MATHEMATICAL PROGRAMMING

SWARM INTELLIGENCE -1

(ANT-COLONY OPTIMIZATION)

CO - 4

Session - 22

Dr Lakshmi Ramani B
Associate Professor

AIM`

1. To familiarise students to the fundamental principles of Swarm Intelligence and the Ant Colony Optimization algorithm.

INSTRUCTIONAL OBJECTIVES

This session is designed to:

1. Provide Swarm Intelligence including
   a. Ant-colony optimization

LEARNING OUTCOMES

At the end of this session, students will be able to know and apply:

1. Swarm Intelligence concepts
2. Ant-colony optimization.

WHAT IS A SWARM?

- A loosely structured collection of interacting agents.
  - **Agents:**
    - Individuals that belong to a group (but are not necessarily identical).
    - They contribute to and benefit from the group.
    - They can recognize, communicate, and/or interact with each other.
- The natural perception of swarms is a group of agents in motion – but that does not always have to be the case.
- A swarm is better understood if thought of as agents exhibiting a collective behavior.

SWARM INTELLIGENCE (SI)

- An artificial intelligence (AI) technique based on the collective behavior in decentralized, self-organized systems.
  - Generally made up of agents who interact with each other and the environment.
  - No centralized control structures.
  - Based on group behavior found in nature.
- "The emergent collective intelligence of groups of simple agents." (Bonabeau et al, 1999)

EXAMPLES OF SWARMS IN NATURE:

- Classic Example: **Swarm of Bees**.
- Can be extended to other **similar systems**:
  - **Ant colony**
    - **Agents: ants**
  - **Flock of birds**
    - **Agents: birds**
  - **Traffic**
    - **Agents: cars**
  - **Crowd**
    - **Agents: humans**
  - **Immune system**
    - **Agents: cells and molecules**

CHARACTERISTICS OF SWARMS

- Composed of many individuals
- Individuals are homogeneous
- Local interaction based on simple rules
- Self-organization (No centralized Control)

SWARM INTELLIGENCE (SI) - ALGORITHM

- Inspiration from swarm intelligence has led to some highly successful optimisation algorithm.

  - **Ant Colony (-based) Optimisation** – a way to solve optimisation problems based on the way that ants indirectly communicate directions to each other.

ANT COLONY OPTIMIZATION (ACO)

- The study of artificial systems modeled after the behavior of real ant colonies and are useful in solving discrete optimization problems.
- Introduced in 1992 by Marco Dorigo.
  - Originally called it the Ant System (AS).
  - Has been applied to
    - Traveling Salesman Problem (and other shortest path problems).
    - Several NP-hard Problems.
- It is a population-based metaheuristic used to find approximate solutions to difficult optimization problems.

ACO CONCEPT

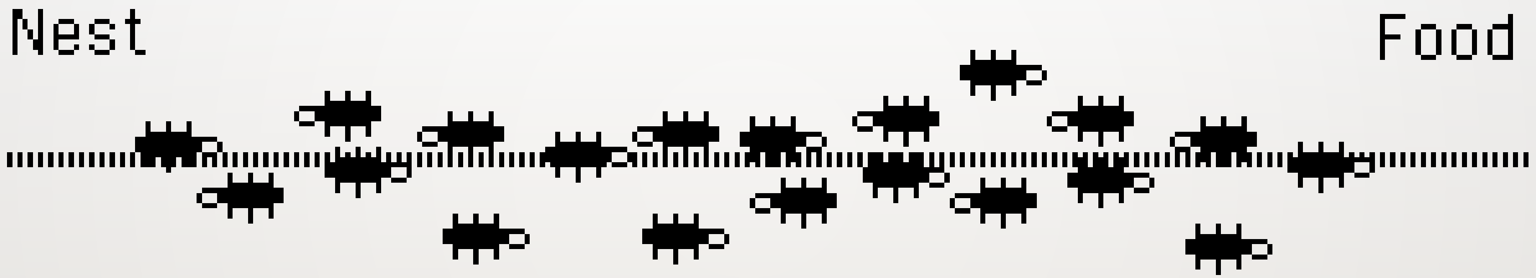
- Ant Colony Optimization (ACO) studies artificial systems that take inspiration from the *behavior of real ant colonies* and which are used to solve discrete optimization problems."
  - Ants navigate from nest to food source. Ants are blind!
  - Shortest path is discovered via pheromone trails. Each ant moves at random
  - Pheromone is deposited on path
  - More pheromone on path increases probability of path being followed

A KEY CONCEPT: STIGMERGY

- **Stigmergy** is: **indirect communication via interaction with the environment.**

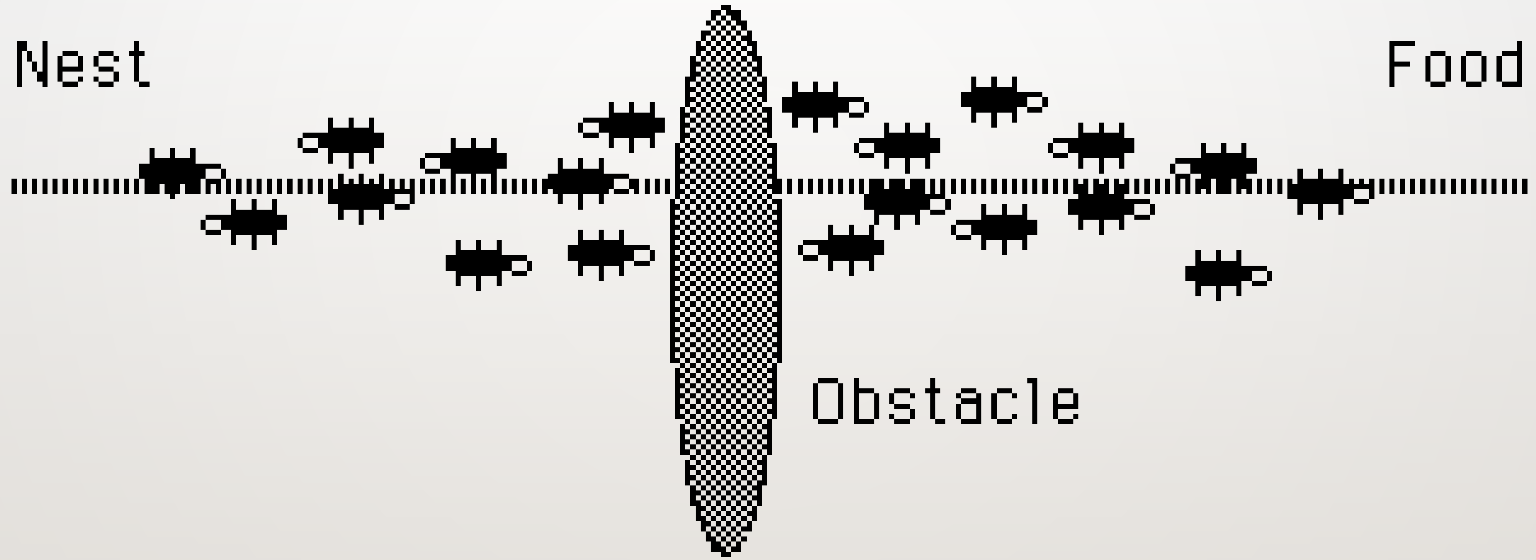
  - A problem gets solved bit by bit ..
  - Individuals communicate with each other in the above way, affecting what each other does on the task.
  - Individuals leave *markers* or *messages* – these don't solve the problem in themselves, but they affect other individuals in a way that helps them solve the problem …

