Computational Intelligence

Samaneh Hosseini

Isfahan University of Technology

Ant Colony Optimization

Ants: An Introduction

Inspiration From Nature

1 quadrillion ants
7 billion humans

Ants

- There are about 10^{18} living insects
- About 2% of all insects are social
- About 50% of all social insects are ants
- The total weight of ants is about the total weight of humans
- Ant colony size can be as few as 30 ants and as large as million ants
- Ants colonize the world since 100,000,000 years, humans only since 50,000 years

Ants

- Ants use Stigmergy that was discovered by Pierre-Paul Grasse who was studying termites.

- Properties of Stigmergy
  - indirect agent interaction modification of the environment
  - environmental modification serves as external memory
  - work can be continued by any individual

- Ants walking, to or from, a food source deposit a chemical substance on its way,
  - This substance is referred to as the ***pheromone***.

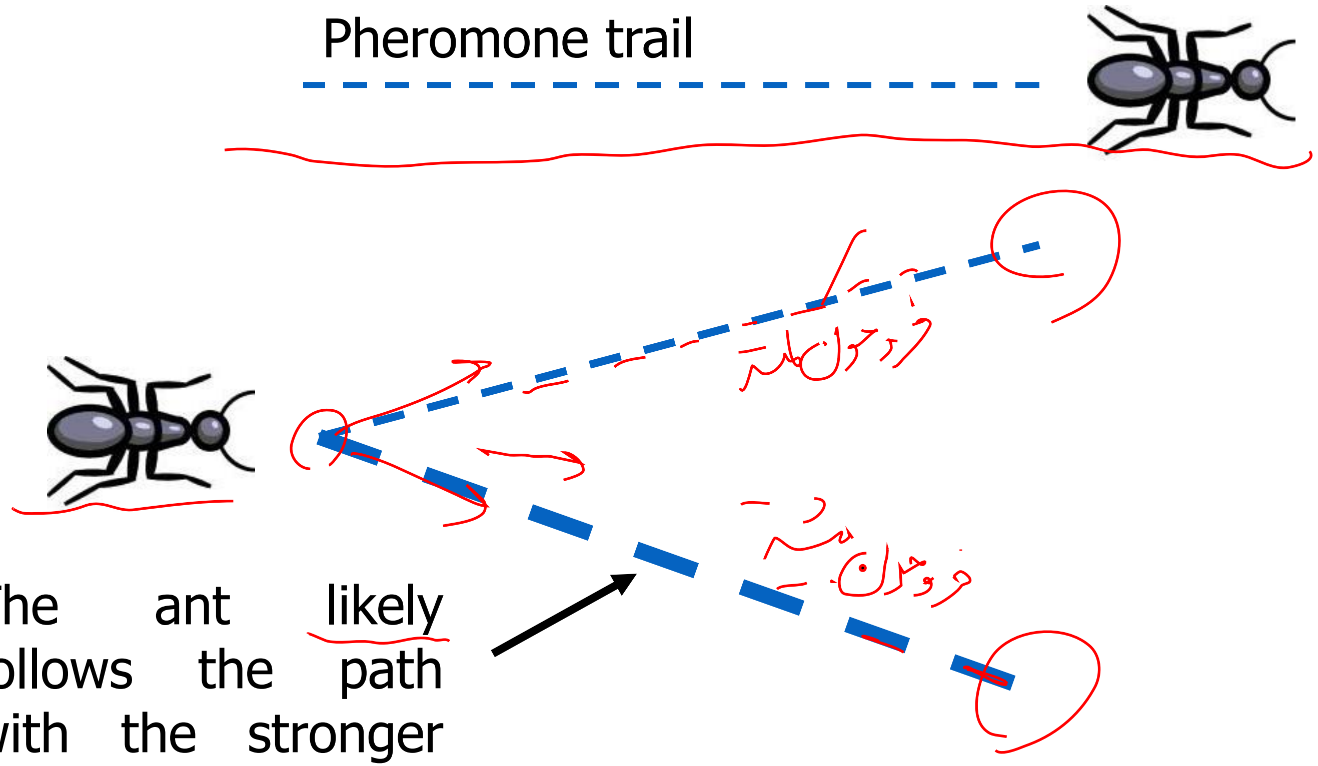
- P. P. Grasse."Recherches sur la biologie des termites champignonnistes (*Macrotermitina*)". Ann. Sc. Nat., Zool. Biol. anim., 6, 97, 1944.

