ANT COLONY OPTIMIZATION

Clay McLeod

April 27, 2015

University of Mississippi

- Ant Colony Optimization (ACO) is a swarm algorithm that attempts to emulate the behavior of ants searching for food.

LISTS

- Ant Colony Optimization (ACO) is a swarm algorithm that attempts to emulate the behavior of ants searching for food.
- First proposed by Marco Dorigo for his Master's thesis in 1992.

LISTS

- Ant Colony Optimization (ACO) is a swarm algorithm that attempts to emulate the behavior of ants searching for food.
- First proposed by Marco Dorigo for his Master's thesis in 1992.
- Useful in solving minimum optimization problems, especially those that closely emulate the biological system by which they are inspired.

INSPIRATION

- Ants communicate through an indirect, distributed communication called *stigmergy*.

INSPIRATION

- Ants communicate through an indirect, distributed communication called *stigmergy*.
- Process involves modifying their environment by placing pheromone as they travel towards food.

INSPIRATION

- Ants communicate through an indirect, distributed communication called *stigmergy*.
- Process involves modifying their environment by placing pheromone as they travel towards food.
- Ants can sense the presence and density of the pheromone on the ground.

INSPIRATION

- Ants communicate through an indirect, distributed communication called *stigmergy*.
- Process involves modifying their environment by placing pheromone as they travel towards food.
- Ants can sense the presence and density of the pheromone on the ground.
- Through a natural instinct, ants are attracted to pheromone, making them more likely to follow the path of previous ants.

ALGORITHM OVERVIEW

initialization;
while *not sufficiently sure of optimal solution* **do**
  for *m ants* **do**
    currentPosition = N_{NEST};
    while *currentPosition* := N_{FOOD} **do**
      Randomly travel to a connected node, paths with pheromone are more likely to be chosen.
      Update currentPosition to our current node.
    end
  end
  Update global pheromone map
end

Algorithm 1: Simple ACO algorithm

Undirected Graph, $G \in (V, E)$

- Vertices are the environment, including the nest, food source, and intermediate landmarks.
- Edges are the connected paths for the ants to walk between these vertices.

Undirected Graph, $G \in (V, E)$

- Vertices are the environment, including the nest, food source, and intermediate landmarks.
- Edges are the connected paths for the ants to walk between these vertices.

Artificial Ants

- Represented by mathematical constructs used to simulate an ant performing a walk between two vertices.

Undirected Graph, $G \in (V, E)$

- Vertices are the environment, including the nest, food source, and intermediate landmarks.
- Edges are the connected paths for the ants to walk between these vertices.

