Introduction to Heuristic Algorithms 152117127

Week 11: Ant Colony Optimization

Asst. Prof. Dr. Emrah ATILGAN

Ant Colony Optimization

Another Swarm Optimization Algorithm

- Remember the swarm behavior
  - No central control
  - Only simple rules for each individual
  - Emergent phenomena
  - Self-Organization

Ant Colony Optimization

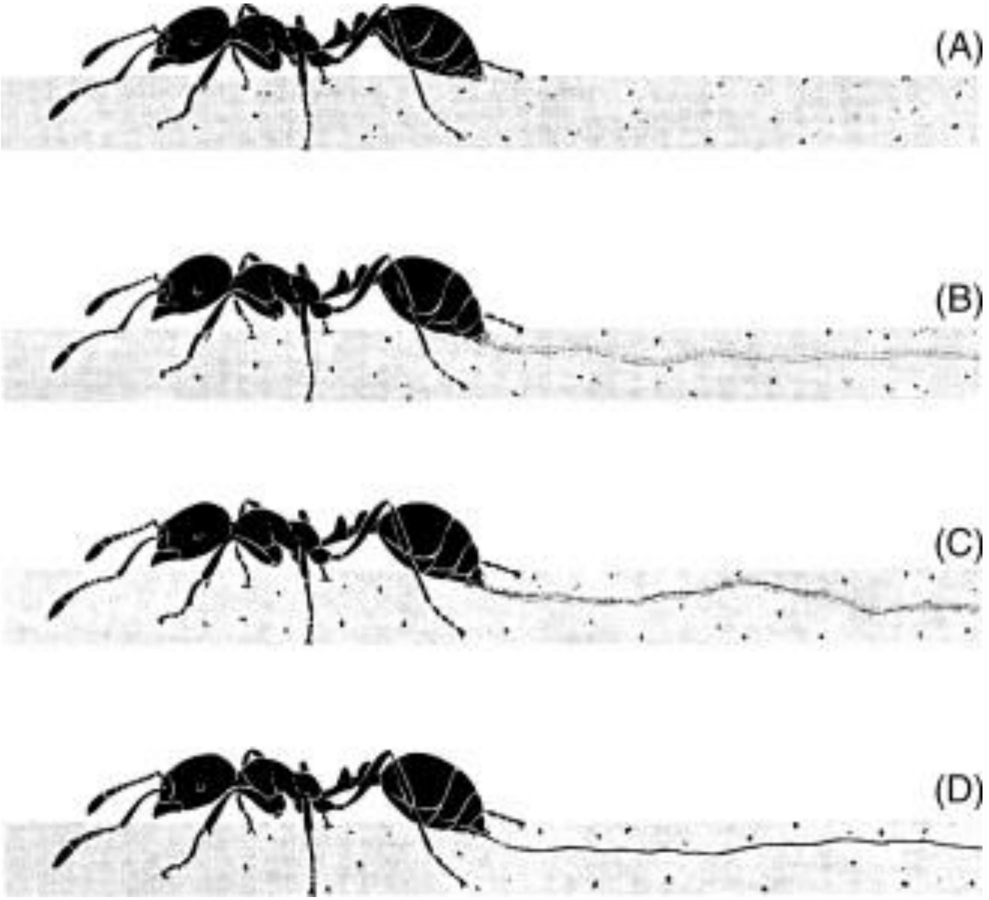
Ant Colony Optimization

- Inspired by foraging behavior of ants.
- Ants find shortest path to food source from nest.
- Ants deposit pheromone along traveled path which is used by other ants to follow the trail.
- This kind of indirect communication via the local environment is called stigmergy.
    - Two individuals interact indirectly when one of them modifies the environment and the other responds to the new environment later.
- Has adaptability, robustness and redundancy.

