# Ant Colony Optimization

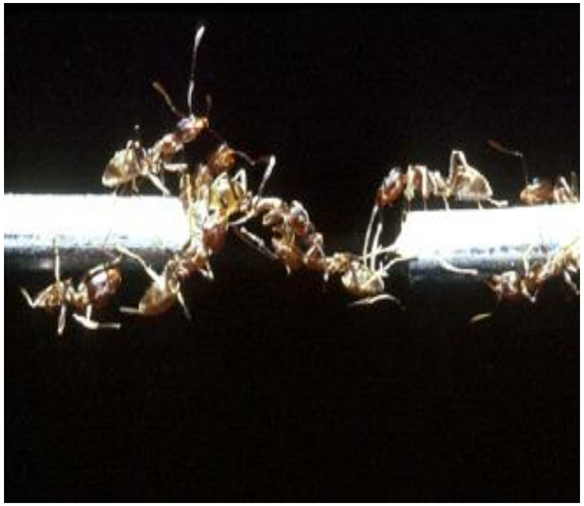
Winter 2011

# Motivation

- Inspired by the behavior of **ants** in finding paths from the colony to food
- Known for their highly organized **colonies**, which may consist of millions of individuals.
- Colony works as a unified entity

# Better Together

- Individually, ants are blind and may be stupid, but put enough of them together and they manage **cooperatively** (decentralized and self-organized) to build nests, grow fungus as food, **milk aphids**, and **weave their own shelters**


# Pheromone-driven Search

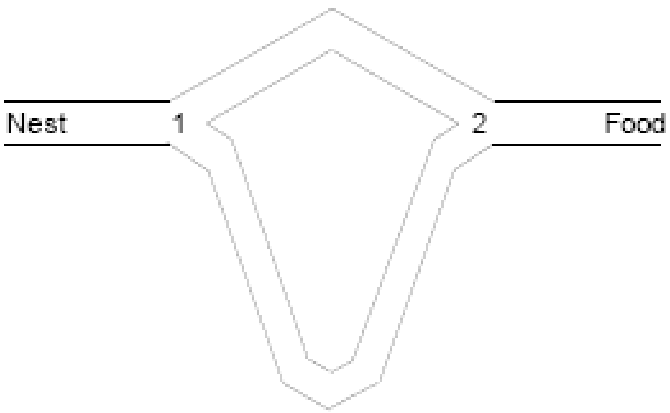
- When ants forage, they randomly wander the forest or jungle floor and lay a trail for nest-mates to lead them to a source of food.
- Many individual ants may discover different routes to the same food but the **shortest path** that leads to it will have the strongest concentration of **pheromone**, a chemical indicator laid down by the ants.