NATURALLY OBSERVED ANT BEHAVIOR



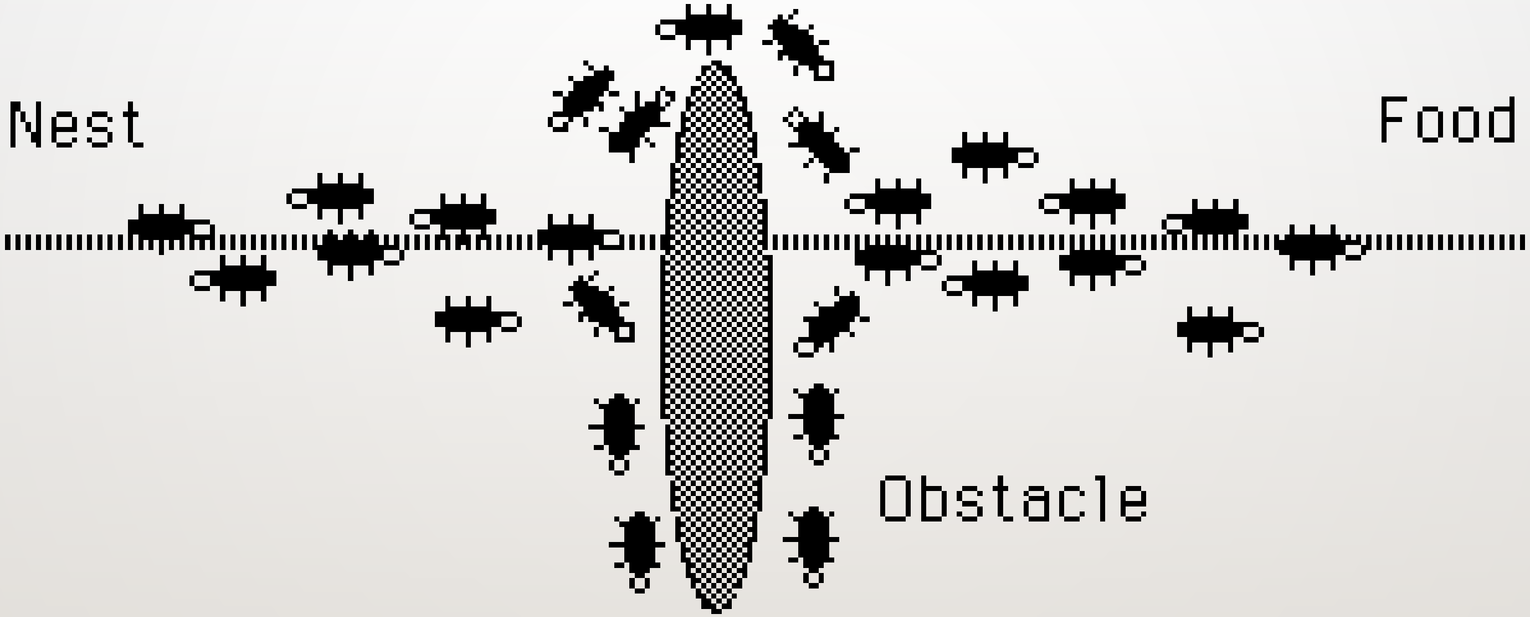
All is well in the world of the ant.

NATURALLY OBSERVED ANT BEHAVIOR



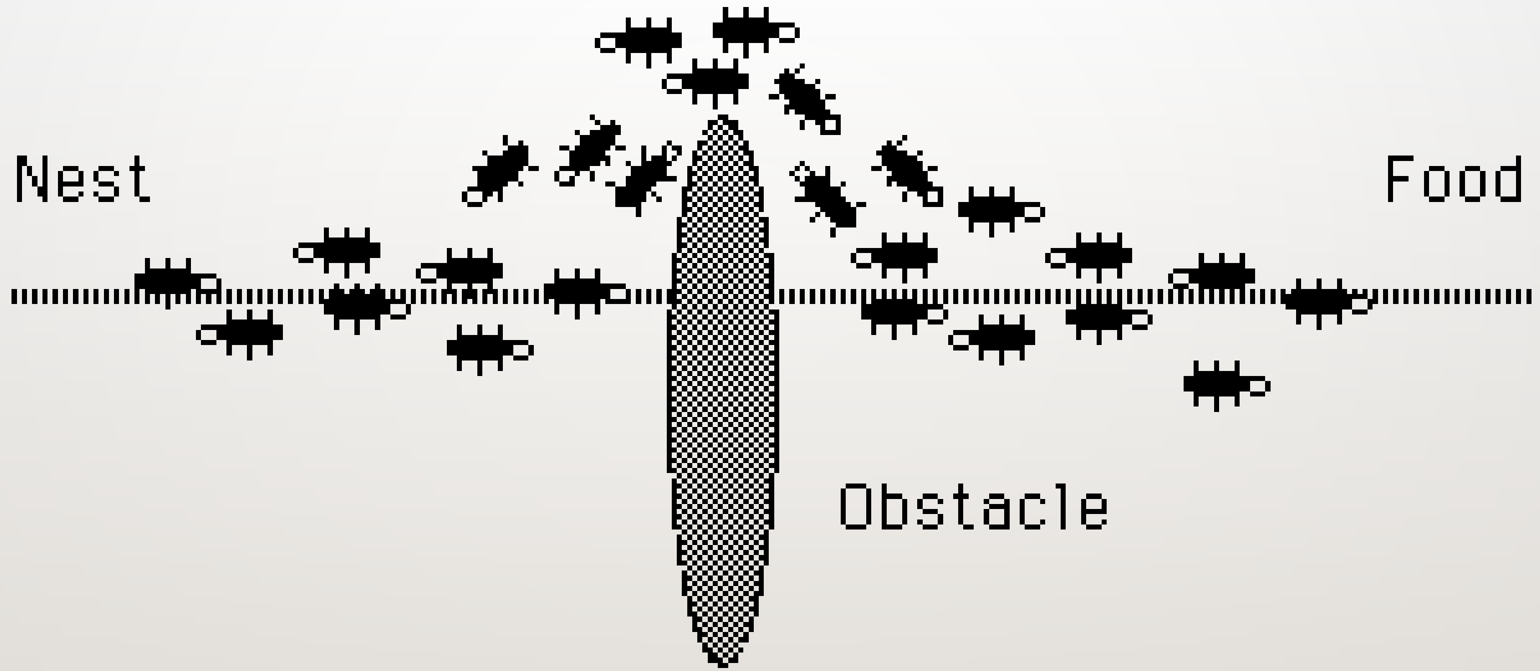
Oh no! An obstacle has blocked our path!