Ant Colony Optimization

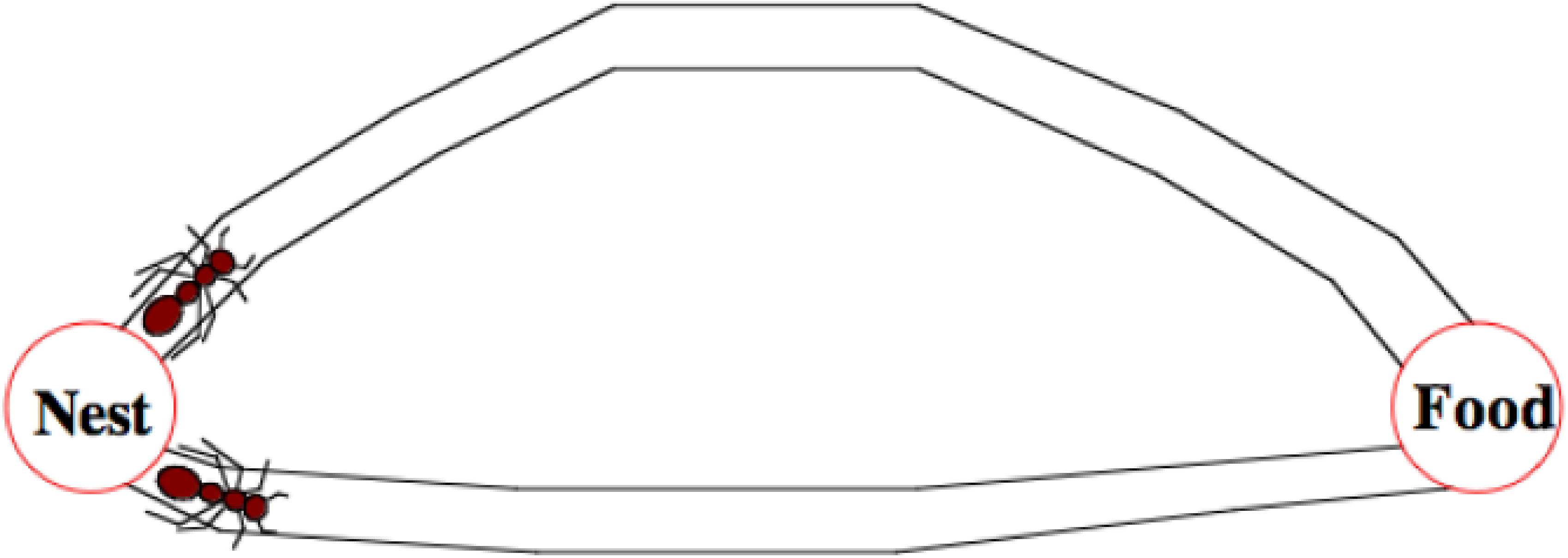
Pheromone

- A **pheromone** is a secreted or excreted chemical factor that triggers a social response in members of the same species.
- Pheromones are chemicals capable of acting like hormones outside the body of the secreting individual, to affect the behavior of the receiving individuals.
    - Aggregation
    - Alarm
    - Epideictic
    - Territorial
    - Trail
    - Sex
    - Other