- P. P. Grasse."La reconstruction du nid et les coordinations interindividu- elles chez *bellicositermes natalensis* et *cubitermes* sp. La theorie de la stig- mergie: essai d'interpretation du comportement des termites constructeurs". Insectes Sociaux, 6, 41, 1959.

Pheromone

- Other ants can smell this pheromone,
- Its presence will influence the ants' choice of a certain path,
- Ants tend to follow strong pheromone concentrations.
- ***Pheromone trails***, is formed by the pheromone deposited on the ground,
- Pheromone trails help the ants to reach good food sources that have been previously identified by other ants.

Pheromone

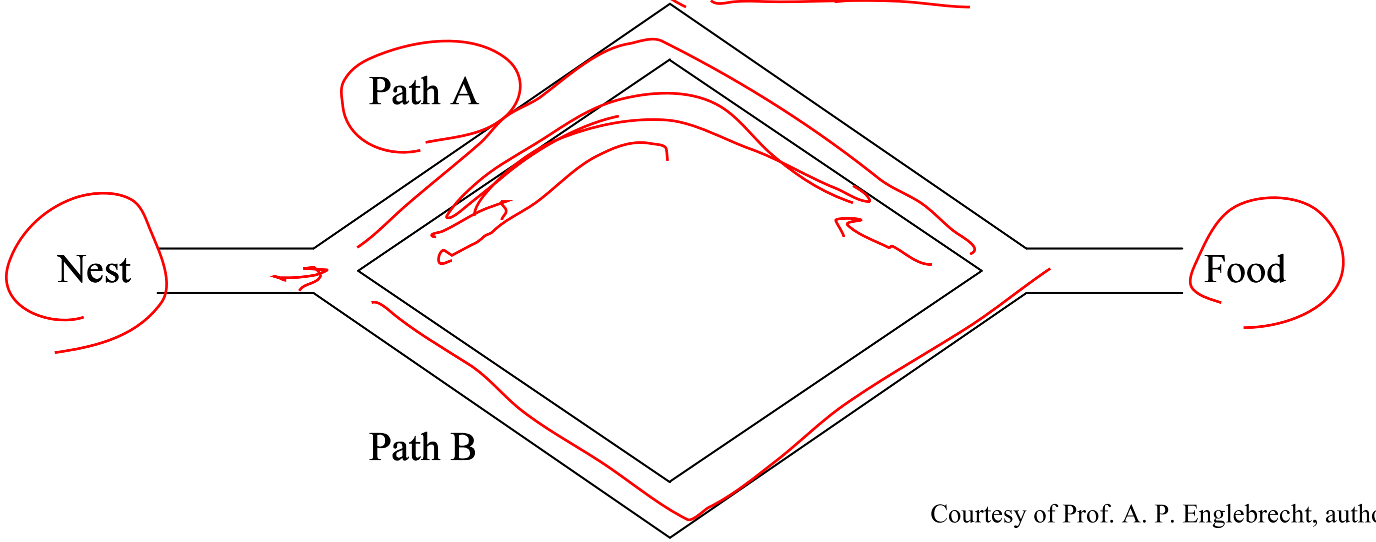


The ant likely follows the path with the stronger pheromone trail

احتمال انتخاب مسیرها
فرومون بیشتر بالاتر است

Binary Bridge Experiment

- A ***binary bridge experiment*** was done by Deneubourg et al.,
- The ants nest was connected to a food source through two equal length bridges,
- The ants behaviour was observed until they converge towards the use of the same bridge.