# Binary Bridge Experiment

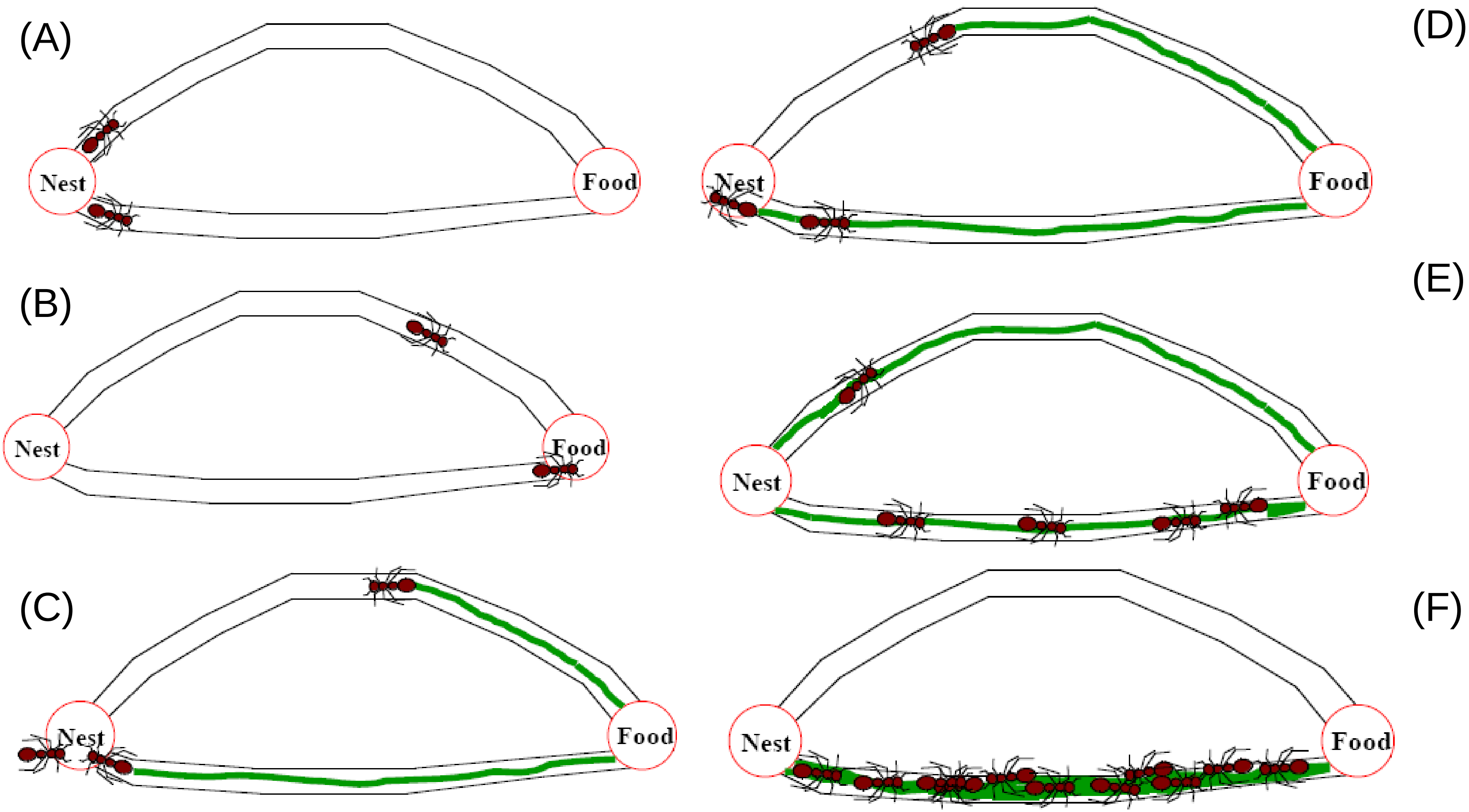
- After $t$ time units, $m_1$ ants had used the first bridge and $m_2$ the second one, the probability $p_1$ for the $(m_1+m_2+1)th$ ant to choose the first bridge can be given by

$$p_1 = \frac{(m_1 + k)^h}{(m_1 + k)^h + (m_2 + k)^h}$$

$$k \cong 20, h \cong 2$$



# Search for the Shortest



# Ant Colony Optimization (ACO)

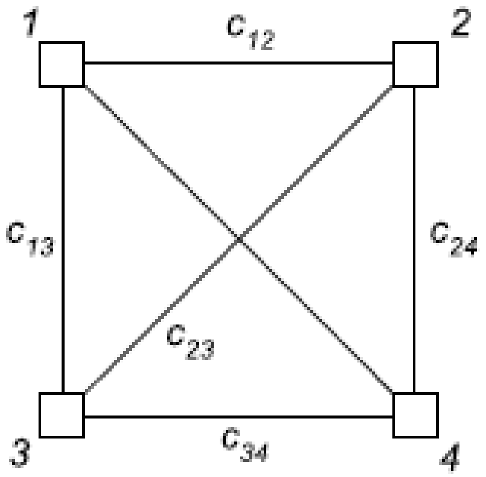
- References
  - Dorigo, M. and Stützle, T. (2004). *Ant Colony Optimization*, MIT Press
  - Dorigo. M., Birattari, M., Stützle, T. (2006). Ant Colony Optimization-- Artificial Ants as a Computational Intelligence Technique, *IEEE Computational Intelligence Magazine*
  - http://iridia.ulb.ac.be/~mdorigo/ACO/