Ant Colony Optimization

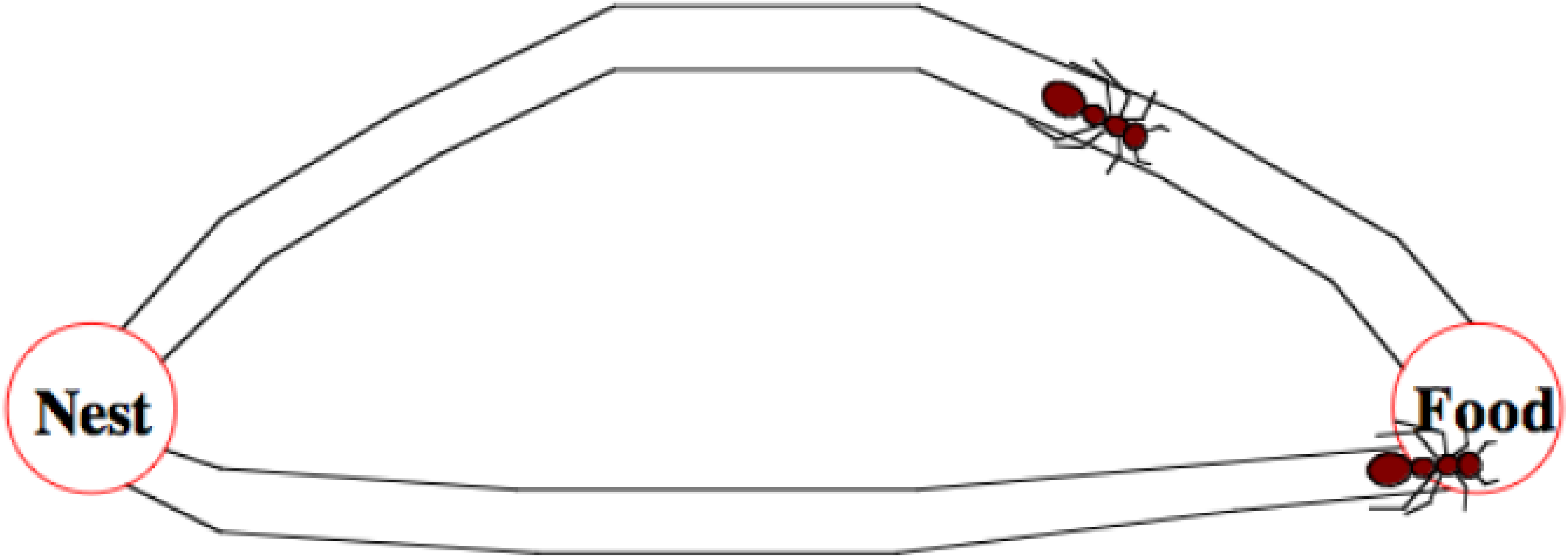
Foraging Behavior of Ants



- Two ants start with equal probability of going on either path.

