Universidade do Minho
Escola de Engenharia
Departamento de Informática

Mestrado Integrado em Engenharia Informática
Mestrado em Engenharia Informática
Computação Natural
2019/2020

Paulo Novais, Cesar Analide, Filipe Gonçalves

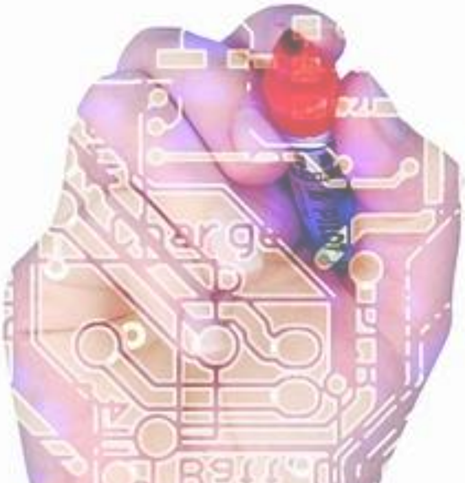
Synthetic Intelligence Lab

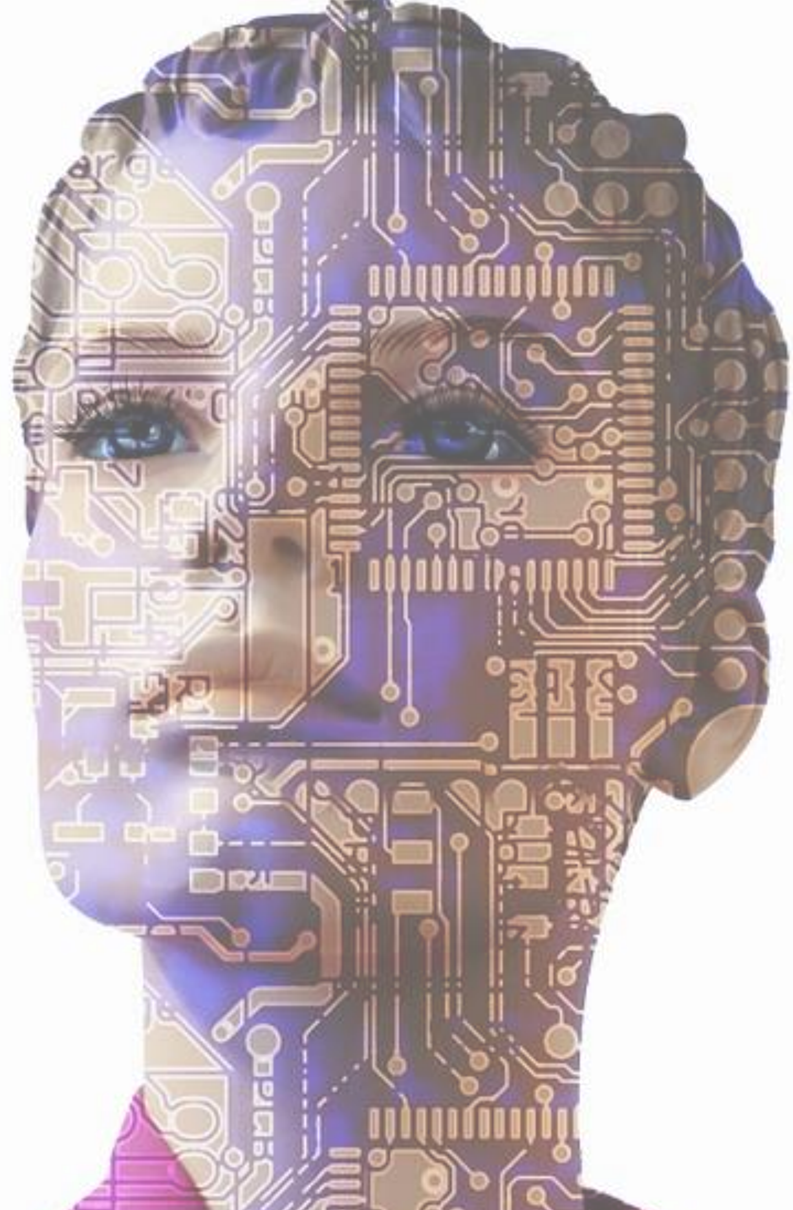
Computação Natural@ 2019/2020

- Paulo Novais – pjon@di.uminho.pt
- Cesar Analide – cesar.analide@di.uminho.pt
- Filipe Gonçalves – fgoncalves@algoritmi.uminho.pt

