# Algorithms and AI lab – week 1

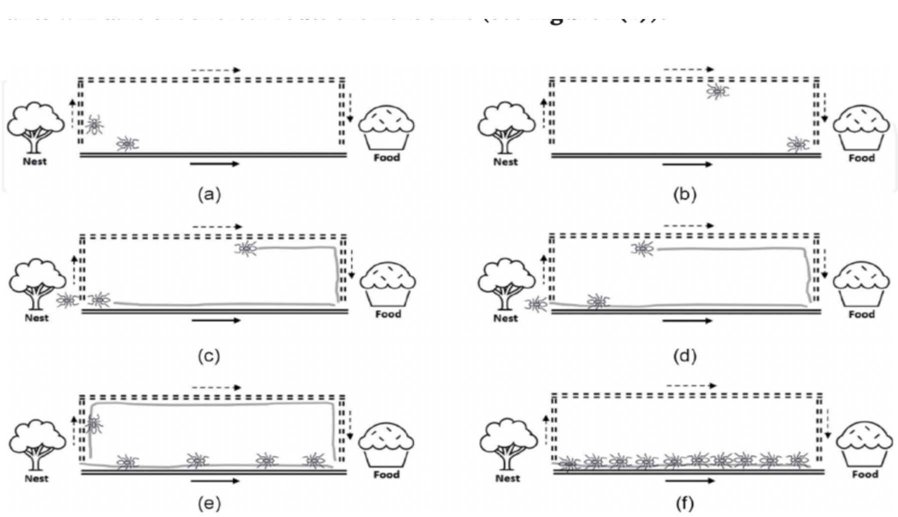
Specification document of Filip Ubović, exchange student, for subject Algorithms and AI Lab, taken in 3rd period of school year 2024/2025, on Master's Programme in Computer Science at University of Helsinki. The project language is English.

## Problem formulation

My initial idea was to create an algorithm for orders in a food delivery app. It is a case of travelling salesman problem, with many depots from which you can deliver, that are equipped with multiple trucks, to multiple stores that need a delivery. Academically the problem is named M-mTSP. After doing my research, I realized that the problem complexity is too high, since TSP is already an NP-hard problem, meaning that with number of clients that need a delivery, the problem complexity rises exponentially. I decided on solving an mTSP problem, meaning that you have many trucks in a **single** depot that can deliver to said stores. Next, there are minisum and minimax solutions to the problem. Minisum algorithms optimize the problem so that the sum of all the routes is minimal. Usual problem is that the workload doesn't get distributed evenly between the trucks (or salesmen, or bikes, etc…), but some routes can be significantly longer than others. Minimax aims to minimize the longest route, thus it is much more useful in practice. Even more useful in practice is if we aim to distribute the routes as evenly as possible. In that situation, the problem is called BmTSP (Balanced multiple travelling salesmen problem).

## The algorithm

Now the optimization I want to use is Ant colony optimization (ACO). The inspiration came from my courses Operations research 1 and 2 at University of Belgrade, where we have been though how to solve many optimization problems, learned about heuristics, meta-heuristics, etc. Algorithm/optimization that caught my eye was called ACO, and since then I have wanted to try it out. Idea of this meta-heuristic is as follows – ants while searching for food move and leave pheromones behind them, smelly substances, that help them and other ants orient and reach the food.



Imagine that we have food with two possible paths reaching it. Initially, all paths (i.e. vertices of a graph) have the same amount of pheromone. Two ants start and pick a different path. Since pheromone also evaporates, the one who came back, left more pheromone more recently, so the new ants are likely to start taking that path. That is how the colony finds it way (Soofastaei, 2022).

Since it is a heuristic, it doesn't guarantee an optimal solution, but it finds a solution in a shorter amount of time than other exact algorithms.

I have read research on many different modifications of ACO, and picked ACO-BmTSP solution proposed by a team of researchers in the article *Ant-Balanced Multiple Traveling Salesmen: ACO-BmTSP* released in the journal *Algorithms*. The algorithm seems promising as it has an objective function that tries to minimize the distance while keeping the tours balanced (de Castro Pereira, Solteiro Pires, & de Moura Oliveira, 2023), it solves problems in acceptable time, and doesn't look way too hard to implement, taken into consideration that I have never implemented an algorithm of this kind. I am still yet to fully grasp how it works.

## Programming languages

For implementing this algorithm, I will probably use Python. Since these kinds of algorithm requires lots of computing, I am thinking of doing it in Rust, if I have enough time on my hands to learn it. But Python also offers many more ready to use libraries.

I can read code in C, C#, Java, Haskell and Python.

## Input and output

Inputs should be:

1) time needed to go from depot to each store,
2) time needed to go between stores that are in proximity to each other,
3) number of trucks
4) average time spent with the client, per delivery
5) setting ACO parameters

Why time and not distance? Usually, due to the traffic in the city, the distances are irrelevant. Longer routes can take less time. People's working day is limited, not the distance that they can traverse.

Future versions of algorithm may introduce a coefficient which calculates the worth of taking a more distant route and spending more fuel, even if it takes less time.

Outputs should be:

1) routes for every truck
2) time needed for each route

## References


Soofastaei, A. (2022). *Introductory Chapter: Ant Colony Optimization.*

de Castro Pereira, S., Solteiro Pires, E., & de Moura Oliveira, P. (2023). Ant-Balanced Multiple Traveling Salesmen: ACO-BmTSP. *Algorithms*, 16.