NATURALLY OBSERVED ANT BEHAVIOR



Where do we go? Everybody, flip a coin.

NATURALLY OBSERVED ANT BEHAVIOR



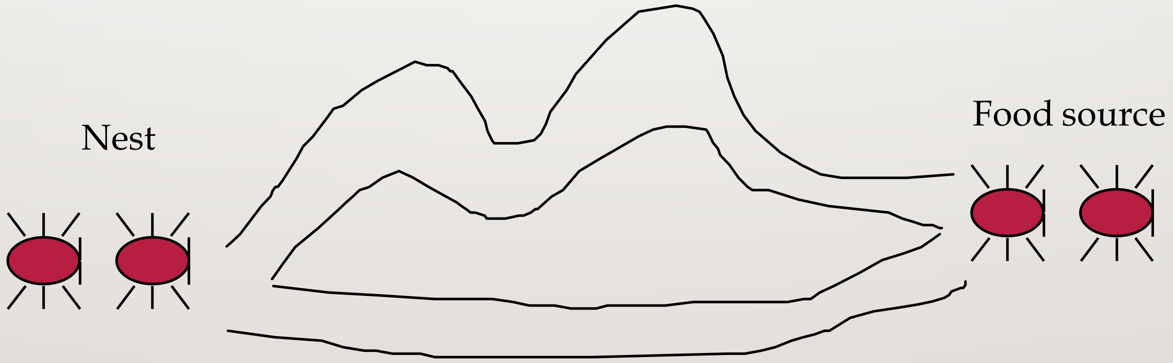
Shorter path reinforced.

STIGMERGY IN ANTS

- Ants are behaviorally unsophisticated, but collectively they can perform complex tasks.
- Ants have *highly developed sophisticated sign-based stigmergy*
  - They communicate using pheromones;
  - They **lay trails of pheromone** that can be followed by other ants.
- If an ant has a **choice of two pheromone trails** to follow, one to the NW, one to the NE, but the NW one is *stronger* – which one will it follow?

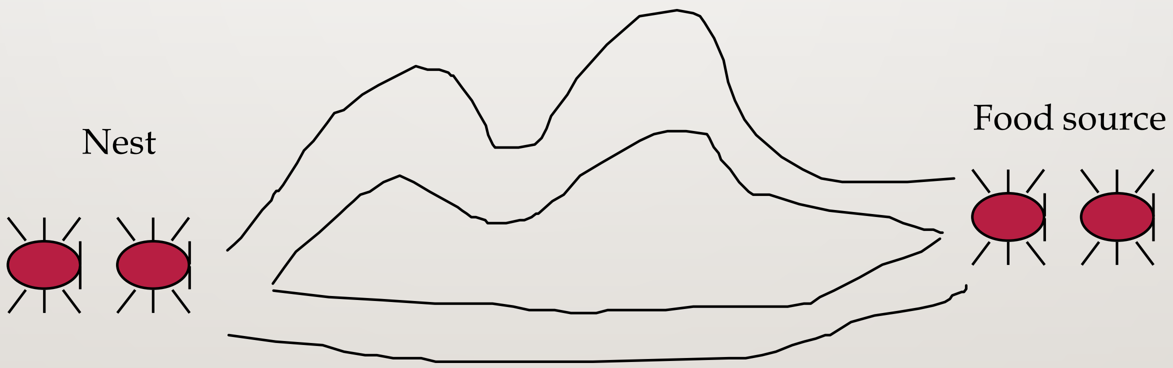
PHEROMONE TRAILS

- Individual ants lay pheromone trails while travelling from the nest, to the nest or possibly in both directions.
- The pheromone trail gradually evaporates over time.
- But pheromone trail strength accumulate with multiple ants using path.

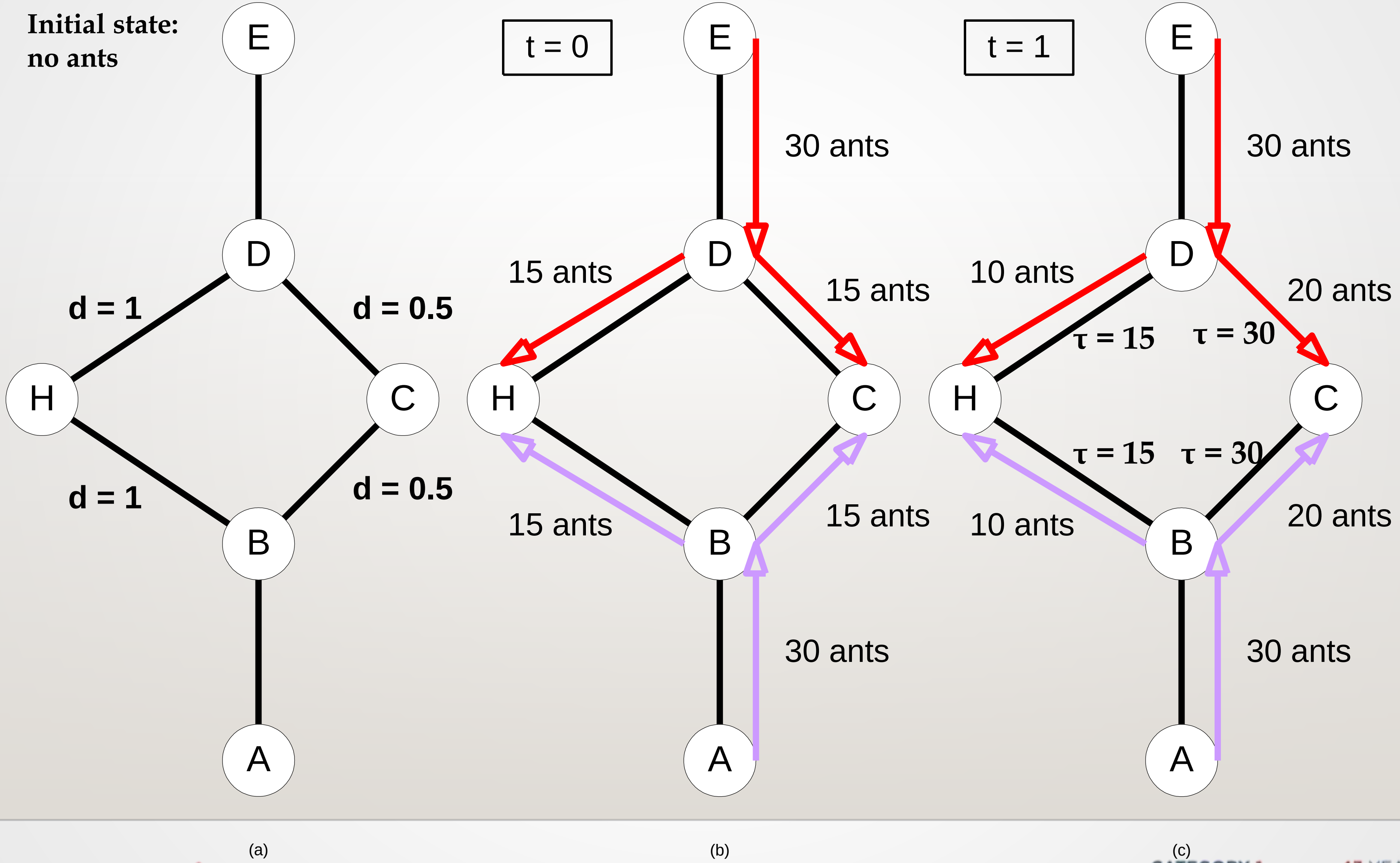


PROPERTIES OF THE PHEROMONE

- The pheromone is olfactive and volatile.
- The pheromone is stronger if more ants go along the same path(reinforced by number).
- The pheromone is stronger if the path from the nest to the food is shorter.