Ant Colony Optimization

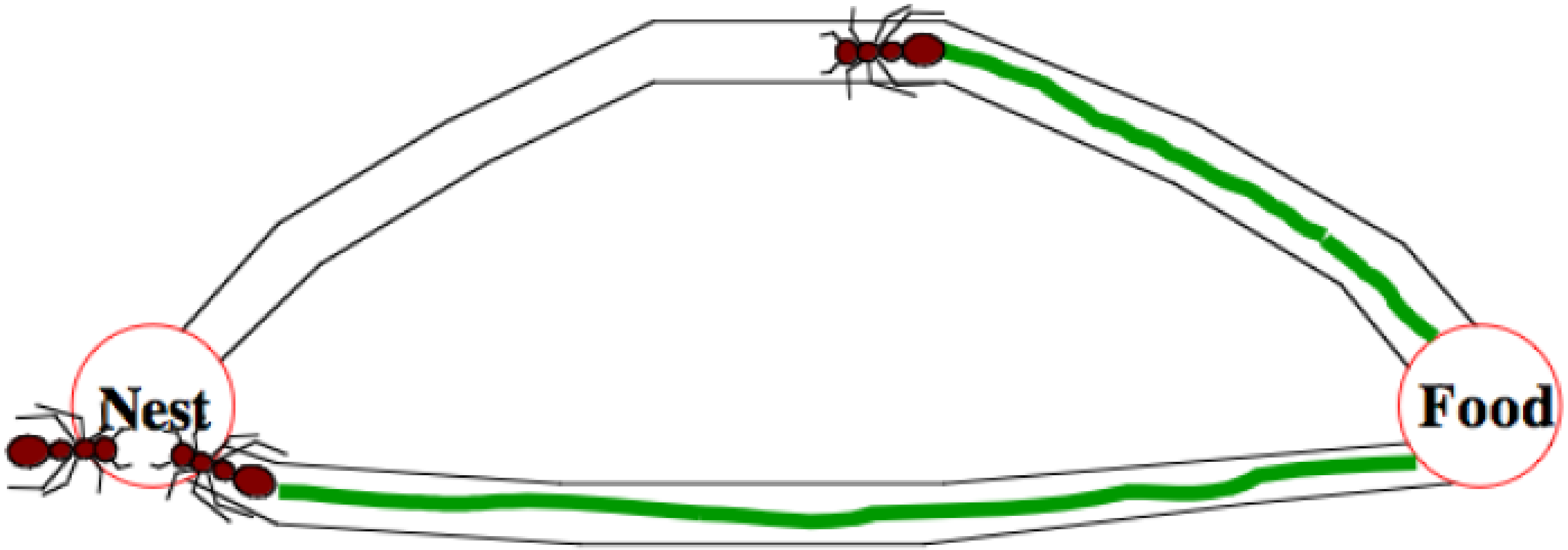
Foraging Behavior of Ants



- The ant on the shorter path reached the food earlier than the other.

Ant Colony Optimization

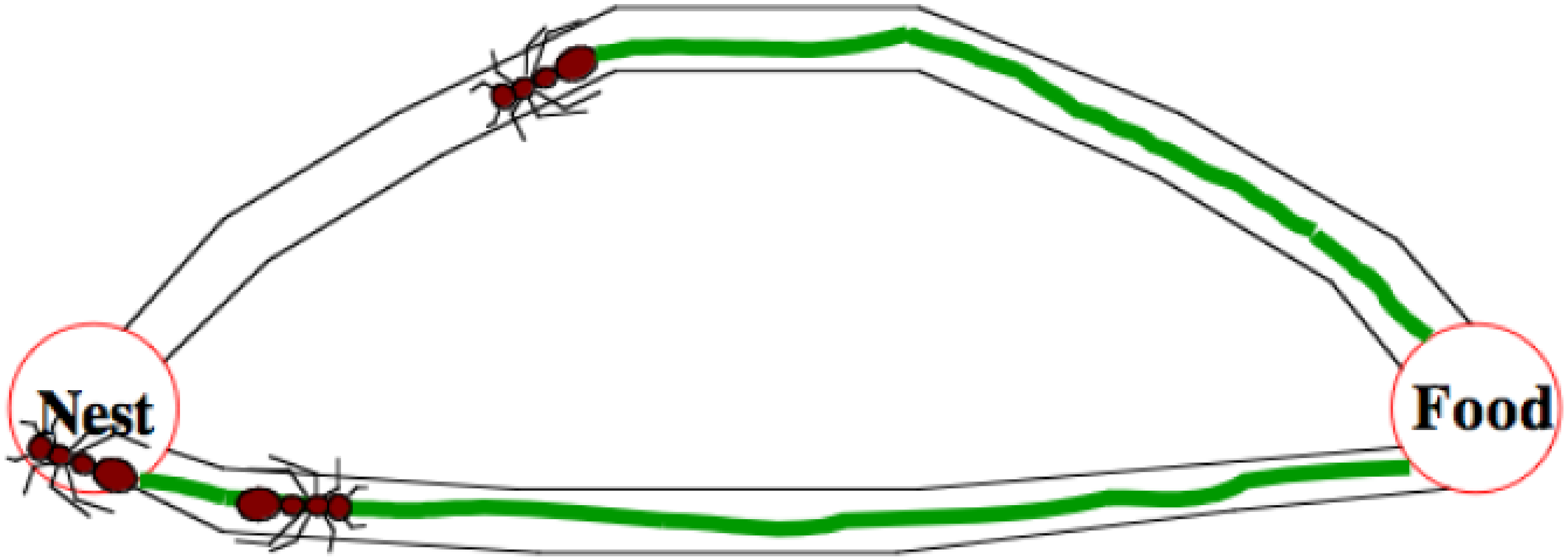
Foraging Behavior of Ants



- The density of pheromone on the shorter path is higher because of 2 passes by the ant (as compared to 1 by the other).