- Departamento de Informática
  Escola de Engenharia
  Universidade do Minho
- Grupo ISLab – (Synthetic Intelligence Lab)
- Centro ALGORITMI
  Universidade do Minho

Ant Colony Optimization





- A meta-heuristic is a heuristic method to solve generic optimization problems
- Meta-heuristics are generally applied to problems for which no efficient algorithm is known
- They use a combination of random choices and historical knowledge of previous results acquired by the method to guide and search the search space in neighborhoods within the search space, which avoids great locations

- Throughout time, different metaheuristic methods have been proposed, and there are currently hundreds of alternative methods, although sharing the fundamental characteristics
- Some of the most relevant are:
    - Particle swarm optimization
    - Genetic algorithms
    - Simulated annealing
    - Tabu search
    - Artificial immune systems
    - Ant colony optimization

- The meta-heuristic Ant Colony Optimization (ACO) is based on the actual behavior of the ants
  - Behavior allows to find the smallest path between a food source and its colony
  - This phenomenon occurs because, during its trajectory, the ants deposit in the way a substance called pheromone. When choosing a trajectory, they choose the one that has the highest amount of pheromone, since it is the trajectory that the largest number of ants has already performed
  - It suggests that it is the best trajectory, either because it is the shortest one or the safest trajectory (e.g., that it avoids predators)

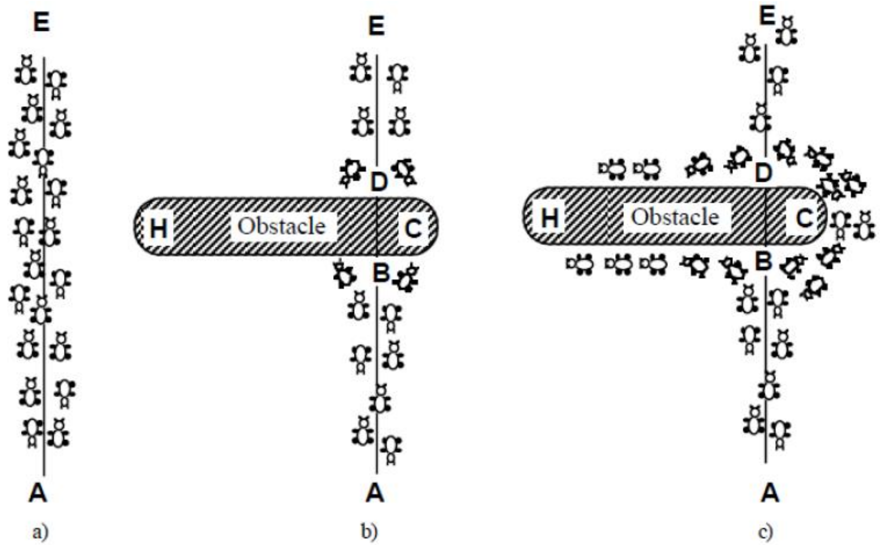


Fig. 1. An example with real ants.

a) Ants follow a path between points A and E.
b) An obstacle is interposed; ants can choose to go around it following one of the two different paths with equal probability.
c) On the shorter path more pheromone is laid down.

M. Dorigo, V. Maniezzo and A. Colorni, "Ant system: optimization by a colony of cooperating agents," in *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 26, no. 1, pp. 29-41, Feb 1996.