# Applications of Ant Colony

- Primarily combinatorial optimization problems (with discrete and finite solution spaces)
  - TSP
  - Assignment and Facility layout
  - Scheduling of limited resources
  - Cutting
  - Transportation and vehicle routing (deliver goods from central depots to customers)
  - Construction Engineering: design of steel frames and water distribution systems

# Graphical Representation

- A number of artificial ants are simulated to move on a graph that encodes as left with vertices (nodes) and edges (links)
- The goal is to find the shortest path
- A variable called **pheromone** is associated with each edge and can be read and modified by ants.



# Idea (1)

- At each iteration, each of the ants builds a solution by walking from vertex to vertex on the graph with the constraint of not visiting any vertex previously visited
- At each step of solution construction, an ant selects the next vertex according to a stochastic mechanism that is biased by the **pheromone**: an unvisited vertex $j$ can be selected with a probability that is proportional to the **pheromone** associated with edge $(i, j)$.

# Idea (2)

- At the end of an iteration, on the basis of the quality of the solutions constructed by the ants, the **pheromone** values are modified (**increase intensity while evaporating**) in order to bias ants in future iterations to construct solutions similar to the best ones previously constructed
- Pheromone evaporation is to **avoid the convergence to a locally optimal solution**

# Sketch (1)

1. Initial population
   – Create a population of ants; distribute them across the vertices when the start vertex is unknown
2. Ant movement
   – Use state transition probability (will come back to this later) to determine next **unvisited vertex** to reach

$$p_j = \frac{\tau(i, j)^{\alpha} \eta(i, j)^{\beta}}{\sum_{h \in J} \tau(i, h)^{\alpha} \eta(i, h)^{\beta}}$$

# Sketch (2)

$$p_j = \frac{\tau(i,j)^{\alpha}\,\eta(i,j)^{\beta}}{\sum_{h \in \mathrm{J}} \tau(i,h)^{\alpha}\,\eta(i,h)^{\beta}}$$

$p_j$ : probability of visisitng node $j$

$\tau(i,j)$ : level of pheromone on the edge $(i,j)$

$\eta(i,j)$ : heuristic function (e.g., inverse of the distance) on the edge $(i,j)$; also called "visibility"

$h$ is an unvisisted node, belonging to the succedding set J

$\alpha$ and $\beta$ are weighting parameters

# Sketch (3)

3. Ant tour
   – An ant completes its tour when it visits all the vertices (**no cycle is permitted**; thus need to maintain a list of unvisited vertices)
   – Evaluate the length of the entire tour based on a list of vertices in the current tour

# Sketch (4)

4. Pheromone intensification and evaporation

$$\tau(i, j) = (1-\rho)\cdot\tau(i, j) + \sum_{k=1}^{m}\Delta\tau_k(i, j)$$

$\rho$ : evapoeration rate

$\Delta\tau_k(i, j)$ : quantity of pheromone on edge $(i, j)$ laid by ant $k$

$$\Delta\tau_k(i, j) = \begin{Bmatrix} Q/L_k & \text{if ant } k \text{ used edge } (i, j) \text{ in its tour} \\ 0 & \text{otherwise} \end{Bmatrix}$$

$Q$ : constant; $L_k$ : tour length

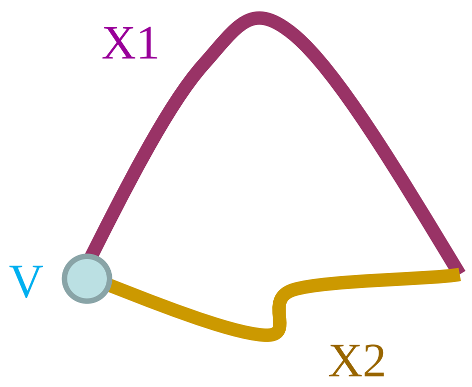
# Sketch (5)