Ant Colony Optimization

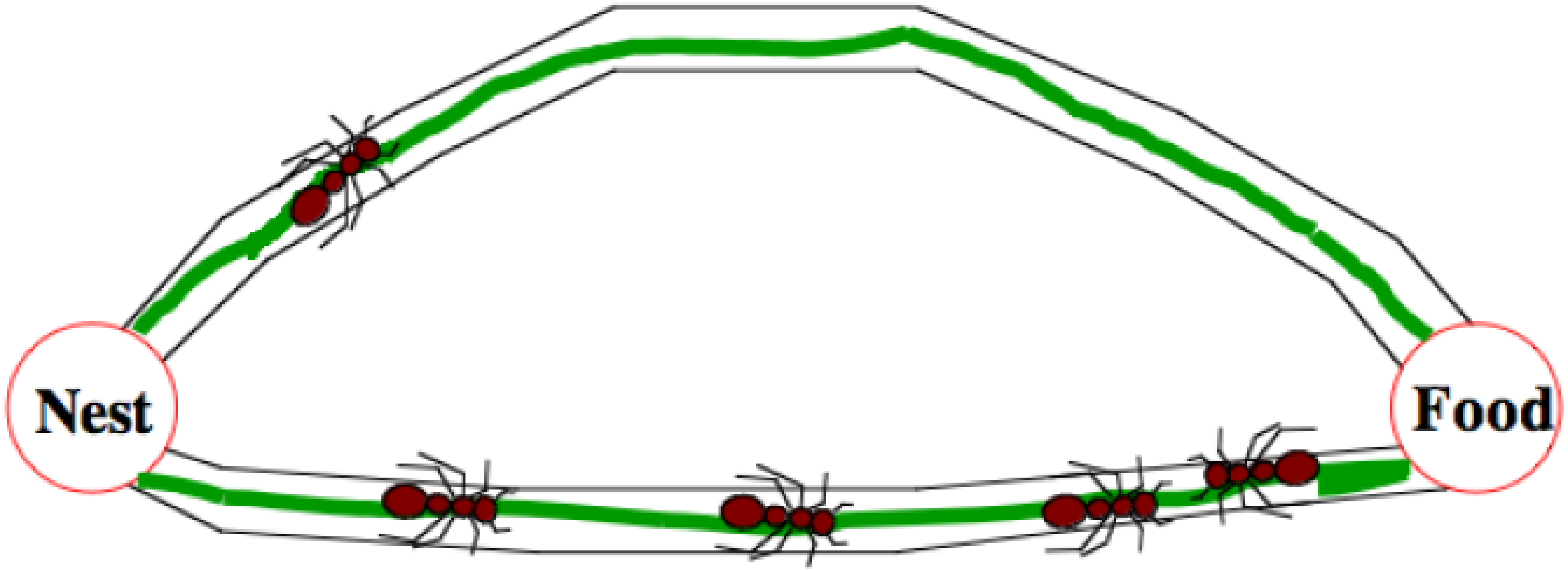
Foraging Behavior of Ants



- The next ant takes the shorter route.

