Machine Learning

Ant Colony Optimization

“All good work is done the way ants do things: Little by little…”
- Lafcadio Hearn

What is Ant Colony Optimization?

What is ACO?

Ant Colony Optimization (**ACO**) is a population-based search technique for solving combinatorial optimization problems inspired by the behavior of ants.

Ants are able to create a near optimal path between their nest and a food source using very meager “computational power”.

Ants as Agents

Ants are:

- Blind (almost)
- Incapable of solving complex problems individually
- Reliant upon "swarm intelligence" to achieve goals
- Able to communicate with each other via pheromones

Swarm Intelligence

A **Swarm Intelligence** (**SI**) is a collective system capable of accomplishing difficult tasks in dynamic and varied environments without any external guidance or control and with no central coordination.

A SI is capable of accomplishing a collective task which could not be achieved by a single agent acting alone.

Stigmergy

Stigmergy is a communication method used in decentralized systems in which individuals communicate with each other by changing the surrounding environment.

Ants use stigmergic communication via the pheromone trails they lay down while foraging for food.

Stigmergy (2)

Watch this video:
https://www.youtube.com/watch?v=5CAjWaZx2Ks

Humans use stigmergy too!

Matt Johnson, Ph.D. 7

Autocatalyzation

A chemical reaction is said to be **autocatalytic** if one of the reaction products is also a catalyst for the same or coupled reactions.

The stigmergic communication between ants is an autocatalytic process.

What does this mean?

Matt Johnson, Ph.D. 8

Autocatalyzation (2)

When an ant scout sets out to find food, it leaves behind a small pheromone trail.

If food is found, the scout lays down heavier pheromone upon its return to the nest.

Ant pheromone evaporates over time. Its half-life is just a few minutes.

Matt Johnson, Ph.D. 9

Autocatalyzation (3)

Other ants forage for food by choosing a trail with a probability based upon the strength of the pheromone deposit.

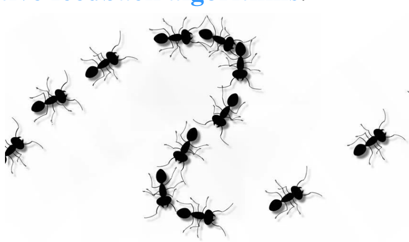
An ant then follows the selected trail, leaving behind its own pheromone on its way to the food and back again to the nest.

The more ants who follow a given trail, the stronger that trail's scent becomes.

Pheromone therefore builds up on shorter paths faster as it doesn't have as much time to evaporate away completely. Therefore, more and more ants will follow it.

Autocatalyzation (4)

ACO algorithms are called **autocatalytic positive feedback algorithms**.

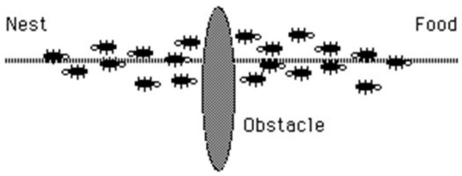


Path Optimization

An ant superhighway

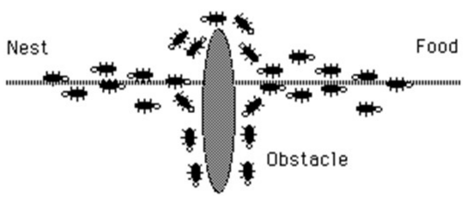
Path Optimization (2)

An obstacle appears…



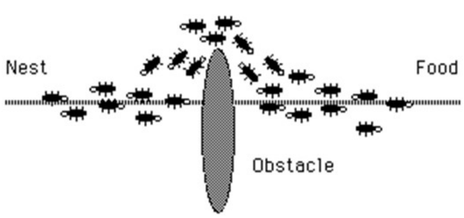
Path Optimization (3)

The correct path to choose is a 50/50 guess…



Path Optimization (4)

The shortest path is reinforced



Artificial Ants vs. Real

Unlike the natural precedent, ACO problems have:

- Memory in the form of data structures
- Sensory information about the environment beyond just pheromonal levels
- Discrete time increments
- Well-defined problem statements (optimization)

Development of ACO

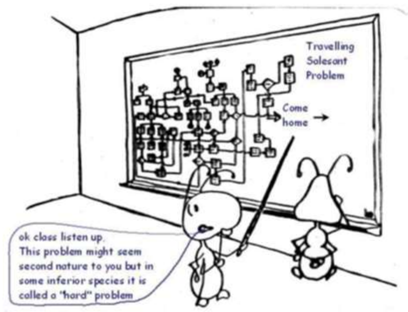
- Pierre-Paul Grass discovered “significant stimuli” beneficial to both the individual ant as well as the colony, and coined the word stigmergy in 1959
- In the 1980s, the collective behavior of ants was studied by Deneubourg and others through use of a **double bridge experiment**
- ACO initially proposed by Marco Dorigo in his doctoral thesis in 1992

“An ant on the move does more than a dozing ox.”