5. Start next iteration, pass on pheromone information
6. Clear memory of ants (create a new list of unvisited vertices)
7. Place ants on vertices
8. Ants work on new tours according to the updated pheromone and the visibility

# Sample Iteration (1)

- At iteration t, two ants Z1 and Z2 started at vertex V; each takes a different path: X1 and X2
- The pheromone level on X1 and X2 is currently 0.4 and 0.6
- After reaching the end, Z1 traveled 20 steps whereas Z2 traveled 10 steps
- Other parameters are

  $Q = 10; \alpha = 1; \beta = 2; \rho = 0.3$



# Sample Iteration (2)

- After this iteration, pheromone becomes

For X1, $\tau_1$ = (1-0.3)×0.4 + 10/20 = 0.78

For X2, $\tau_2$ = (1-0.3)×0.6+ 10/10 = 1.42

- Restart ants at vertex V

$$p_1 = \frac{\tau(i,j)^\alpha \eta(i,j)^\beta}{\sum_{h \in J} \tau(i,h)^\alpha \eta(i,h)^\beta} = \frac{0.78^1 \times (1/20)^2}{0.78^1 \times (1/20)^2 + 1.42^1 \times (1/10)^2} = 0.121$$

$$p_2 = \frac{1.42^1 \times (1/10)^2}{0.78^1 \times (1/20)^2 + 1.42^1 \times (1/10)^2} = 0.879$$

- Ants would pick X2 with higher probability

# Sample Iteration (3)

- Suppose both ants pick X2

For X1, $\tau_1 = (1\text{-}0.3)\times 0.78 + 10/20 = 1.046$

For X2, $\tau_2 = (1\text{-}0.3)\times 1.42 + 10/10 = 1.994$

- Restart ants at vertex V

$$p_1 = \frac{\tau(i,j)^{\alpha}\eta(i,j)^{\beta}}{\sum_{h\in J}\tau(i,h)^{\alpha}\eta(i,h)^{\beta}} = \frac{1.046^1 \times (1/20)^2}{1.046^1 \times (1/20)^2 + 1.994^1 \times (1/10)^2} = 0.116$$

$$p_2 = \frac{1.994^1 \times (1/10)^2}{1.046^1 \times (1/20)^2 + 1.994^1 \times (1/10)^2} = 0.884$$

- Ants pick X2 with a slightly higher probability

# Suggested Parameters

- Weighting parameters
  - Usually $\beta \geqq \alpha$; visibility is a greater determinant than pheromone
- Evaporation rate
  - Avoid locally optimal solutions
  - Usually $\rho \geqq 0.4$
- Number of ants
  - Problem specific; may increase as progress

# TSP Review

- Find, for a given set of $n$ cities with distance $d_{ij}$ between each pairs of cities, a shortest tour that contains every city exactly once
- Distance $d_{ij}$ is recorded in a square matrix (symmetrical or unsymmetrical); the diagonal elements are set to be large to prohibit cycling

# ACO applied to TSP

- Pheromone information is stored in a square matrix; each element denotes the quantity of pheromone on an edge ($i$,$j$)
- A city list is kept for each ant at the start of each iteration. When a city is chosen, it will be removed from the list

# Pseudocode for ACO in TSP

```
Initialize pheromone values
Repeat
    For each ant k
        S = {1, 2, 3,…n}
        Choose city i randomly
        S = S \ {i}
        Repeat
            Choose city j with p_ij
            S = S \ {j}
            i = j
        Until S = Ø
    End For
    For all (i,j)
        Pheromone Intensification
            and Evaporation
    End For
Until stopping criterion is met
```

# Other Applications

- In addition to TSP, ACO can be applied to many other combinatorial problems
- Key is to alter the heuristic function to reflect the solution quality, i.e., objective value

$$p_j = \frac{\tau(i,j)^{\alpha}\boxed{\eta(i,j)^{\beta}}}{\sum_{h\in J}\tau(i,h)^{\alpha}\boxed{\eta(i,h)^{\beta}}}$$