Ant Colony Optimization

Foraging Behavior of Ants

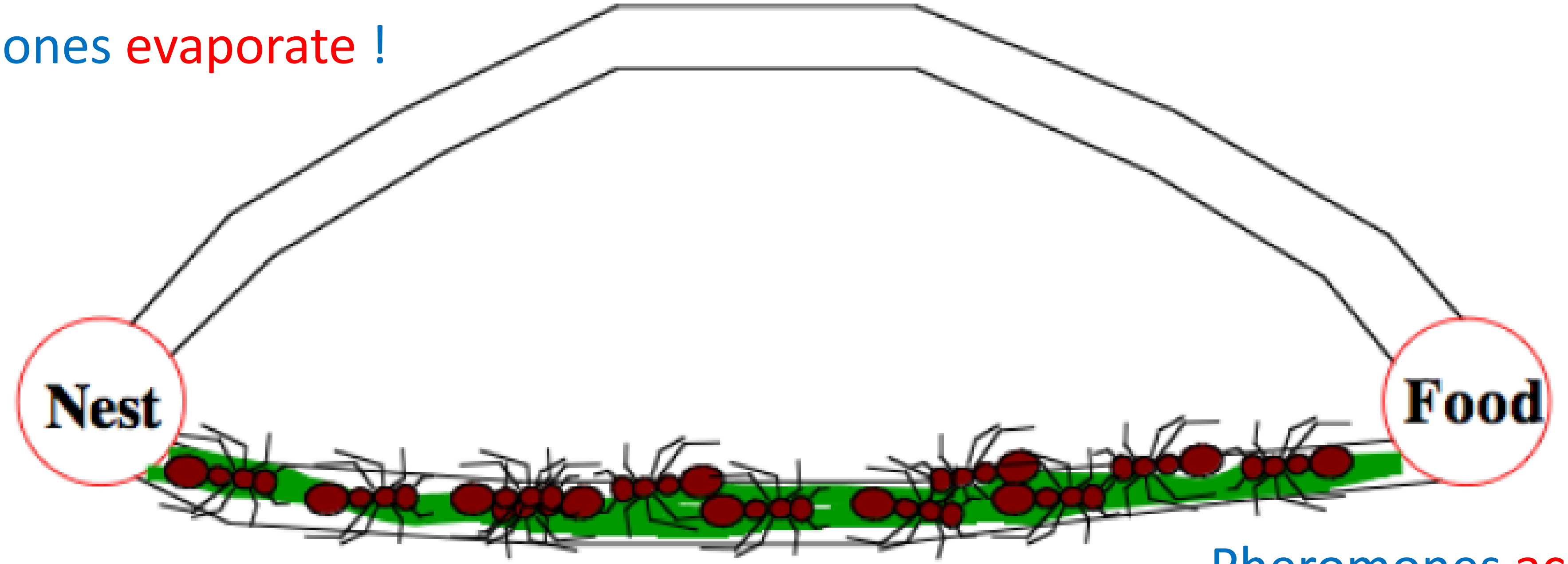


- As the iterations increases, more ants begin using the path with higher pheromone, thereby further reinforcing it.

Ant Colony Optimization

Foraging Behavior of Ants

Pheromones evaporate !