Courtesy of Prof. A. P. Englebrecht, author of text

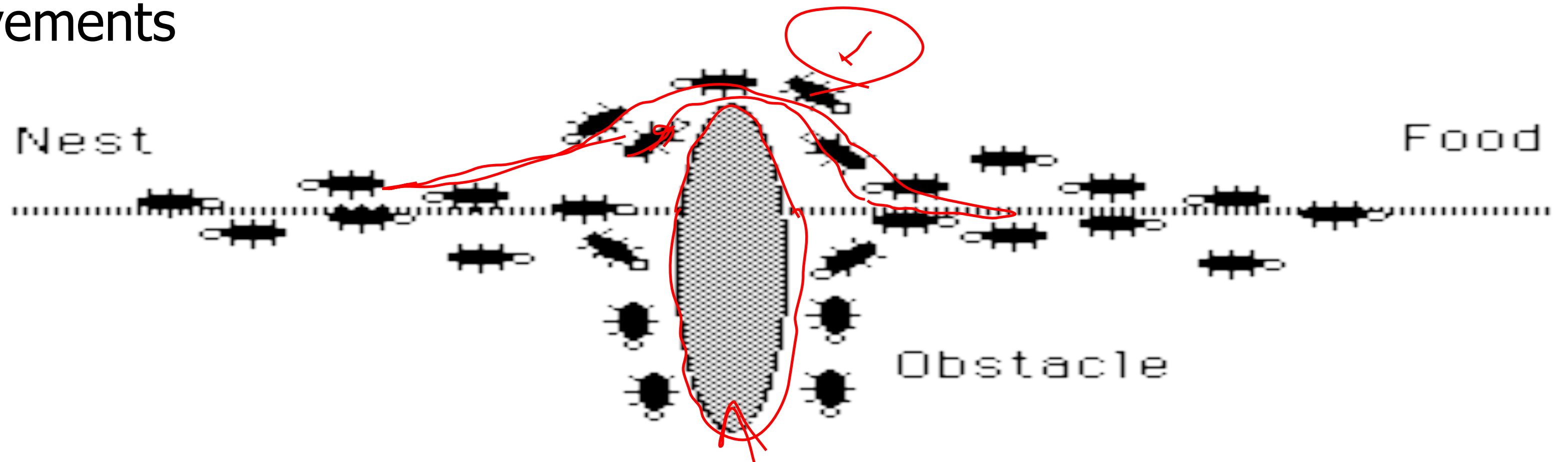
J. L. Deneubourg, S. Aron, S. Goss and J. M. Pasteels."The self- organizing exploratory pattern of the Argentine ant". Journal of Insect Behaviour, 3, 159, 1990.

Binary Bridge Experiment

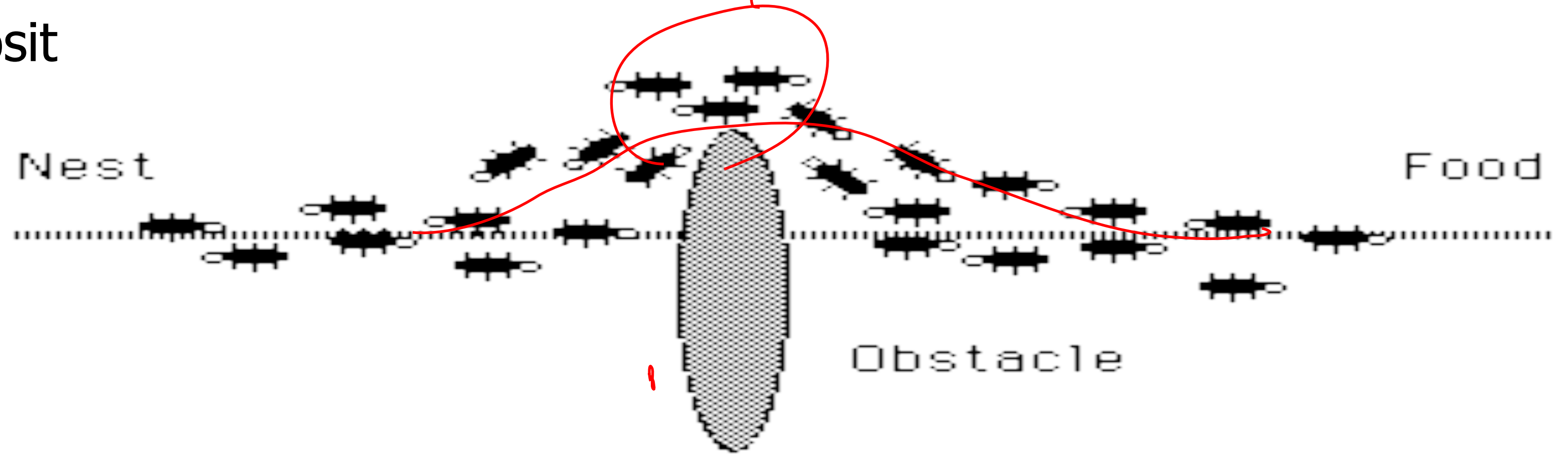
- Initially, the ants randomly choose the bridge to follow,
- After a while, the ants will tend to follow the bridge with ***stronger pheromone trails***,
- The selection of one bridge is due to ***random fluctuations*** causing it to have higher pheromone concentration.
- Pheromone trail acts as a ***collective memory*** for the ants to communicate through by sensing and recording their foraging experience
- Pheromone ***evaporates*** over time introducing some changes in the environment.