Pheromone Trails continued



ANT COLONY OPTIMISATION ALGORITHMS: BASIC IDEAS

- Ants are *agents* that:
  - Move along between nodes in a graph.
- They choose where to go based on pheromone strength.
- An ant's path represents a specific candidate solution.
- When an ant has finished a solution, pheromone is laid on its path, according to quality of solution.
- This pheromone trail affects behaviour of other ants by `stigmergy' …

USING ACO

- The optimization problem must be written in the form of a path finding problem with a weighted graph
- The artificial ants search for “good” solutions by moving on the graph
  - Ants can also build infeasible solutions – which could be helpful in solving some optimization problems
- The meta heuristic is constructed using three procedures:
  - Construct Ants Solutions
  - Update Pheromones
  - Daemon Actions

CONSTRUCT ANTS SOLUTIONS

- Manages the colony of ants.
- Ants move to neighboring nodes of the graph.
- Moves are determined by stochastic local decision policies based on pheromone trails and heuristic information.
- Evaluates the current partial solution to determine the quantity of pheromones the ants should deposit at a given node.

UPDATE PHEROMONES

- Process for modifying the pheromone trails
- Modified by
  - Increase
    - Ants deposit pheromones on the nodes (or the edges)
  - Decrease
    - Ants don't replace the pheromones and they evaporate
- Increasing the pheromones increases the probability of paths being used (i.e., building the solution)
- Decreasing the pheromones decreases the probability of the paths being used (i.e., forgetting)

DAEMON ACTIONS

- Used to implement larger actions that require more than one ant
- Examples:
  - Perform a local search
  - Collection of global information

A GENERAL ALGORITHM

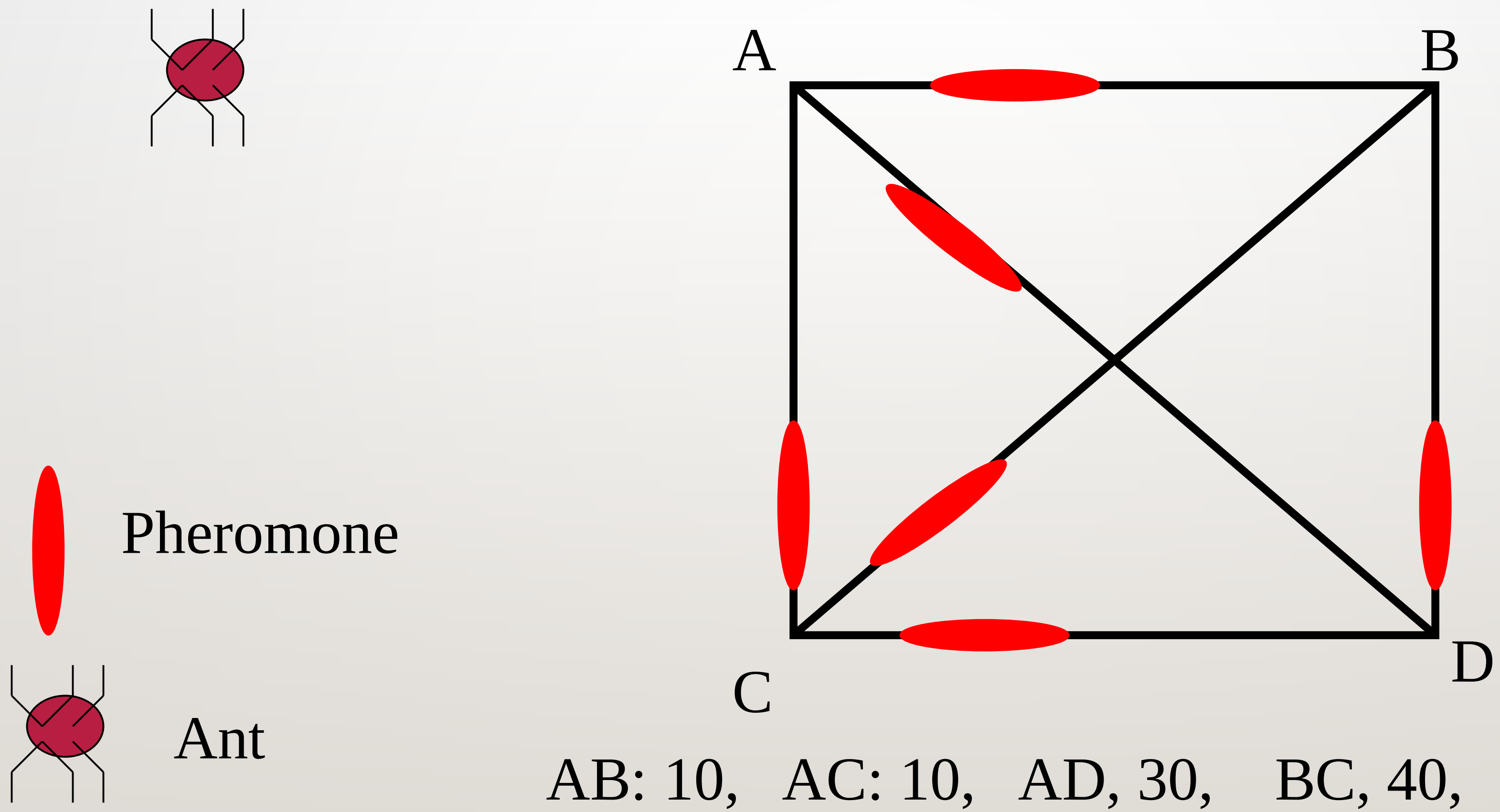
- Step1: Initialize the pheromone information.
- Step 2 : for each ant, do the following:
  - Find a solution (a path) based on the current pheromone trail.
  - Reinforcement : add pheromone.
  - Evaporation: reduce pheromone.
- Step 3 : stop if terminating condition satisfied, return to step 2 other wise.