- Usually, the heuristic function is to treat minimization problem: taking an inverse if necessary

# Variants

- In the following, we discuss some variants of ACO
  1. Add Local Search
  2. Ant Colony System
  3. Population-based ACO

# Variant: Add Local Search

- Some suggest to perform local search **after** each ant constructs a solution
- For example, if the solution is {1 6 3 4 2 5}; swap two cities until no improvement can be made
- Because local search may be time-consuming, allow only the few best ants to perform local search

# Variant: Ant Colony System (1)

- Each ant builds a tour by repeatedly applying a stochastic greedy rule (**state transition rule**)
- **While** constructing its tour, an ant also modifies the amount of pheromone on the visited edges by applying the local updating rule (Originally, the pheromone is updated **after** all ants complete tour)
- Once all ants have completed their tour, the amount of pheromone on edges is modified again (**global update**)

# Variant: Ant Colony System (2)

- State transition rule from *i* to *j*

$$j = \begin{cases} \arg\max_{u \in S} \{[\tau(i,u)^{\alpha} \cdot \eta(i,u)^{\beta}\} \ \text{if } q \le q_0 & \text{(exploition)} \\ \text{Regular ACO rule} & \text{(biased exploration)} \end{cases}$$

Greedy rule: pick the best with pre-specified probability $q_0$

# Variant: Ant Colony System (3)

- Ants change pheromone level of an edge **during** their visit

$$\tau(i, j) = (1 - \rho) \cdot \tau(i, j) + \rho \cdot \Delta\tau(i, j)$$

$0 \leq \rho \leq 1$; $\rho$ is a tuning parameter

$\Delta\tau(i, j) =$ initial pheromone level

# Variant: Ant Colony System (4)

- Only **globally best ant** (i.e., the ant which constructed the shortest tour in all the past iterations) is allowed to deposit pheromone

$$\tau(i, j) = (1-\gamma)\cdot\tau(i, j) + \gamma\cdot\Delta\tau(i, j)$$

$$0 \le \gamma \le 1; \gamma \text{ is a tuning parameter}$$

$$\Delta\tau(i, j) = \begin{cases} {L_{global}}^{-1} & \text{if } (i, j) \in \text{global best tour} \\ 0 & \text{otherwise} \end{cases}$$

- This is done **after** all ants complete tours

# Variant: Population-based ACO (1)

- P-ACO maintains a small population P of the *k* best solutions in past iterations; "elite archive"
- P is updated after every iteration; pick the best to enter P
- P's size does not exceed *k*: remove the oldest solution leaves when a new one enters
- The pheromone matrix is derived anew in every iteration from population P

# Variant: Population-based ACO (2)

- Each pheromone value is increased by

$$\tau(i, j) = \tau(i, j) + \zeta_{ij} \cdot \Delta$$

$$\zeta_{ij} = \text{number of solutions in P with } (i, j)\text{; always integer}$$

$$\Delta = \text{user} - \text{specified constant}$$

- The update of pheromone values can be done as a population update: increasing as (*i*,*j*) enters P, decreasing as (*i*,*j*) leaves P
- No evaporation

# Advanced Topics (1)

- Dynamic optimization problems
  - Search space changes during time
    - Distances may change; cities are added or removed
    - New customers join the request list as vehicles are out
    - Telecommunication networks
  - Key: Optimization time is highly restricted
    - Advantage of using artificial ants