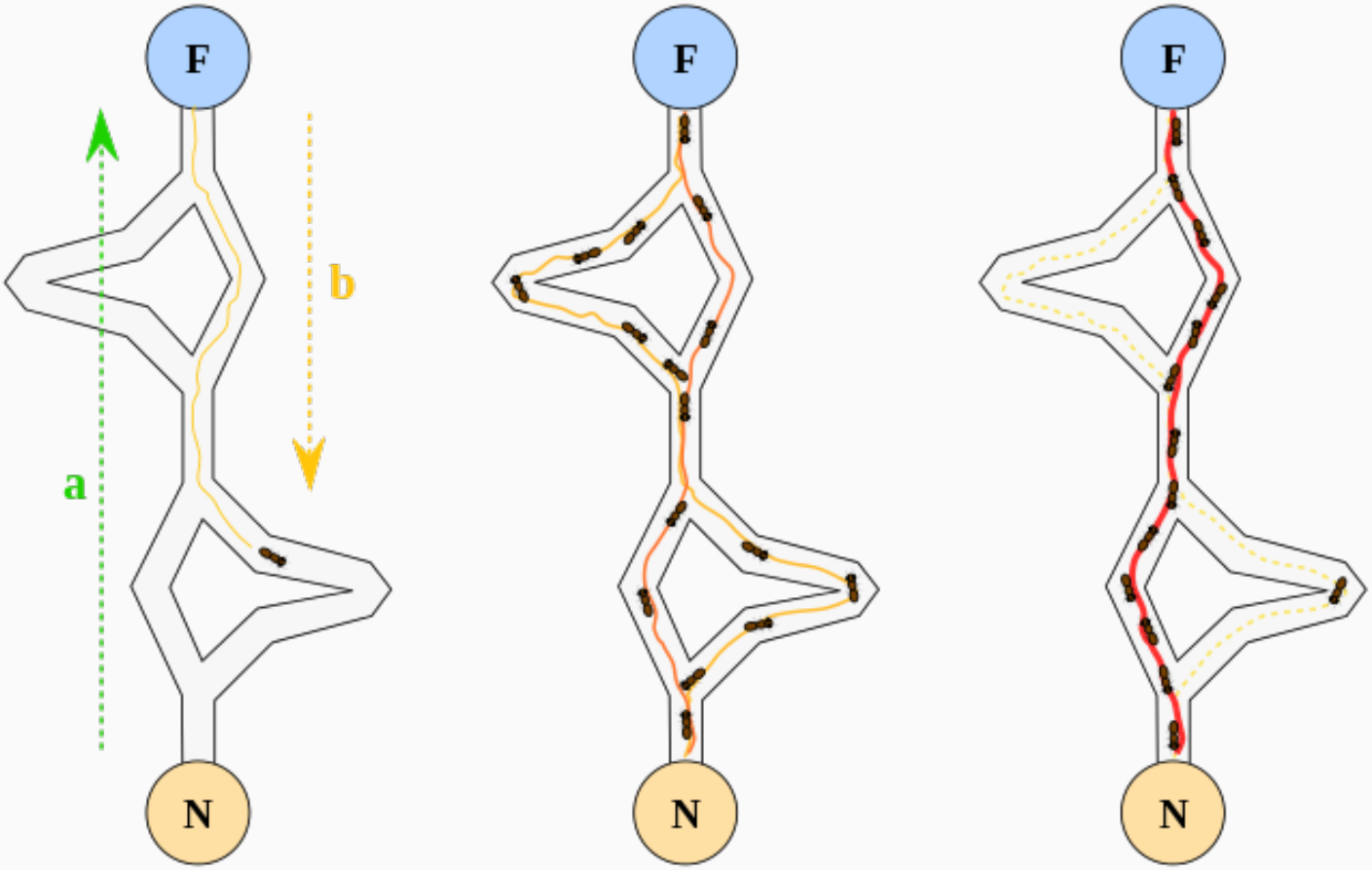
Artificial Ants

- Represented by mathematical constructs used to simulate an ant performing a walk between two vertices.

Pheromone

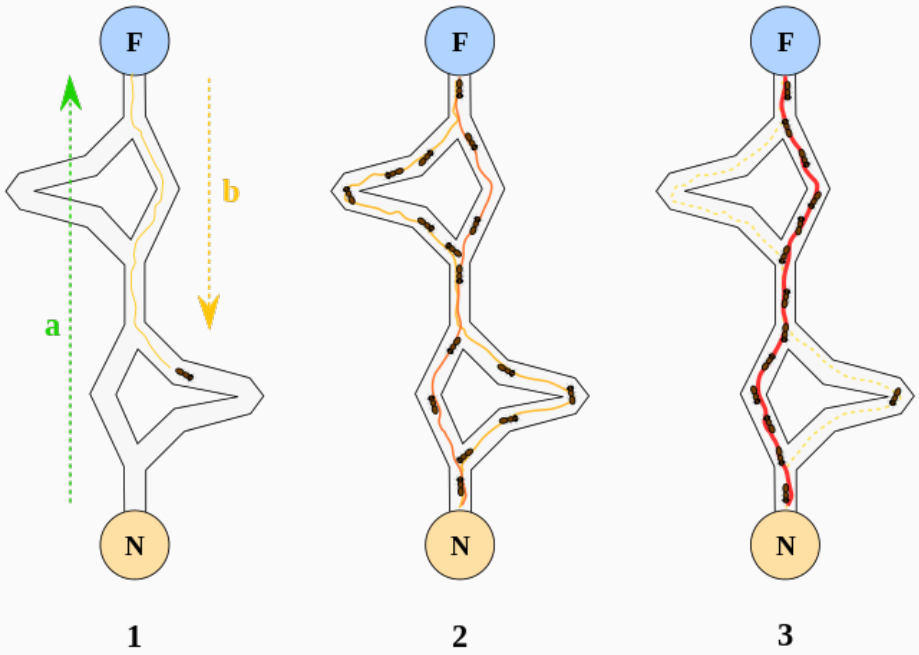
- Represented by normalized weights on each edge in the graph.