APPLICATIONS OF ACO

- Vehicle routing with time window constraints
- Network routing problems
- Assembly line balancing
- Heating oil distribution
- Data mining
- Robotic Path Problem

E.G. A 4-CITY TSP

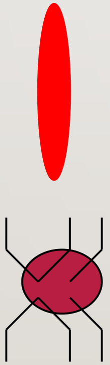
Initially, random levels of pheromone are scattered on the edges



AB: 10, AC: 10, AD, 30, BC, 40, CD 20

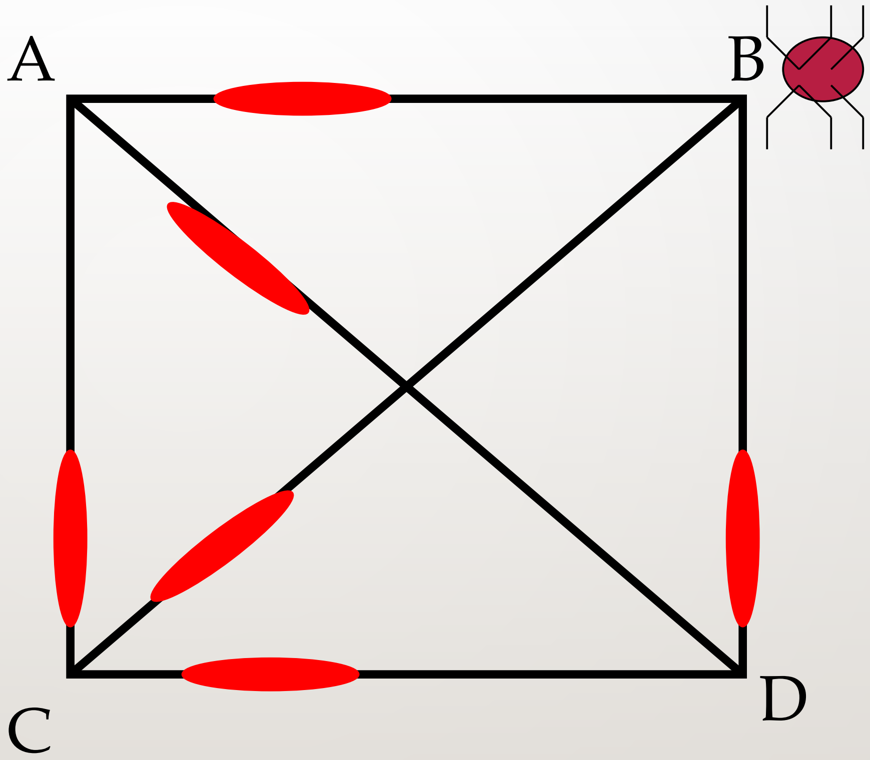
E.G. A 4-CITY TSP

An ant is placed at a random node



Pheromone

Ant

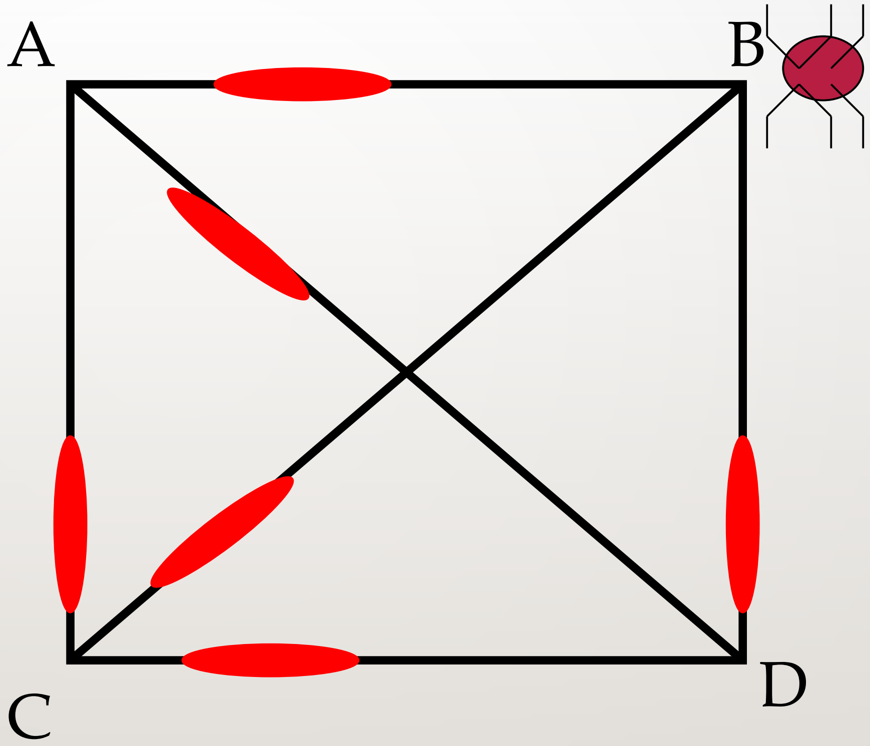


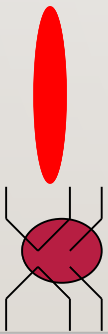
AB: 10, AC: 10, AD, 30, BC, 40, CD 20

E.G. A 4-CITY TSP

The ant decides where to go from that node, based on probabilities calculated from:
- pheromone strengths,
- next-hop distances.

Suppose this one chooses BC



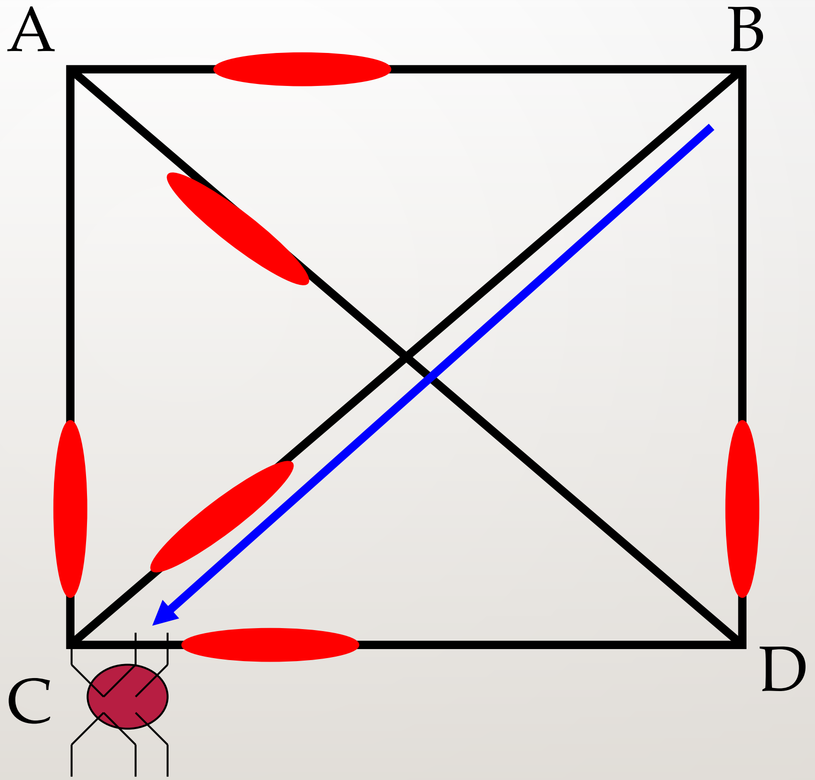


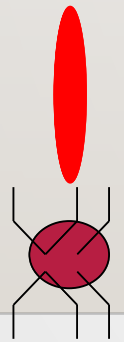
AB: 10, AC: 10, AD, 30, BC, 40, CD 20

E.G. A 4-CITY TSP

The ant is now at C, and has a `tour memory' = {B, C} – so he cannot visit B or C again.