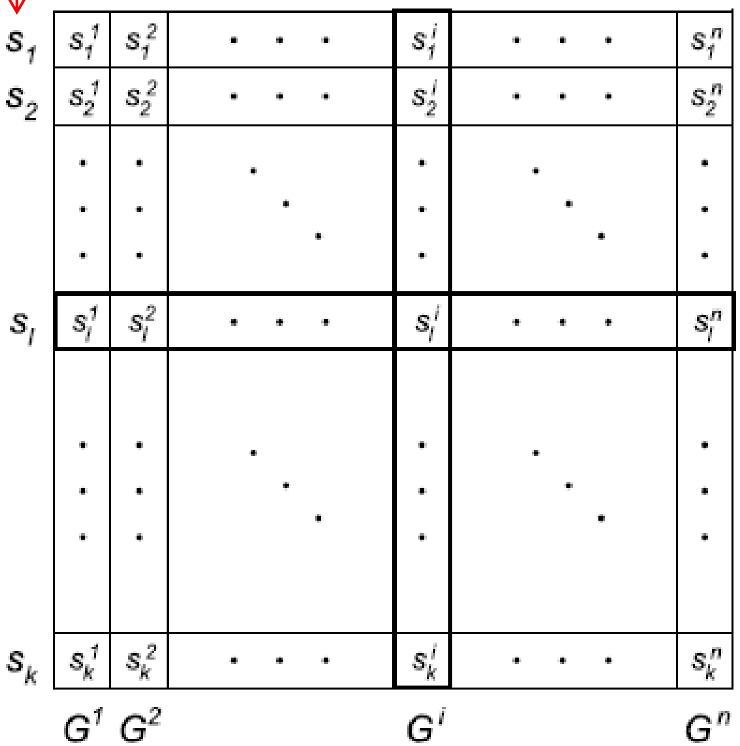
# Advanced Topics (2)

- Continuous optimization problems
  - Divide the domain of each variable into a finite set of intervals; then use regular ACO to solve it
  - Ants sample next solutions from a **continuous** probability density function (PDF), not from a **discrete** list

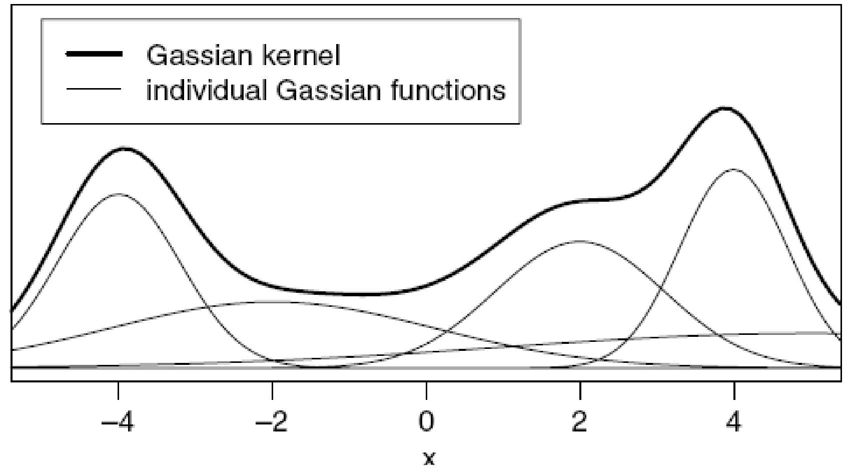
# Advanced Topics (2)

solutions

Pheromone Table




variables




$$G^i(x) = \sum_{l=1}^{k} \omega_l g_l^i(x) = \sum_{l=1}^{k} \omega_l \frac{1}{\sigma_l^i \sqrt{2\pi}} e^{-\frac{(x-\mu_l^i)^2}{2\sigma_l^{i\,2}}}$$

weights

Socha, K. and Dorigo, M. (2008). “Ant colony optimization for continuous domains.” European Journal of Operational Research, 185, pp. 1155-1173.

# Advanced Topics (3)

- Parallel implementation
  - Multi-colony ACO: Large subpopulations are assigned to single processors and information exchange is relatively rare
  - Master-slave ACO: Central processor collects solutions, updates the pheromone matrix, and send back to other processors for future iteration

# Advanced Topics (4)

- Multi-objective optimization problems
  - Multiple pheromone matrices
  - Ants are attracted to the matrices with various weights
  - Using an elite archive to intensify pheromone on good solutions

# Conclusions

- ACO has been used to solve combinatorial optimization problems with success
- Similar to other meta-heuristics, ACO is currently evolving
- Consider more ill-structured / dynamic / stochastic problems