Synthetic Intelligence Lab

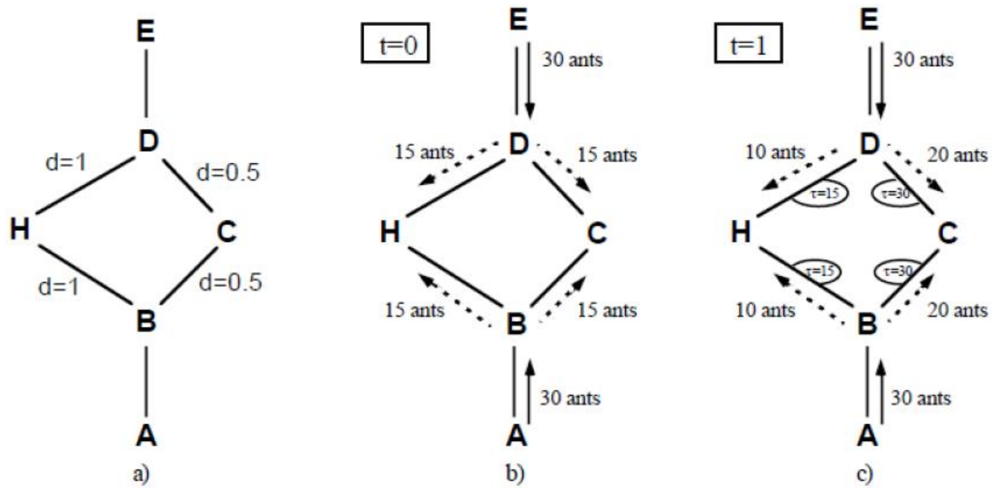


Fig. 2. An example with artificial ants.

a) The initial graph with distances.
b) At time t=0 there is no trail on the graph edges; therefore, ants choose whether to turn right or left with equal probability.
c) At time t=1 trail is stronger on shorter edges, which are therefore, in the average, preferred by ants.

M. Dorigo, V. Maniezzo and A. Colorni, "Ant system: optimization by a colony of cooperating agents," in *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 26, no. 1, pp. 29-41, Feb 1996.

- The fundamental idea is that, if at some point an ant has to choose between different paths, those that were more often chosen by previous ants are more likely chosen
- Thus, paths with large influx of ants are the shortest paths

- The ACO has been applied in a number of areas, including:
    - Scheduling issues
    - Vehicle Mobility Problems
    - Image Processing
    - Classification Problems
    - Etc.

- However, the application domains where the ACO has been shown to be more robust are:
    - As heuristic for initial solution generation for other more complex methods
    - In solving problems related to graphs and distances
        - E.g., Travelling Salesman Problem (TSP)

- **Traveling Salesman Problem (TSP)**
  - Set of N cities/nodes
  - Matrix (N x N) of distances
  - **Goal** - run all cities/nodes once and return to the starting city, minimizing the total distance traveled
  - **How many admissible solutions? (n-1)!**
    - 5 cities ⇒ 4! = 24 possible routes
    - 10 cities ⇒ 9! = 362,880 possible routes
    - 25 cities ⇒ 24! = 6.2 x 1023 possible paths

- At first the ants are left free to choose the path. No pheromone exist yet
- The ants converge to one of the paths with equal probability
- Due to random fluctuations, one of the paths will have more pheromone and will attract the ants more likely
- Using different sized paths, the ants converge to the shortest
    - The short path is traveled in less time, causing more ants to walk through it at the same time. Therefore, more pheromone is deposited
    - Ants choose, with greater probability, the shortest path (with more pheromone).
    - Ants that use the smallest path come and go faster
        - Pheromone deposition occurs on the smallest path
        - In the end, the ants always use the smallest path
        - The shortest route of all visited ends up being the one that is traveled in less time, soon end up receiving more quantity of pheromone, attracting more and more ants

- The ant colony optimization algorithm "mimics" the behavior of an ant colony by successively "launching" a number of sub-colonies (NS) each with a certain number of ants (NA)
- Each ant has to build an admissible solution to the problem at hand. When all the ants of the colony have completed their solution, the pheromone rails are updated

```
for (s=1; s <= NS; s++)
   for (f=1; f <= NA; f++)
      Ant f builds a solution
   Update the pheromone trails:
```

$$\underbrace{\tau(i,k)}_{\text{Pheromone level between nodes i and k}} \leftarrow \underbrace{(1-\rho)\ \tau(i,k)}_{\text{Evaporation of some amount of pheromone}} + \underbrace{\sum \Delta\tau_f(i,k)}_{\text{Deposition of pheromone between nodes i and k by the ants that used the arc (i, k)}}$$

- In each iteration, the ants decide, according to a probabilistic measure, which way to go, among the unvisited ones
- Each ant has a memory, also called the tabu list, which stores the paths traveled and prevents a path from being visited by the same ant more than once
- The probability of choosing a path is proportional to the pheromone trait and its attractiveness - which varies according to the type and modeling of the problem in which the algorithm is being applied
- If the ant has already traveled that way, the probability of choice is zero; otherwise, it is positive.

