

HSG-WAS-SS23 / exercise-7

Author: Philipp John

Date: April 25, 2023

Task 1

Please visit <https://github.com/johnphilipp/HSG-WAS-SS23-exercise-7>

Task 2

Please visit <https://github.com/johnphilipp/HSG-WAS-SS23-exercise-7>

Task 3

1. How do the parameters α and β impact the performance of your algorithm (comparing the produced solution to the optimal solution)? Describe and interpret the behavior of your ant colony for different parameter values, while considering that the ant population, the number of iterations, and the evaporation rate remain fixed.

α determines the relative importance of the pheromone trail. A higher value of α means the ants will put more emphasis on the pheromone trail when selecting their next city to visit. This can lead to a stronger exploitation of the existing pheromone information and potentially faster convergence towards a good solution. However, if α is set too high, the algorithm may become too greedy and converge to suboptimal solutions.

β determines the relative importance of the heuristic information, which is the inverse of the distance between cities. A higher value of β means the ants will put more emphasis on the distance when selecting their next city to visit. This can lead to a stronger exploration of the solution space, as the ants are more likely to choose shorter paths. However, if β is set too high, the algorithm may become too focused on exploration and take longer to converge to a good solution.

In my case, I found a trade-off between exploration and exploitation to have the best effect. I also learnt that a higher value of a than b is better. For example, $a=2$ and $b=1$ (11555) was better than $a=1$ and $b=2$ (50801). Hence, my implementation seems to correlate with the behavior found in ants, where pheromone plays an important role.

2. How does the evaporation rate ρ affect the performance of your algorithm (comparing the produced solution to the optimal solution)? Describe and interpret the behavior of your ant colony for different evaporation rates, while considering that the ant population, the number of iterations, and the parameters α and β remain fixed.

The evaporation rate p affects the performance of the algorithm by controlling the rate at which the pheromone trails decay. A higher evaporation rate leads to faster decay of pheromone trails, which can promote exploration and prevent the ants from getting trapped in local optima. However, if p is set too high, the algorithm may lose valuable information from previous iterations and take longer to converge to a good solution. Conversely, a lower evaporation rate retains more pheromone information but may lead to premature convergence to suboptimal solutions due to excessive exploitation of existing trails. In my case the values are very similar. A lower p (tested with 0.05) also lead to a lower solution: 11303. A higher p (tested with 0.95) lead to a higher result: 13616.

3. How would you modify your implementation in order to apply the ACO algorithm to a dynamic traveling salesman problem (DTSP), i.e a TSP in which cities can be added or removed at run time?

Update the environment: The environment should be able to handle adding and removing cities during runtime. To do this, we will need to create methods for adding and removing cities within the environment, and update the existing environment data structures like distance maps and pheromone maps accordingly.

Reinitialize pheromone map: Whenever a city is added or removed, the pheromone map needs to be reinitialized to account for the updated environment. The pheromone values should be adjusted to reflect the new number of cities.

Update the ant behavior: Ants should be aware of the changes in the environment and adapt their behavior accordingly. For instance, ants should update their path selection and tour construction methods to account for the new set of cities.

Adapt the ACO algorithm: The main ACO algorithm should be able to handle the dynamic changes in the environment. This involves updating the main loop of the algorithm to consider the possibility of cities being added or removed, and adapting the solution generation and evaluation processes accordingly.

Incorporate real-time feedback: In a dynamic setting, it is crucial to incorporate real-time feedback to adapt the algorithm's behavior. For example, we can update the pheromone evaporation rate or the influence of distance and pheromone levels in path selection based on the changes in the environment. Pheromone would be outdated for certain routes if cities were removed.