CHAP III
Ant Colony Optimization

Ant Colony Algorithm.
1 General

- Ant Colony Optimization, ACO, (Ant Colony Optimization).
- All his abstract ideas are inspired by Deneubourg's work on ants.

- Ant colony optimization (ACO) simulates artificial systems that take inspiration from the behavior of real ant colonies.
- Used to solve discrete optimization problems.

Historical

- This meta-heuristic is relatively recent.
- It was introduced in 1991 by Colorni , Dorigo and Maniezzo to solve the Traveling Salesman problem.
- It became popular , then was improved in 1995 and was successfully applied to other combinatorial optimization problems in 1994.

Ant Colony Algorithm.
2 Definitions

- Ant colony algorithms are algorithms inspired by the behavior of ants and constitute a family of optimization metaheuristics.

- An ant colony algorithm is an iterative population algorithm where all individuals share common knowledge that allows them to guide their future choices and indicate directions to other individuals to follow or, on the contrary, to avoid.

Ant Colony Algorithm.
3 How it works

▪ The original idea comes from the observation of the exploitation of food resources by ants. Indeed, these, although individually having limited cognitive capacities, are collectively capable of finding the shortest path between a food source and their nest.

A model explaining this behavior is as follows:

1. An ant roams the environment around the colony more or less randomly.
2. If it discovers a food source, it returns more or less directly to the nest, leaving a trail of pheromones along its way.
3. these pheromones being attractive, the ants passing nearby will tend to follow, more or less directly, this trail.

Ant Colony Algorithm.
3 How it works

4. Returning to the nest, these same ants will *reinforce* the trail.
5. If two trails are possible to reach the same food source, the shorter one will, in the same time, be traveled by more ants than the long trail.
6. The short track will therefore be increasingly strengthened, and therefore increasingly attractive.
7. The long trail, on the other hand, will eventually disappear, as pheromones are volatile;
8. In the end, all the ants have therefore determined and “chosen” the shortest path.

Ant Colony Algorithm.
3 How it works

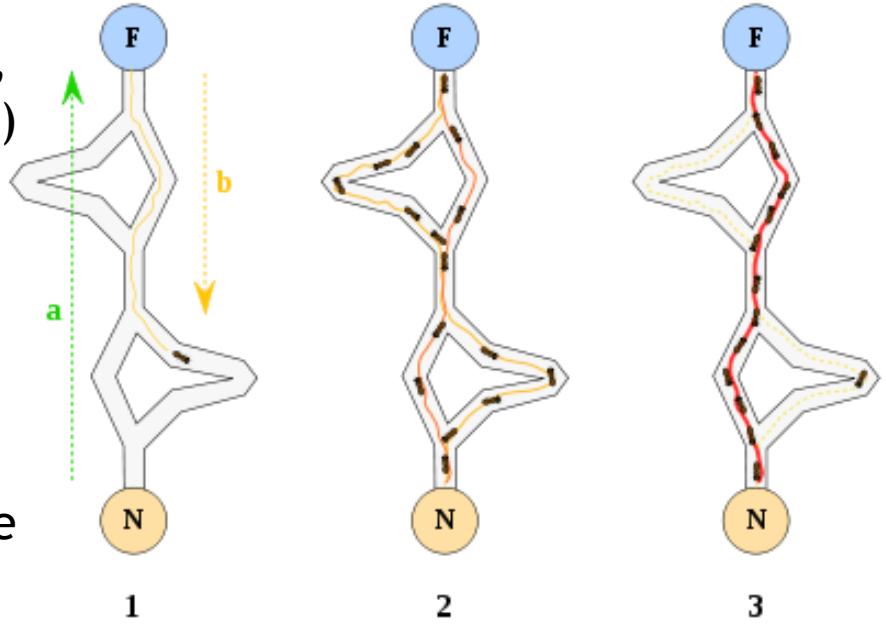
- Ants use the environment as a means of communication: they indirectly exchange information by depositing pheromones, all of which describe the state of their "work". The information exchanged has a local scope; only an ant located at the place where the pheromones were deposited will have access to it. This system is called " stigmergy ", and is found in several social animals (it has been studied in particular in the case of the construction of pillars in termite nests).

Ant Colony Algorithm. Principle

P1) the first ant finds the food source (F), via any path (a), then returns to the nest (N) leaving behind a pheromone trail (b).

P2) ants take all four possible paths indifferently, but the reinforcement of the trail makes the shortest path more attractive.

P3) ants take the shortest path, long portions of other paths lose their pheromone trail.



