Assignment 3 Report

1 Travelling Sales Person and Ant Colony Algorithm

1.1 Part a

Explain the important operations of the employed algorithm.

Ant Colony Optimization (ACO) works like ants searching for food. Ants follow paths, leaving pheromones on good routes. These pheromones help other ants find better paths. ACO solves problems, like the Traveling Salesman Problem (TSP), by learning from each other, just like ants. In my implementation for each iteration, more than one ant tries to find a path using the information of pheromone and the heuristic information which is the edge length, after all of them finish their findings according to their path length, the pheromones of the path get stronger. And previous pheromones decay a certain amount. After this next iteration starts. There are important operations in this algorithm.

1.1.1 Initialization

First the distance matrix of cities is calculated to use as heuristic information.This matrix holds the information of each city pairs distance.And pheromone matrix is initialized to 1/cityCount for all edges between the cities. Setting the initial pheromone levels to 1 divided by the number of cities helps keep the search balanced, avoiding quick decisions and making it easier to find the best solutions.

1.1.2 Path creation for each ant

Each ant creates its own path by choosing edges probabilistically based on pheromone levels and distances. Inverse distances are used as heuristic information guiding the path construction because the nearest city is more desirable. This process repeats for each ant, contributing to the collective exploration of the solution space. The formulation of probabilistic rule :

$$P_{r,s} = \begin{cases} \frac{\tau_{rs}^{\alpha} \cdot \eta_{rs}^{\beta}}{\sum_{u \in J_k(r)} \tau_{ru}^{\alpha} \cdot \eta_{ru}^{\beta}} & \text{if } s \in J_k(r) \\ 0 & \text{otherwise} \end{cases}$$

where:

- τ_{rs} is the pheromone level of the edge e_{rs}.
- η_{rs} is the heuristic information guiding the path construction.
- $J_k(r)$ is the set of nodes connected to r but not visited by the ant.
- α and β are weights representing the trade-off between the heuristic and pheromones.

1.1.3 Pheremon Update

After all ants have created their paths, the pheromone levels on the edges are updated. Ants add pheromones to the paths they used, with shorter paths receiving proportionally more pheromones. This helps future ants find better paths by following stronger pheromone trails. The pheromone levels also decrease over time according to the evaporation function:

$$\tau_{rs}(t) = (1 - p) \cdot \tau_{rs}^{t-1} + \Delta\tau_{rs}$$

where $\tau_{rs}(t)$ represents the pheromone level on edge (r, s) at time t, p is the evaporation rate, and $\Delta\tau_{rs}$ is the amount of pheromone deposited by the ants on edge (r, s). $\Delta\tau_{rs}$ is calculated with below:

$$\Delta\tau_{rs} = \sum_{k=1}^{m} \Delta\tau_{rs}^{k}$$

where:

$$\Delta\tau_{rs}^{k} = \begin{cases} \frac{1}{C(S_k)}, & \text{if ant } k \text{ uses edge } (r, s) \\ 0, & \text{otherwise} \end{cases}$$

$C(S_k)$ is the path length generated by ant k.

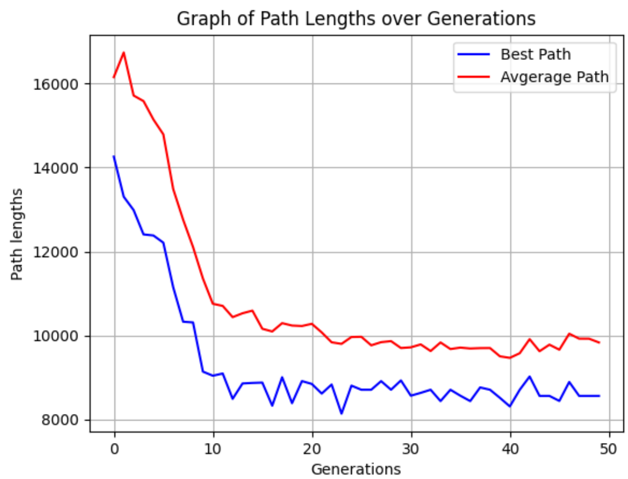

Figure 1: Ant colony evaluation

1.2 Part b

Illustrate how the performance of the population evolves with generations (with a figure.) The above figure illustrates a rapid improvement initially, but around the tenth generation, progress begins to slow down and stabilize. However, it quickly identifies a solution that surpasses a path with a length of 9000. This implementation's best path length is 8135.

1.3 Part c

Compare the results found by this algorithm with the results found with GA. Analyse the differences between the algorithms and the difference in the performance between both.

When we look at the Ant Colony Optimization (ACO) algorithm and the Genetic Algorithm (GA), we can see that they have different ways of working and show different results. ACO, inspired by ant behavior, explores solutions using pheromone trails, favoring thorough exploration. GA operates on a population of solutions, refining them through genetic operations. When comparing the outcomes of the ACO algorithm to those of GA, differences emerge in their performance and solutions. ACO consistently delivers better solutions in notably shorter timeframes compared to GA. This disparity arises from GA's tendency to navigate the search space randomly, often getting stuck at local minima due to limited changes in each generation. In contrast, ACO uses heuristics and past experiences to pursue more global minima, focusing on collective knowledge rather than the initial population setup. Moreover, GA's performance heavily relies on parameter selection, such as mutation rate and population size, necessitating extensive experimentation for optimization. ACO is less reliant on specific parameter values while still achieving valuable results. In summary, the ACO algorithm's effective utilization of heuristics and collective knowledge enables more efficient exploration of the solution space compared to GA. When analyzing the results from the implementations, it is clear that ACO worked faster and found better solutions than GA. ACO worked with 40 ants and it find the paths lesser than 90 before the 15th generation.After 50th generation it found a path with 8300 length. But GA worked with 100 individual population with 25000 generations even than it couldn't find better solution.