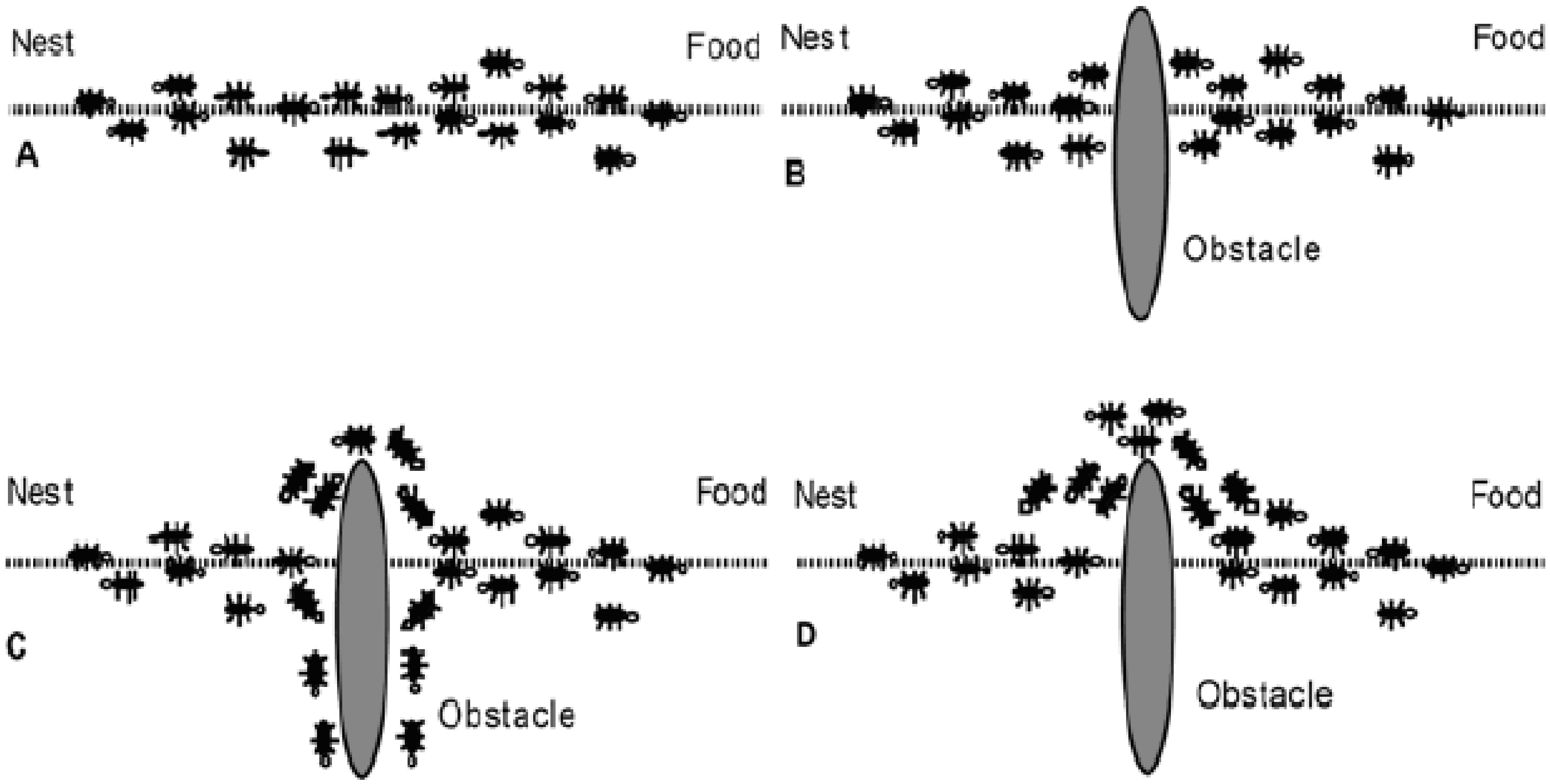


Pheromones accumulate with multiple ants using the path

- Eventually, the shorter path is almost exclusively used.

Ant Colony Optimization

Self-adapting behavior



Ant Colony Optimization

Ant Colony Optimization Algorithm

- The algorithm was proposed by Marco Dorigo in 1992.
  - M. Dorigo, "Optimization, learning and natural algorithms (*in italian*)," Ph.D. dissertation, Dipartimento di Elettronica, Politecnico di Milano, Italy, 1992.
- With contribution by other scientists:
  - M. Dorigo, V. Maniezzo, and A. Colorni, "Ant System: Optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics—Part B*, vol. 26, no. 1, pp. 29–41, 1996.
- Developed further in 1990s with more contribution.
- Nowadays, we have lots of variants of ACO.

Ant Colony Optimization

- Ant Colony Optimization (ACO) is a population based heuristic optimization method.

- It was originally designed for difficult combinatorial optimization problems.
  - Travelling Salesman
  - Vehicle Routing
  - Sequential ordering
  - Graph Coloring
  - Routing in communications network

Ant Colony Optimization

Main Idea

- Ant colony strategy formalized into a metaheuristic.
- Artificial ants build solutions to an optimization problem and exchange info on their quality.
- A combinatorial optimization problem reduced to a construction graph.
- Ants work concurrently and independently
- Ants build partial solutions in each iteration and deposit pheromone on each edge.
- This collective interaction via indirect communication leads to a good solution.