VISUALIZATION

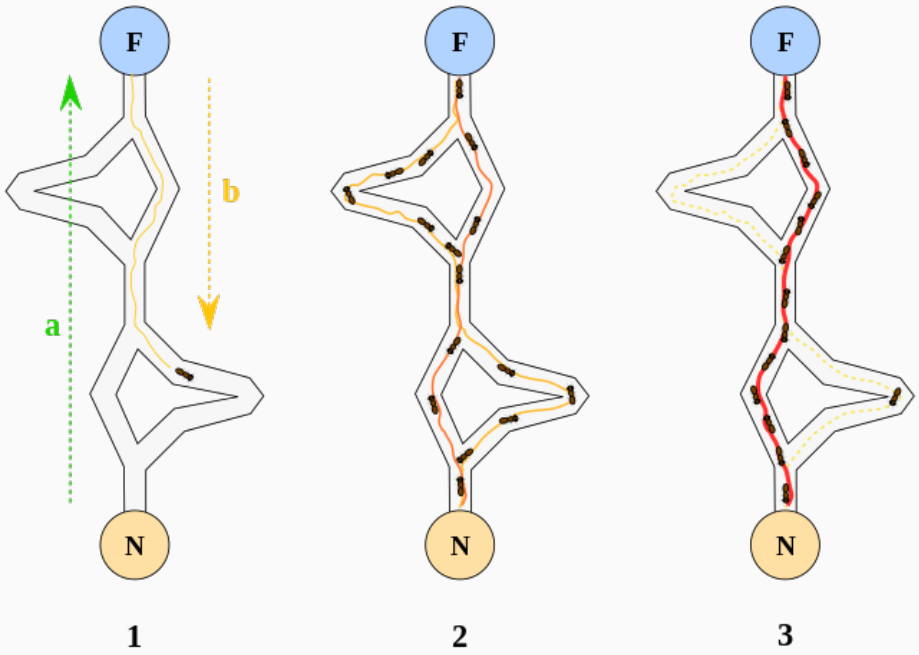


VISUALIZATION

Shorter paths are traveled more quickly by each ant, meaning that they become more saturated with pheromone as time goes on.



This, in turn, makes the ants more likely to follow the path because they are attracted to the pheromone, effectively converging on the optimal shortest path.



VARIATIONS

Ant System

- Pheromone is updated by all m ants after every ant has built a solution (like in the implementation described earlier).
- Baseline comparison, reasonable results.

Ant System

- Pheromone is updated by all m ants after every ant has built a solution (like in the implementation described earlier).
- Baseline comparison, reasonable results.

Ant Colony System

- Introduces ants also updating the pheromone levels after each step in their path construction.
- Mixed results based on application.

VARIATIONS

Ant System

- Pheromone is updated by all m ants after every ant has built a solution (like in the implementation described earlier).
- Baseline comparison, reasonable results.

Ant Colony System

- Introduces ants also updating the pheromone levels after each step in their path construction.
- Mixed results based on application.

MAX-MIN Ant System

- Only the best ant can place new pheromone levels in the graph, amount of minimum and maximum pheromone is bounded for each edge.
- Produces consistently better results than the first two approaches.

APPLICATIONS

- Optimal - Vehicle Routing, Bayesian Networks, Project Scheduling[1]

[1] Highly constrained results.

SELECTED APPLICATIONS

- Optimal - Vehicle Routing, Bayesian Networks, Project Scheduling[1]
- Good, not great - Traveling salesman, Max Clique Problem

[1] Highly constrained results.

SELECTED APPLICATIONS

- Optimal - Vehicle Routing, Bayesian Networks, Project Scheduling[1]
- Good, not great - Traveling salesman, Max Clique Problem
- Experimental - Protein Folding

[1] Highly constrained results.

Questions?