Again, he decides next hop (from those allowed) based on pheromone strength and distance; suppose he chooses CD





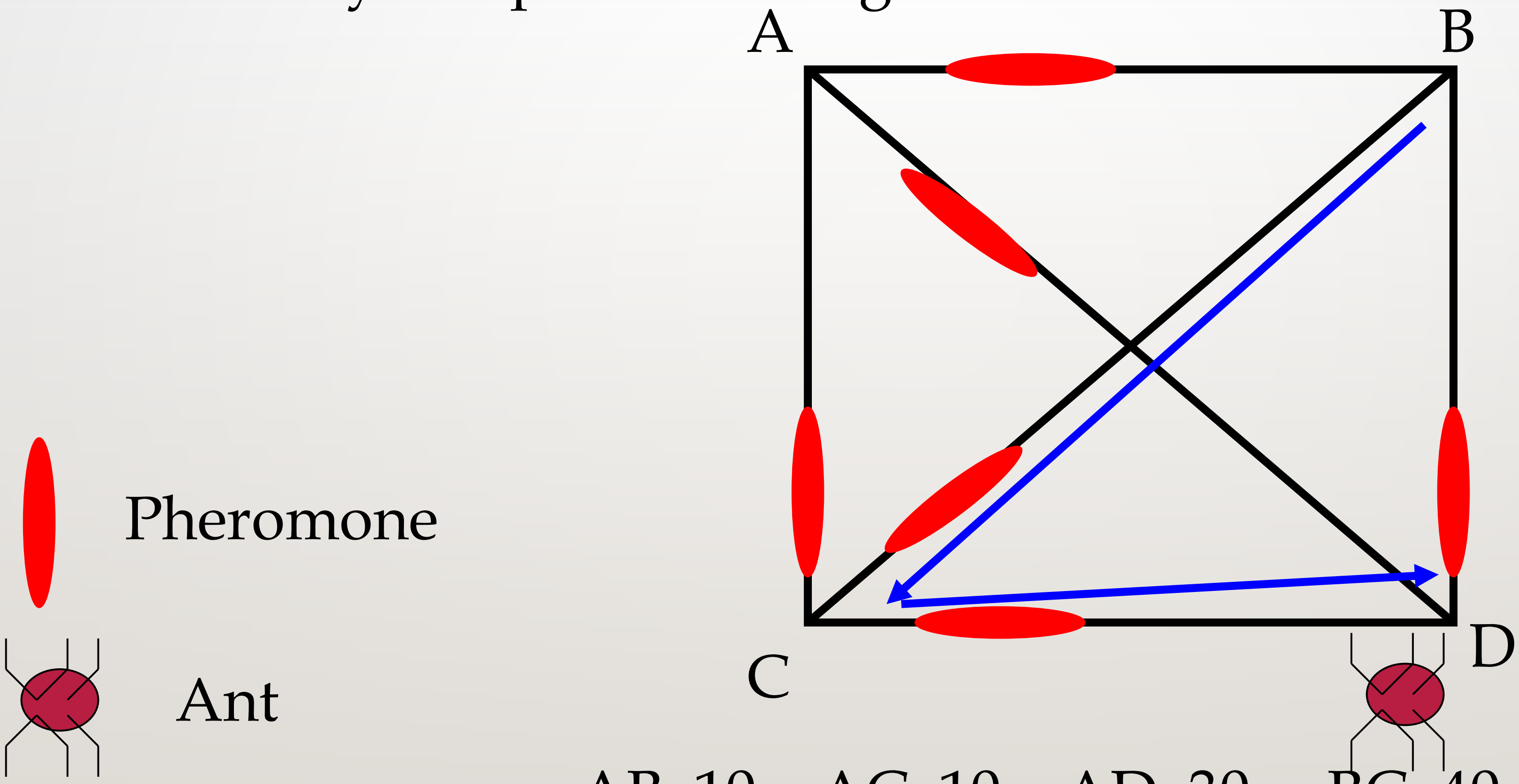
Pheromone

Ant

AB: 10, AC: 10, AD, 30, BC, 40, CD 20

E.G. A 4-CITY TSP

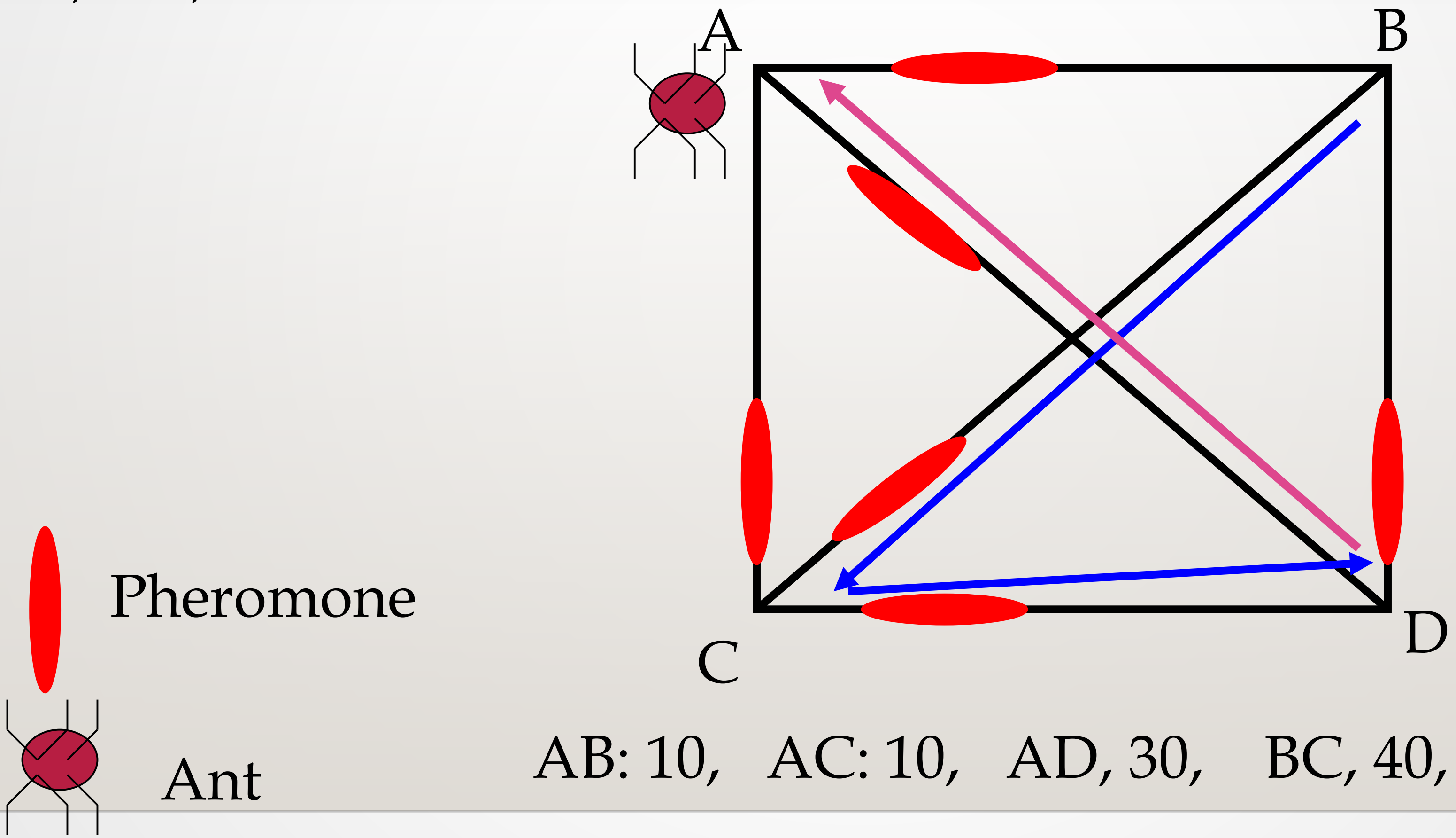
The ant is now at D, and has a `tour memory' = {B, C, D}
There is only one place he can go now:



