3.2 P-metaheuristics
Ant Colony Optimization (ACO)

P-metaheuristics – Ant Colony Optimization (ACO)

Swarm intelligence

Swarm intelligence refers to the social behavior of a group of particles that compete for foods. These particles are simple and non sophisticated agents, they *cooperate* by an indirect communication medium, and do movements in the decision space. Ant Colony Optimization is one example of swarm intelligence based algorithms.

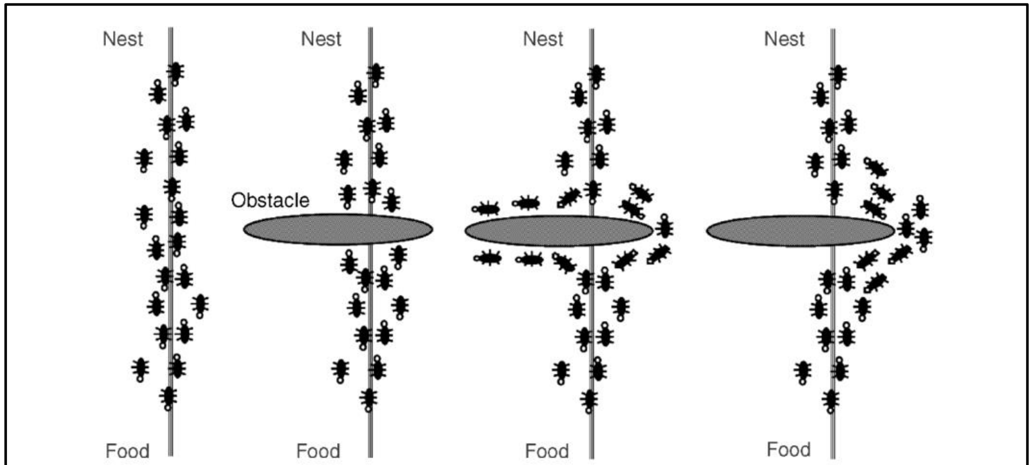
P-metaheuristics – Ant Colony Optimization (ACO)

Swarm intelligence

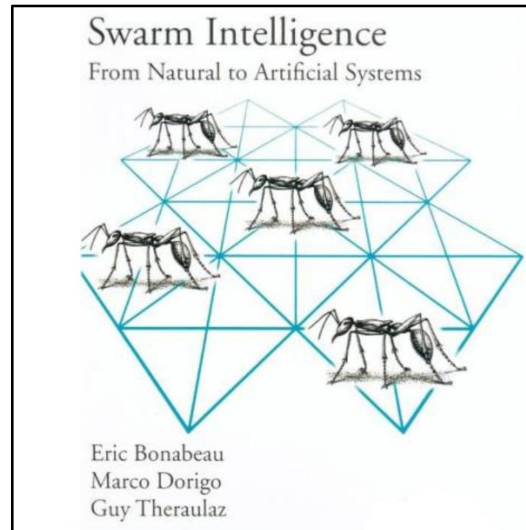
Swarm intelligence refers to the social behavior of a group of particles that compete for foods. These particles are simple and non sophisticated agents, they *cooperate* by an indirect communication medium, and do movements in the decision space. Ant Colony Optimization is one example of swarm intelligence based algorithms.

Ant Colony Optimization

Imitate the cooperative behavior of real ants to solve optimization problems.



Inspiration from an ant colony searching an optimal path between the food and the nest.



P-metaheuristics – Ant Colony Optimization (ACO)

Template of the ACO.

Initialize the pheromone trails ;
Repeat
  For each ant **Do**
    Solution construction using the pheromone trail ;
    Update the pheromone trails:
      Evaporation ;
      Reinforcement ;
Until Stopping criteria
Output: Best solution found or a set of solutions.

P-metaheuristics – Ant Colony Optimization (ACO)

Ant colony algorithm for the TSP problem (ACO–TSP).

Memorize characteristics of good solutions

Initialize the pheromone information ; $n \times n$ matrix τ initialized by the same values

Repeat

For each ant **Do** **[Generate M ants]**

Solution construction using the pheromone trails:

$S = \{1, 2, \ldots, n\}$ /* Set of potentially selected cities */

Random selection of the initial city i ; $S = S - \{i\}$

Repeat

- α , β: relative influence in the solution construction
- α = 0, closest cities more likely selected
- β = 0, pheromone will guide search but rapid stagnation

Select new city j with probability $p_{ij} = \frac{\tau_{ij}^{\alpha} \times \eta_{ij}^{\beta}}{\sum_{k \in S} \tau_{ik}^{\alpha} \times \eta_{ik}^{\beta}}$; η_{ij} equal to $1/d_{ij}$

$j \in S$; not yet visited solutions

$S = S - \{j\}$; $i = j$;

Until $S = \emptyset$

End For

Update the pheromone trail:

For $i, j \in [1, n]$ **Do**

$\tau_{ij} = (1 - \rho)\tau_{ij}$ /* Evaporation */ ; $\rho \in]0, 1]$ represents the reduction rate of the pheromone

For $i \in [1, m]$ **Do**

Reinforcement: Each ant leaves pheromone according to its solution quality

$\tau_{i\pi(i)} = \tau_{i\pi(i)} + \Delta$

$\Delta = 1/f(\pi)$

Until Stopping criteria

Output: Best solution found or a set of solutions.

At least two Quality-based pheromone update strategies:
- Elitist update -> best ant, *m=1*
- Best K update -> best k ants, *m=k, k < M*

P-metaheuristics – Ant Colony Optimization (ACO)

- Lab session 

Propose your own implementation of the ant colony optimization algorithm for the traveling salesman problem then use a python API from your choice. Compare the obtained results by the two versions.