# Object Oriented Programming 2022/23

Travelling salesmen problem by ant colony optimization

LEEC/MEEC – IST

## 1 Problem

The Traveling Salesman Problem (TSP) is a crucial optimization problem belonging to a class of NP-complete problems. There is no known efficient (polynomial time) algorithm to solve any NP-complete problem. Therefore, approximations to the optimal solution are necessary. Ant colony optimization is a promising approach in this regard. The TSP has several practical applications, including planning, logistics, and microchip manufacturing.

The TSP consists in finding the shortest cycle in a weighted graph that passes through all its nodes (only once). A *weighted graph* is a triple $G = (N, E, \mu)$ composed by a finite set of nodes $N$, a set of edges $E$, where an edge is represented by a set of two nodes, and a function $\mu : E \to \mathbb{R}^+$ that weighs each edge. As an example, consider the weighted graph depicted in Figure 1.

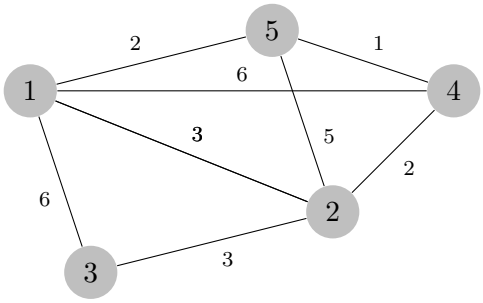


Figura 1: Example of a weighted graph.

A node is said to be *adjacent* to another node if an edge connects them. A *path* is a sequence of nodes $n_1, \ldots, n_k$ such that, for every $i = 1, \ldots, k-1$, node $n_i$ is adjacent to node $n_{i+1}$. The weight of a path is the sum of the weights of the edges connecting consecutive nodes, that is,

$$\mu(n_1, \ldots, n_k) = \sum_{i=1}^{k-1} \mu(\{n_i, n_{i+1}\}).$$

Finally, a *cycle* is a path $n_1, \ldots, n_k$ such that $n_1 = n_k$. A cycle is said to be *Hamiltonian* if it contains all nodes and they occur only once in the cycle, except for $n_1$, which occurs twice. The TSP aims at finding the shortest Hamiltonian cycle in a weighted graph.

## 2 Ant colony optimization algorithms

The algorithms based on *ant colony optimization* (ACO) appeared recently for approaching complex graph optimization problems. The idea is to simulate ants that randomly traverse the graph and lay down pheromone trails upon finding a solution to the problem. The wide variety of algorithms, either for optimization or other hard problems, seeking self-organization in biological systems has led to the concept of *swarm intelligence*, which is a very general framework in which ant colony algorithms fit.

In the case of the TSP, the following rules govern the ACO algorithm:

1. Ants do not visit twice the same node, except the starting node, also called the *nest*. Therefore, an ant needs to store the path it traversed so far.

2. Let $i$ be the current node of the ant and $J$ the set of non-visited nodes adjacent to $i$. If $J$ is not empty, the ant randomly chooses one node $j$ in $J$ with probability proportional to the pheromone level of the edge connecting $i$ to $j$, and inversely proportional to the weight of this edge. Thus, if the ant is moving from node $i$ and it can move to nodes $J = \{j_1, \ldots, j_\ell\}$ the probability of moving to $j_k \in J$ is given by

$$P_{ij_k} = \frac{c_{ij_k}}{c_i},$$

   where:

   - $c_{ij_k} = \frac{\alpha + f_{ij_k}}{\beta + a_{ij_k}}$,
   - $c_i = \sum_{k=1}^{\ell} c_{ij_k}$,
   - $f_{ij_k}$ is the pheromone level of the edge connecting $i$ to $j_k$,
   - $a_{ij_k}$ is the weight of the edge connecting connecting $i$ to $j_k$, and
   - $\alpha$ and $\beta$ are input parameters.

   Upon moving, the ant should update its path consistently.

3. If an ant has no choice but to visit a node already visited ($J = \emptyset$), it randomly chooses any adjacent node with a uniform distribution. Afterward, it should update its path by removing the cycle created in the last move. As an example, consider the graph in Figure 1. If the ant contains the path $1 \to 2 \to 3$, it can only visit nodes 1 or 2, both already visited. One of these nodes should be picked up evenly; suppose the ant chooses to visit node 2. The path of the ant should remove the cycle $2 \to 3 \to 2$, and thus update its path to $1 \to 2$.

4. The time an ant takes to traverse an edge between nodes $n_i$ and $n_j$ has an exponential distribution with mean value proportional to the weight of the edge, that is, with mean value $\delta \times a_{ij}$ where $\delta$ is an input parameter.

5. Upon completing a Hamiltonian cycle, and only in this case, the ant lays down pheromones in all edges constituting the cycle. The level of pheromones is inversely proportional to the weight of the cycle. In detail, the ant increments the level of pheromones at the edges of the cycle by

$$\frac{\gamma W}{\mu(n_1, \ldots, n_k, n_1)}$$

   units where:

- $W = \sum_{e \in E} \mu(e) = \sum_{i,j} a_{ij}$ and
- $\gamma$ is an input parameter.

After finding the Hamiltonian cycle, the ant restarts, traversing the graph from the nest to find another one.

6. The level of pheromones of each edge is initialized to zero units.

7. The pheromone trail evaporates over time, thus reducing its attractive strength. This is performed by successive decreases in the pheromone level at the edges of the graph. The pheromones of each edge evaporate independently, and this evaporation occurs in discrete steps; in each step, the pheromone level of the edge is reduced by $\rho$ units. The time between evaporations has an exponential distribution with mean value $\eta$, where $\eta$ is an input parameter.

# 3 Approach

This project aims to give a solution in UML and Java to the TSP using the above ACO algorithm.

The simulation consists of generating at time zero an ant colony with $\nu$ ants and simulating this colony until the final instant $\tau$. It is assumed that the node $n_1$ representing the nest is an input parameter.

During the simulation, an ant will traverse the graph to find a Hamiltonian cycle and return to the nest, as described in Section 2. The shortest Hamiltonian cycle discovered until the simulation's current time should be stored to be provided to the user whenever needed. The simulation is guided through Discrete Stochastic Simulation (DSS) with the following discrete events:

- Ant move: An ant randomly chooses which node will move according to rules 1-3 in Section 2. The time to traverse the edge from node $i$ to node $j$ has an exponential distribution with mean value $\delta \times a_{ij}$.

- Evaporation of pheromone: All edges with positive levels of pheromones trigger the respective evaporation event. The level of pheromones is reduced by $\rho$ units with an exponential distribution with mean $\eta$ between evaporations (according to rule 7 in Section 2).

The simulation ends when the next event to simulate occurs after the final instant $\tau$.

**Incomplete version** This is a preliminary version of the project description. Start by getting familiar with the problem, but there will still be a clarification of the project in a lecture class, explanation and support slides for discrete stochastic simulation, and the final outlining of the expected project parameters and results.