. ⍺ represents the impact of the pheromone on edge between city i to j while β is the impact of the distance. The higher the impact factor is, the more likely the selection is based on one factor rather than the other. Thus, means the attractiveness from city i to j. Because the ants’ choice is random, it’s not guaranteed the edge with the largest attractiveness will always be selected.

When all the ants finish the tour, the pheromone on the selected edges is updated. There are many ways to update pheromone. The traditional ACO first updates the evaporated pheromone with the evaporation rate(ρ) and add extra pheromone on the selected edges. The amount of additional pheromone(Δ𝒯) is inversely proportional to the tour length.

The other way to update pheromone is known as Max-Min Ant System (MMAS), which usually generates a reliable and effective solution to TSP (Eldem, H., & Ülker, E., 2017). Stützle and Iridia also concluded in their research that on average, the solution quality of MMAS is better than Iterated local search, one of the best algorithms for TSP. There are two distinctions between MMAS and traditional ACO. First, only the ant with the best solution is allowed to add the pheromone. It can be the global best or iteration best solution. Then, the minimal and maximal pheromone on the edge is set, which avoids the situation where the best solution dominates the pheromone map and allows the ants to explore other new solutions.

# Methodology

When implementing ACO, we change the added pheromone amount to be instead of to enlarge the difference between updated pheromone and decrease the overlapping edges. Then, we tested the algorithm through some existing TSPs with solutions. Because the performance of traditional ACO is very unstable, MMAS is the final algorithm structure due to its stability.

A controlled experiment is conducted to explore the parameters’ effect, specifically the impact factor of pheromone (⍺), the impact factor of the distance (β), and the evaporation rate (ρ). We use a control group where ⍺ = 1, β = 7 and ρ = 0.8. Three are three sets of experimental groups for three parameters respectively. ⍺ value of experimental groups starts from 3 to 19 in steps of 2, so we have ⍺ = 3, 5, 7,..., 17, 19. There are 9 groups. β value starts from 1 to 15 in steps of 2 and excludes 7, which is the control group, so we have β = 1, 3, 5, 9, 11, 13, 15. There are 7 groups. For ρ, the value starts from 0.1 to 0.9 in steps of 0.1 and excludes control group 0.8, so we have 0.1, 0.2,..., 0.7, 0.9. There are 8 groups. Therefore, in total, we have 1 control group and 24 experimental groups from three sets. Each group is repeated 20 times to record the mean of the length. Then we will analyze the confidence interval of the best group to access the stability of MMAS to TSP. The performance is evaluated by , which is always greater than 1.

# Results

The mean lengths are plotted in the following graphs with parameter values(⍺, β, ρ) and performance(), where the smaller value is desired. In Figure 1, the impact of pheromone and of distance has a similar influence on the performance. starts with a high value and drops as the ⍺ or β value increases. Then it increases after reaching the best solution in this set of experimental groups. The lowest value is found in the alpha experimental groups when ⍺ = 7.



The evaporation rate doesn’t behave the same as the other two parameters. The value is low and remains stable when the evaporation rate is less than 0.4. A positive correlation between and evaporation rate is shown in the graph when the evaporation rate is more than 0.4.

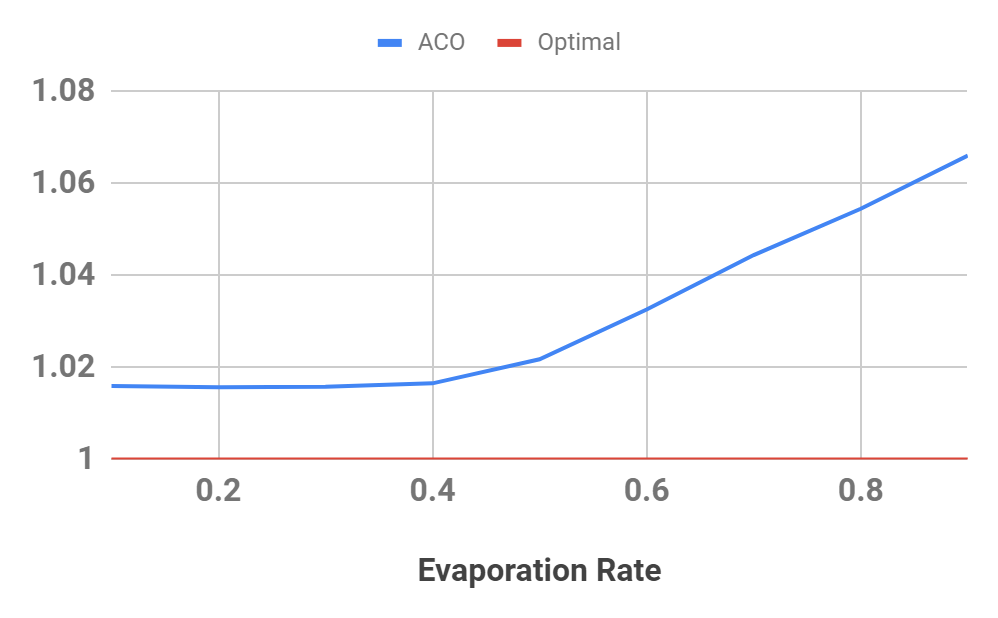


Figure 2: the relationship between performance and evaporation rate

In Figure 1, the reason why small ⍺ and β doesn’t generate good solution is missing important information. The ants either doesn’t take pheromone or distance into account when constructing solutions. Large beta value results in the closeness to the nearest neighbor algorithm where the ants always choose the closest city. That is why the performance value is more stable than the one with large ⍺. This highlights the importance of the distance and pheromone for the ants to make wise decisions. Similarly, in Figure 2, when the evaporation rate is too large, the remaining pheromone has little contribution to the next iteration’s edge selection process, which neglects the pheromone information.

The group with the best performance value is the experimental one when ⍺ = 7. From the result of these 20 repeated experiments ([33548.8, 33600.6, 33548.8, 33588.3, 33600.6, 33600.6, 33588.3, 33523.7, 33588.3, 33607.7, 33588.3, 33523.7, 33600.6, 33523.7, 33588.3, 33548.8, 33548.8, 33588.3, 33600.6, 33632.7]), the confidence obtained is 99% in range of [33558.61813, 33595.33187], which means the value will be in the range of [1.015032287, 1.016142751] with 99% certainty.

# Reflection

I was surprised by the amount of time spent on implementation and data collection. The time of implementation is less than I thought while collecting and analyzing data takes a surprisingly long time. I also realize the importance to do enough research before implementing. This preparation process saves time to “reinvent the wheel”. In the research, we change the implementation from traditional ACO to MMAS because of the poor performance and MMAS actually works better as the previous work showed.

# Citation

Wei Fang, Xiaodong Li, Mengjie Zhang, and Mengqi Hu, “Nature-Inspired Algorithms for Real-World Optimization Problems,” *Journal of Applied Mathematics, vol*. 2015, Article ID 359203, 2 pages, 2015. <https://doi.org/10.1155/2015/359203>.

Eldem, H., & Ülker, E. (2017). The application of ant colony optimization in the solution of 3D traveling salesman problem on a sphere. Retrieved from <https://www.sciencedirect.com/science/article/pii/S2215098617309783>

Stützle, T., & Iridia, M. (1999). ACO Algorithms for the Traveling Salesman Problem. Retrieved from <http://staff.washington.edu/paymana/swarm/stutzle99-eaecs.pdf>