AB: 10, AC: 10, AD, 30, BC, 40, CD 20

E.G. A 4-CITY TSP

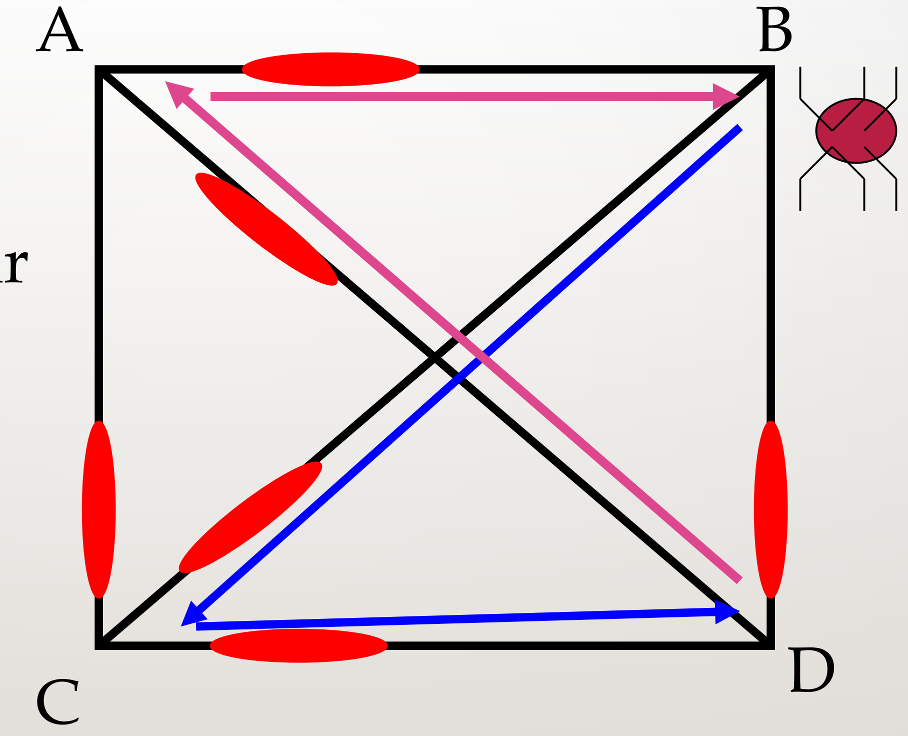
So, he has nearly finished his tour, having gone over the links: BC, CD, and DA.



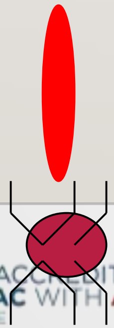
AB: 10, AC: 10, AD, 30, BC, 40, CD 20

E.G. A 4-CITY TSP

So, he has nearly finished his tour, having gone over the links: BC, CD, and DA. AB is added to complete the round trip.



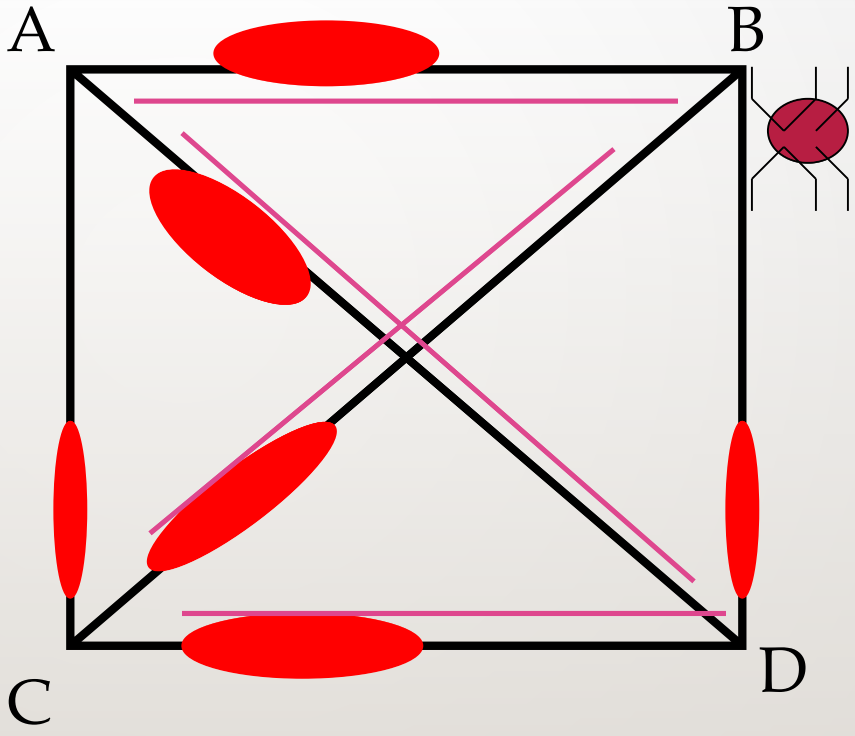
Now, pheromone on the tour is increased, in line with the fitness of that tour.