- Lao Tzu

ACO and the Traveling Salesman Problem

ACO and TSP



ACO and TSP (2)

Ant Colony Optimization was first applied to the Traveling Salesman Problem:

- It's a natural fit since TSP is a routing problem
- You need to go somewhere efficiently, then come back again…

Some aspects of TSP are very different from ant behavior though:

- TSP needs to visit several places
- TSP needs to form a cycle without repeats

Ant System for the TSP

The idea behind each iteration of the algorithm is:

- Each of the *m* ants in the colony starts in a different (or random) city
- Each ant then constructs its own tour of the *n* cities independently
- After a tour is complete, each ant lays down its pheromone on the edges of the graph it travelled. The shorter its tour, the more pheromone is applied
- Some evaporation of all pheromone then takes place

Constructing Tours

How do Ants construct their tours?

- Each ant at city i chooses to travel to the next city j with probability $p(i, j)$. This is a function of both the distance between the two cities and the strength of the pheromone along the path connecting them
- Transitions to already visited cities are not allowed

Tabu List

Each ant must keep a **tabu list** of cities it has already visited while constructing a tour.

- At the beginning of each iteration, the tabu list is initialized with the ant's starting city (as it can't revisit this as one of the n cities in the tour)
- When a new city is visited, it is added to list
- The final step of the ant's tour is to return to its starting city, which is the $(n+1)^{th}$ city visited. Thus, the ant's starting city needs to be remembered separately from the tabu list

Matrices

The TSP Ant System requires the use of three matrices:

- A distance matrix $d[n, n]$
- A visibility matrix $\eta[n, n]$
- A pheromone matrix $\tau[n, n]$

Distance Matrix

The **distance matrix** $d[n, n]$

- This stores the distances between all cities in the graph
- It's needed to calculate the length of the tour found by each ant
- Remember that the tour must go back to the starting city!

Visibility Matrix

The **visibility matrix** $\eta[n, n]$

- This is calculated as $1/d[n, n]$
- This is the heuristic used for guiding the construction of each ant's tour
- It's a greedy heuristic: the nearest town is best
- *Emphasizes **local** goodness*

Pheromone Matrix

The **pheromone matrix** $\tau[n, n]$

- This stores the pheromone strength of the path between every two cities
- $\tau[i, j]$ is the favorability of choosing j after i
- *Emphasizes **global** goodness*

The Probability Rule

$p(i, j)$ is chosen according to the **Probability Rule**:

$$p(i,j) = \frac{[\tau(i,j)].[\eta(i,j)]^{\beta}}{\sum_{g \in \text{allowed}} [\tau(i,g)].[\eta(i,g)]^{\beta}}$$

where:

- β is some constant (e.g. 2)
- $\Sigma_{g \in \text{allowed}}$ is normalized over all towns g still permitted in the tour

Pheromone Evaporation

After each iteration, the pheromone trail evaporates a small amount according to the following rule:

$$\tau[i, j] = \rho \cdot \tau[i, j] + \Delta\tau_{ij}$$

where $0 < \rho < 1$ is the **evaporation constant**

Pheromone Accrual

The amount of pheromone laid on edge (i, j) by the m ants at each time step is:

$$\Delta\tau_{ij} = \sum_{k=1}^{m} \Delta\tau_{ij}^{k}$$

where:

- $\Delta\tau_{ij}^{k} = Q/L_k$ is the pheromone density of k's tour
- Q is a constant
- L_k is the length of k's tour

TSP Ant System Algorithm

Initialize pheromone intensities on all edges to some small value α

Repeat

- Position m ants on initial cities
- Place each ant's starting city on the ant's tabu list
- Remember this city as the ant's starting city
- Create a tour for each ant
  - Each ant moves from town to town according to the Probability Rule while avoiding cities on its Tabu List

TSP Ant System Algorithm (2)

  - When a city is selected, it's added to the Tabu List
  - After n cities are selected and the tabu list is full, the starting city is selected as the final choice
- Compute $\Delta\tau_{ij}^{k}$ and L_k
- Save the shortest path found (and output it)
- Empty Tabu Lists
- Update pheromone strengths

Repeat until **stagnation** (all ants make the same tour) or until a maximum tour counter is reached

Tweaks and Extensions

The TSP Ant System algorithm works great for short tours (less than 30 cities). To improve performance further, try:

- Global Updating: only the best performing ant is allowed to lay down pheromone at each iteration
- Local Updating: $\Delta\tau_{ij}$ is a constant based upon other TSP approximation methods
- ε percent of the time allow an ant to choose a random allowable city next
- or whatever else you can imagine…

"I think everybody should study ants. They have an amazing four part philosophy. Never give up, look ahead, stay positive and do all you can."
- Jim Rohn

ACO Applications

Application Areas

- Bin Packing Problem
- Vehicle Routing
- Scheduling
- Assignment Problem
- Set Problem
- Protein Folding
- Network Routing
- Personnel Placement

Application Areas (2)

- Data Mining
- Image Processing
- Circuit Design
- Intelligent Testing Systems
- Distributed Information Retrieval
- Graph Coloring
- Sequential Ordering

Etc…