4. Ant colony algorithm for the traveling salesman problem TSP:

Until the stopping criterion is reached do

For k=1 to m do

Choose a city at random

For each city not visited i do

Choose a city j, from the list of remaining cities according to (P1)

End For

Place a track on the path (t) according to (P2)

End For

Evaporate the tracks according to (P3)

End As long as

4. Ant colony algorithm for the traveling salesman problem TSP:

- The initial goal of the method was to solve the traveling salesman problem. The algorithm presents the method proposed by the authors. Considering a traveling salesman problem with n cities, each ant k travels the graph and constructs a path of length n.

For each ant, the path from city i to city j depends on:

- the list of cities not yet visited, which defines the possible movements at each step, when ant k is on city i, N_i^k
- the inverse of the distance between cities $\eta_{ij} = \frac{1}{d_{ij}}$, called visibility.

This information is used to direct ants to nearby cities and thus avoid long journeys.

- the amount of pheromone deposited on the edge connecting two $\tau_{ij}(t)$ is called the intensity of the trail. This quantity defines the attractiveness of a trail and is modified after the passage of an ant. It is the pseudo-memory of the system.

4. Ant colony algorithm for the traveling salesman problem TSP :

- The movement rule is as follows:

$$P_{i,j}^{k}(t)=\begin{cases}\dfrac{[\tau_{ij}(t)]^{\alpha}.\ \eta_{ij}^{\beta}}{\sum_{l\epsilon N_i^k(t)}[\tau_{il}(t)]^{\alpha}.\ \eta_{il}^{\beta}} & si\ j\in N_i^k \\ 0 & sinon\end{cases}$$

P1

α and β are two parameters that control the relative importance between pheromones and visibility. After a complete turn, each ant deposits a quantity of pheromone ($\Delta\tau_{ij}^{k}$) over its entire path. This quantity depends on the quality of the solution found and is defined by:

4. Ant colony algorithm for the traveling salesman problem TSP :

$$\Delta\tau_{ij}^{k} = \begin{cases} \dfrac{Q}{L_k(t)} & \text{si } (i,j) \in T_k(t) \\ 0 & si\ non \end{cases}$$

P2

où $T_k(t)$ est le trajet effectué par la foumi k à l'itération t, $L_k(t)$ est la longueur de $T_k(t)$ et Q est un paramètre fixé.

Enfin, il est nécessaire d'introduire un processus d'évaporation des phéromones. En effet, pour éviter de se faire piéger dans des solutions sous-optimales, il est nécessaire qu'une fourmi oublie les mauvaises solutions. La règle de mise à jour est donc :

$$\tau_{ij}(t+n) = (1-\rho).\tau_{ij}(t) + \Delta\tau_{ij}(t)$$

P3

Où $$\Delta\tau_{ij}(t) = \Sigma_{k=1}^{m} \Delta\tau_{ij}^{k}(t)$$

Et m est le nombre de fourmis

Ant Algorithm Variants

- ***The Elitist Ant Algorithm***
  - Proposed by M. Dorigo , Maniezzo and Colori (1996)
  - The best trick found up to the current iteration receives an extra pheromone

- ***ranking -***
  - Proposed by M. Bullnheimer , Hartl and Strauss (1999)
  - Ants are sorted according to their constructed solution lengths, pheromone updating is done according to the contribution of each ant

- ***Best local round'***
  - Process by M. Tony , Simon and Terri (2003)
  - Elitist Ant algorithm , each ant keeps its best local tower and reinforces it in updating pheromones at each iteration.

- ***The 'Max Min' Ant Algorithm***
  - Method by M. Stutzle and Hoos (2000)
  - An explicit limit is imposed on the pheromone
  - Pheromones are initialized to the upper bound

- ***Implementation of the hypercube framework in the ant colony system***
  - Proposed by M. Dorigo and Chritian (2004)
  - Implementation of the ant algorithm in a hypercube space

5. Domains application :

- Applications to the symmetric and asymmetric traveling salesman problem.
- Applications to the sequential scheduling problem.
- Applications to quadratic assignment problems.
- Applications to vehicle routing problems.
- Applications to scheduling problems.
- Applications to graph coloring problems.
- Applications to partitioning problems.
- Applications to telecommunications networks.
- Parallel implementations.

6. Conclusion:

The ant colony algorithm is a general heuristic used to solve various combinatorial analysis problems.
Main disadvantage: relatively high cost of generating solutions.
It begins to be adapted to ongoing problems.

Bibliography :

1. ***fr.wikipedia.org***
2. ***Optimization by ant colonies. COSTANZO Andrea. LUONG Thé Van. MARILL Guillaume .***
3. ***http://khayyam.developpez.com/articles/algo/voyageur-de-commerce/colonies-de-fourmis /***