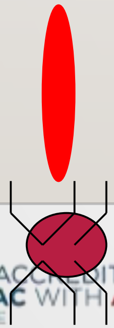


AB: 10, AC: 10, AD, 30, BC, 40, CD 20

E.G. A 4-CITY TSP

Next, pheromone everywhere is decreased a little, to model decay of trail strength over time





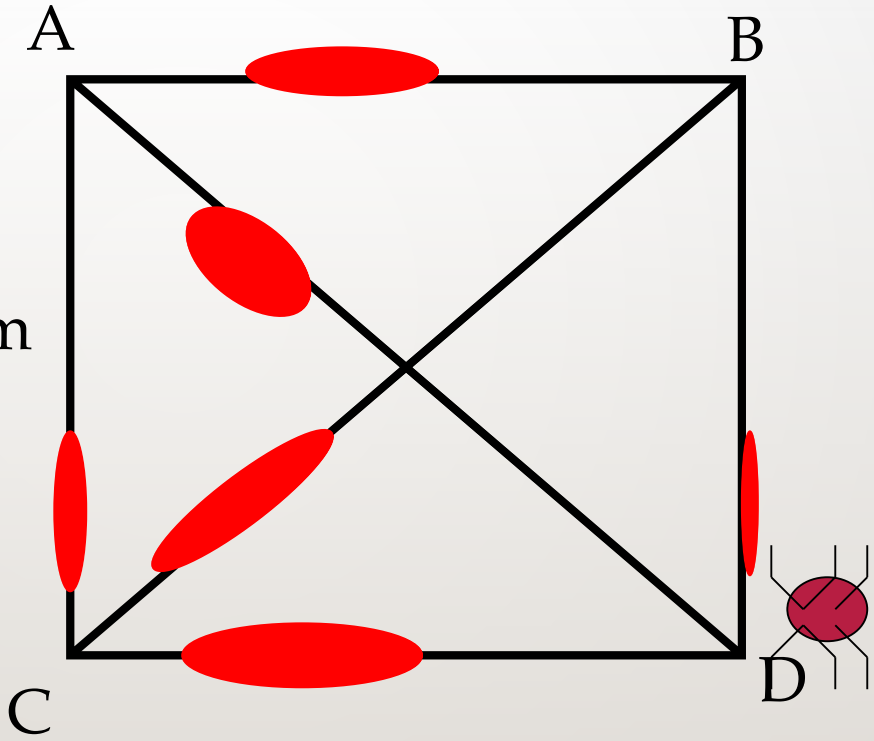
AB: 10, AC: 10, AD, 30, BC, 40, CD 20

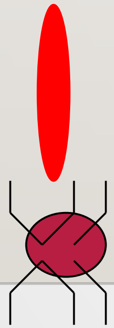
E.G. A 4-CITY TSP

We start again, with another ant in a random position.

Where will he go?

Next , the actual algorithm and variants.





Pheromone

Ant

AB: 10, AC: 10, AD, 30, BC, 40, CD 20

THE ACO ALGORITHM FOR THE TSP

We have a TSP, with *n* cities.

1. We place some ants at each city. Each ant then does this:
    - It makes a complete tour of the cities, coming back to its starting city, using a *transition rule* to decide which links to follow. By this rule, it chooses each next-city at random, but based partly by the pheromone levels existing at each path, and based partly by *heuristic information.*
2. When all ants have completed their tours.

    Global Pheromone Updating occurs.

    - The current pheromone levels on all links are reduced (I.e. pheromone levels decay over time).
    - Pheromone is laid (belatedly) by each ant as follows: it places pheromone on all links of its tour, with strength depending on how good the tour was.

THE ACO ALGORITHM FOR THE TSP [A SIMPLIFIED VERSION WITH ALL ESSENTIAL DETAILS]

We have a TSP, with *n* cities.

1. We place some ants at each city. Each ant then does this:
   - It makes a complete tour of the cities, coming back to its starting city, using a *transition rule* to decide which links to follow. By this rule, it chooses each next-city at random, but biased partly by the pheromone levels existing at each path, and biased partly by *heuristic information.*
2. When all ants have completed their tours.

   Global Pheromone Updating occurs.

   - The current pheromone levels on all links are reduced (I.e. pheromone levels decay over time).
   - Pheromone is lain (belatedly) by each ant as follows: it places pheromone on all links of its tour, with strength depending on how good the tour was.

Then we go back to 1 and repeat the whole process many times, until we reach a termination criterion.

ACO Algorithm

Set all parameters and initialize the pheromone trails

Loop

Sub-Loop

Construct solutions based on the state transition rule

Apply the online pheromone update rule

Continue until all ants have been generated

Apply Local Search

Evaluate all solutions and record the best solution so far

Apply the offline pheromone update rule

Continue until the stopping criterion is reached

THE TRANSITION RULE

$T(r,s)$ is the amount of pheromone currently on the path that goes directly from city r to city s.

$H(r,s)$ is the heuristic value of this link – in the classic TSP application, this is chosen to be 1/distance(r,s) -- i.e. the shorter the distance, the higher the heuristic value.

$p_k(r,s)$ is the probability that ant k will choose the link that goes from r to s

β is a parameter that we can call the *heuristic strength*

The rule is:

$$p_k(r,s) = \frac{T(r,s)\cdot H(r,s)^{\beta}}{\sum_{\text{unvisited cities } c} T(r,c)\cdot H(r,c)^{\beta}}$$

Where our ant is at city r, and s is a city as yet unvisited on its tour, and the summation is over all of k's unvisited cities

GLOBAL PHEROMONE UPDATE

$A_k(r,s)$	is amount of pheromone added to the (r, s) link by ant k.
m	is the number of ants
ρ	is a parameter called the pheromone decay rate.
L_k	is the length of the tour completed by ant k

$T(r, s)$ at the next iteration becomes:

$$\rho \cdot T(r,s) + \sum_{k=1}^{m} A_k(r,s)$$

Where $A_k(r,s) = 1/L_k$

Ant Colony Optimization

Characteristics

- An ant is a solution.
- Solutions (ants) are at different places in the solution space.
- How they change is based on the probability of changing to a different schedule.
- An ant completes its tour after selection a choice for each stand.
- Utilities (objective function values) of each tour are calculated.
- Pheromone levels are updated after all of the ants have completed all of their tours.

Ant Colony Optimization

Advantages:

- It is intuitive to biologically-minded people, mimicking nature.

- The system is built on positive feedback (pheromone attraction) and negative attractiveness (pheromone evaporation).

- Pheromone evaporation helps avoid convergence to a local optima.

Disadvantages:

- For routing problems it may make more sense, but for harvest scheduling problems, it requires a conceptual leap of faith.

- Fine-tuning the sensitive parameters may require significant effort.

REFERENCES

- Ant Colony Optimization by Marco Dorigo and Thomas Stützle, The MIT Press, 2004
- Swarm Intelligence by James Kennedy and Russell Eberhart with Yuhui Shi, Morgan Kauffmann Publishers, 2001
- Advances in Applied Artificial Intelligence edited by John Fulcher, IGI Publishing, 2006
- Data Mining: A Heuristic Approach by Hussein Abbass, Ruhul Sarker, and Charles Newton, IGI Publishing, 2002
- "Ant Colony Optimization" Curatored by Marco Dorigo, http://www.scholarpedia.org/article/Ant_Colony_Optimization
- "Ant Colony Optimization" by Marco Dorigo, http://iridia.ulb.ac.be/~mdorigo/ACO/ACO.htm;
- "Swarm Intelligence" http://en.wikipedia.org/wiki/Swarm_intelligence
- Picture of Flik, http://www.pixar.com


Thank
you