Ant Colony Optimization

ACO for Travelling Salesman Problem

1. Initializing the pheromone amounts on each route to a positive, small random value.

2. A simple transition rule for choosing the next city to visit, is

$$\Phi_{ij,k}(t) = \begin{cases} \dfrac{\tau_{ij}(t)^a}{\sum_{c \in C_{i,k}} \tau_{ic}(t)^a} & if\ j \in C_{i,k} \\ 0 & if\ j \in C_{i,k} \end{cases}$$

where $\tau_{ij}(t)$ is the pheromone intensity on edge (i,j) between cities i and j, the k_{th} ant is denoted by k, a is a constant, and $C_{i,k}$ is the set of cities and k still have to visit from city i.

Ant Colony Optimization

ACO for Travelling Salesman Problem

$$\Phi_{ij,k}(t) = \begin{cases} \dfrac{\tau_{ij}(t)^{a}}{\sum_{c \in C_{i,k}} \tau_{ic}(t)^{a}} & if\ j \in C_{i,k} \\ 0 & if\ j \in C_{i,k} \end{cases}$$

The transition rule above can be improved by including local information on the desirability of choosing city j when currently in city i, i.e.the next city to visit, is

$$\frac{\tau_{ij}(t)^{a}\eta_{ij}^{\beta}}{\sum_{c \in C_{i,k}} \tau_{ic}(t)^{a}\,\eta_{ic}^{\beta}}$$

where α and β are adjustable parameters that control the weight of pheromone intensity and

$$\eta_{ij} = \frac{1}{d_{ij}} \qquad is\ the\ attractiveness$$

Ant Colony Optimization

ACO for Travelling Salesman Problem

d_{ij} is the Euclidean distance between cities i and j

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

And the end of each route, τ_k, constructed by ant k, the pheromone intensity τ_{ij} on the edges of that route is updated, using

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

where,

$$\Delta\tau_{ij}(t) = \sum_{k=1}^{m} \Delta\tau_{ij,k}(t)$$

Ant Colony Optimization

ACO for Travelling Salesman Problem

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

- The parameter Q *has a value of the same order of the length of the optimal route,* $L_k(t)$ is the length of the route traveled by ant k*, and* m is the total number of ants.
- The constant $p \in [0,1]$, is referred to as the forgetting factor, which models the evaporation over time of pheromone deposits.

Pseudocode

ACO Algorithm

Input :pd, N

%%%% pd number of decision variables in ant, N iterations, Present position (ant) in the search universe X_{id} , ρ evaporation rate, %%%%%%

Output: Best_Solution

1: Initianlize_Node_Graph();

2: Initialize_Phermoni_Node();

3: While (num_of_Iterations>0) do

4: for each Ant

5: η_j ⇐ objective function of the search space

6: TRANSITION_RULE[j]= $p_j^m(t) = \frac{[\eta_j]\times[\tau_{ij}(t)]}{\sum_{i\in I_m}[\eta_i]\times[\tau_{ij}(t)]}$

7: Select node with the highest $p_j^m(t)$

8: Update Pheromone level $\tau_{ij}(t+1) = (1-\rho).\tau_{ij}(t) + \Delta\tau_{ij}(t)$

9: num_of_Iterations--;

10: end While

11:Best_sol ⇐ solution with best η_j

12: output(Best_sol)

Ant Colony Optimization

- Advantages:
  - Retains memory of entire colony instead of previous generation only.
  - Less affected by poor initial solutions (due to combination of random path selection and colony memory).
  - Has been applied to a wide variety of applications.
- Disadvantages:
  - Theoretical analysis is difficult:
    - Due to sequences of random decisions (not independent).
    - Probability distribution changes by iteration.
  - Convergence is guaranteed, but time to convergence uncertain.
  - Coding is somewhat complicated, not straightforward
  - Pheromone "trail" additions/deletions, global updates and local updates.

Ant Colony Optimization – Codes

- Python Codes
  - https://github.com/rochakgupta/aco-tsp
  - https://github.com/HaaLeo/swarmlib
- Matlab Codes
  - https://www.mathworks.com/matlabcentral/fileexchange/52859-ant-colony-optimization-aco
  - https://www.mathworks.com/matlabcentral/fileexchange/69028-ant-colony-optimiztion-aco?s_tid=srchtitle