- The choice of the path to be followed by the ant (construction of the solution) is done according to the Probability of Transition between paths:

Pheromone Quantity

Path Attractiveness (i.e., 1/distance)

$$p_k(r,s) = \begin{cases} \dfrac{\left[\tau_{(r,s)}\right]^{\alpha} \cdot \left[\eta_{(r,s)}\right]^{\beta}}{\sum\limits_{u \notin M_k} \left[\tau_{(r,u)}\right]^{\alpha} \cdot \left[\eta_{(r,u)}\right]^{\beta}}, & \text{IF } s \notin M_k \\ 0, & \text{OTHERWISE} \end{cases}$$

Parameter to control Pheromone Influence

Parameter to control Attractiveness Influence

- If a path has not been traversed by ants, the pheromone deposition on it is zero; otherwise it is positive
- It is said that an ant has changed from "state" when moving from one solution to another

- At the end of each iteration, an evaporation rate removes part of the pheromone, reducing the amount of the substance in the pathways. This prevents the ants from:
  - Getting trapped in great locations
  - Decreases the likelihood of choosing paths that have not been used recently
- The deposition of the pheromone is done at the end of each iteration
- It is considered that, at the end of each iteration, the ants do the inverse path of the solution generation, but, this time, depositing the pheromone. The amount of pheromone associated with each path represents the learning of the colony during the course of the algorithm.

- In the update of the pheromones, two events occur:
  - The evaporation
    - Prevents the accumulated pheromone from growing indefinitely
    - Allows to forget past research decisions
  - The pheromone deposit of all ants

Pheromone Quantity deposited by Ant k

$$\tau_{xy} \leftarrow (1-\rho)\tau_{xy} + \sum_k \Delta\tau_{xy}^k$$

Pheromone Quantity

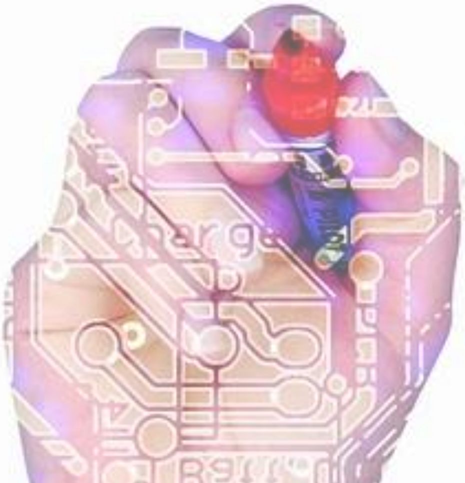
Pheromone Evaporation Coefficient

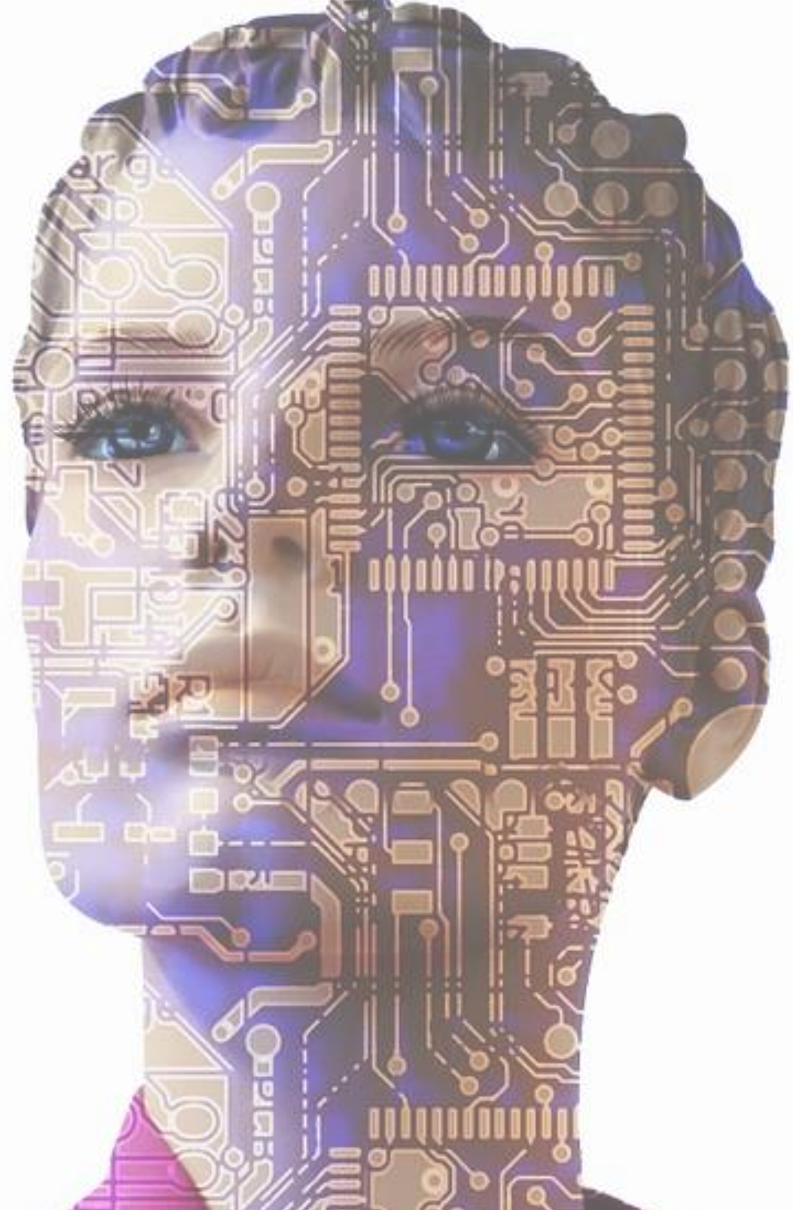
Constant Distance

$$\Delta\tau_{xy}^k = \begin{cases} Q/L_k & \text{if ant } k \text{ uses curve } xy \text{ in its tour} \\ 0 & \text{otherwise} \end{cases}$$

- Dorigo, M., Stutzle, T., (2003), The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances. In Glover, F., Kochenberger, G. Handbook on Metaheuristics, Kluwer, Cap. 9.
- Dorigo M., Di Caro, G., Cambardella L. M., (1999), Ant Algorithms for Discrete Optimization. Artificial Life, 5(2):137-172. Also available as Technical Report No. 98-10 (IRIDIA), Université Libre de Bruxelles, Belgium.
- Dorigo, M., ManiezzoV. and Colorni, A."Ant system: optimization by a colony of cooperating agents," in IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 26, no. 1, pp. 29-41, Feb 1996.
- Ant Colony Optimization page, developed by Marco Dorigo: http://iridia.ulb.ac.be/~mdorigo/ACO/index.html

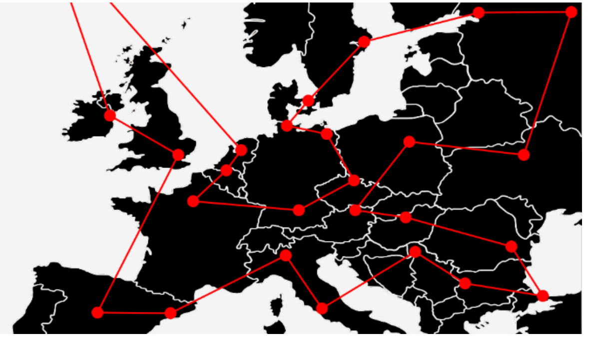
Exercise





Computação Natural@ 2019/2020

- **Objective:** Implement in Python an Ant Colony Optimization algorithm to solve the Traveling Salesman Problem (explained previously)
- **Support Material:** Python script ACO.py presents a small environment and initial configurations
    - 5 Cities
    - 5 Individuals
    - 100 iterations



- **Goal:** Complete the script and study algorithm learning process
- **Analyse first:**
    - Tutorial: https://towardsdatascience.com/using-ant-colony-and-genetic-evolution-to-optimize-ride-sharing-trip-duration-56194215923f
    - Video Guideline: https://www.youtube.com/watch?v=EJKdmEbGre8

Universidade do Minho
Escola de Engenharia
Departamento de Informática

Mestrado Integrado em Engenharia Informática
Mestrado em Engenharia Informática
Computação Natural
2019/2020

Paulo Novais, Cesar Analide, Filipe Gonçalves