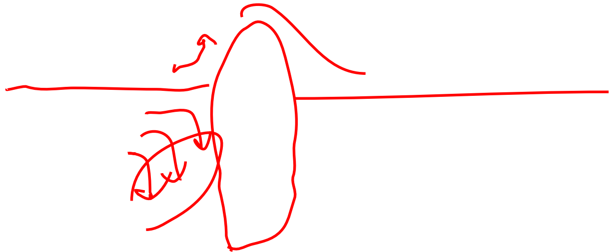
Shortest Path Around an Obstacle

Initial movements



After deposit





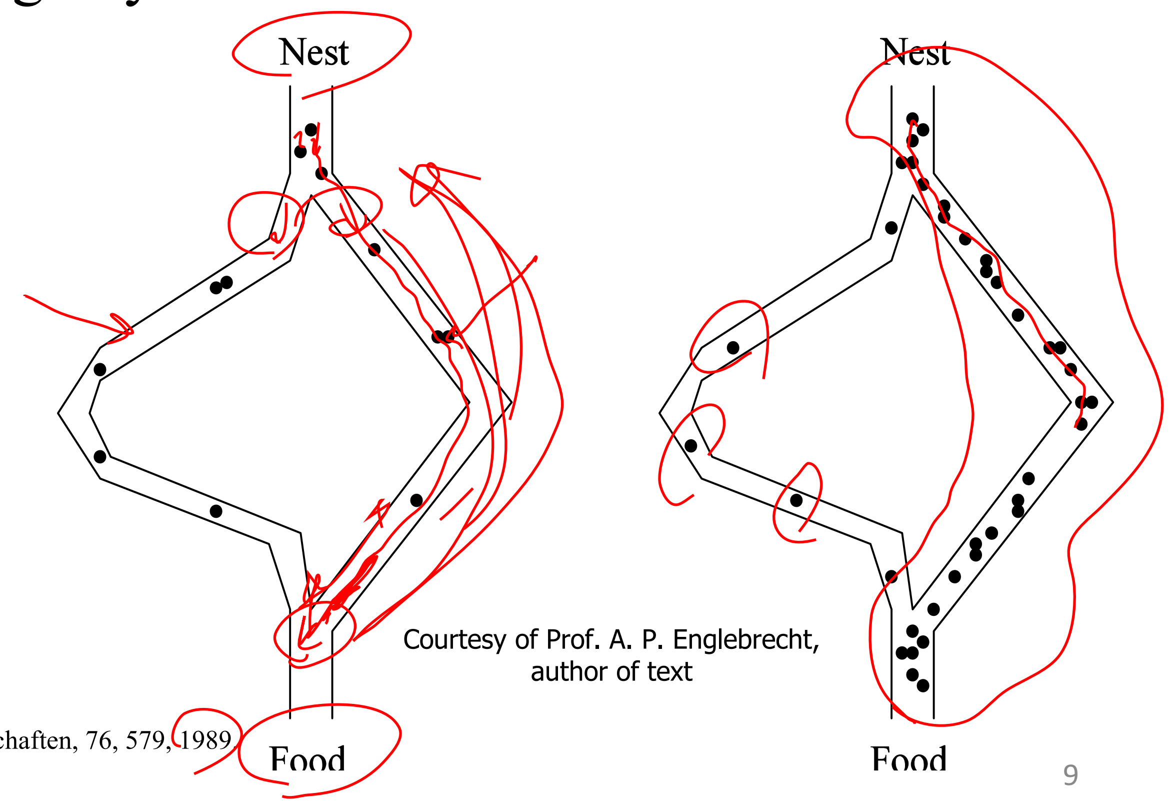
Bridges with Non-equal Length

- Another experiment was carried by Goss et al. [4] were the bridges were not of equal length. One bridge was significantly shorter than the other,

- In this case, the ants following the shorter bridge by chance will be the first to reach the food source.

- They will be also the first to reach the nest as they will take the same path home; since it will have more pheromone,

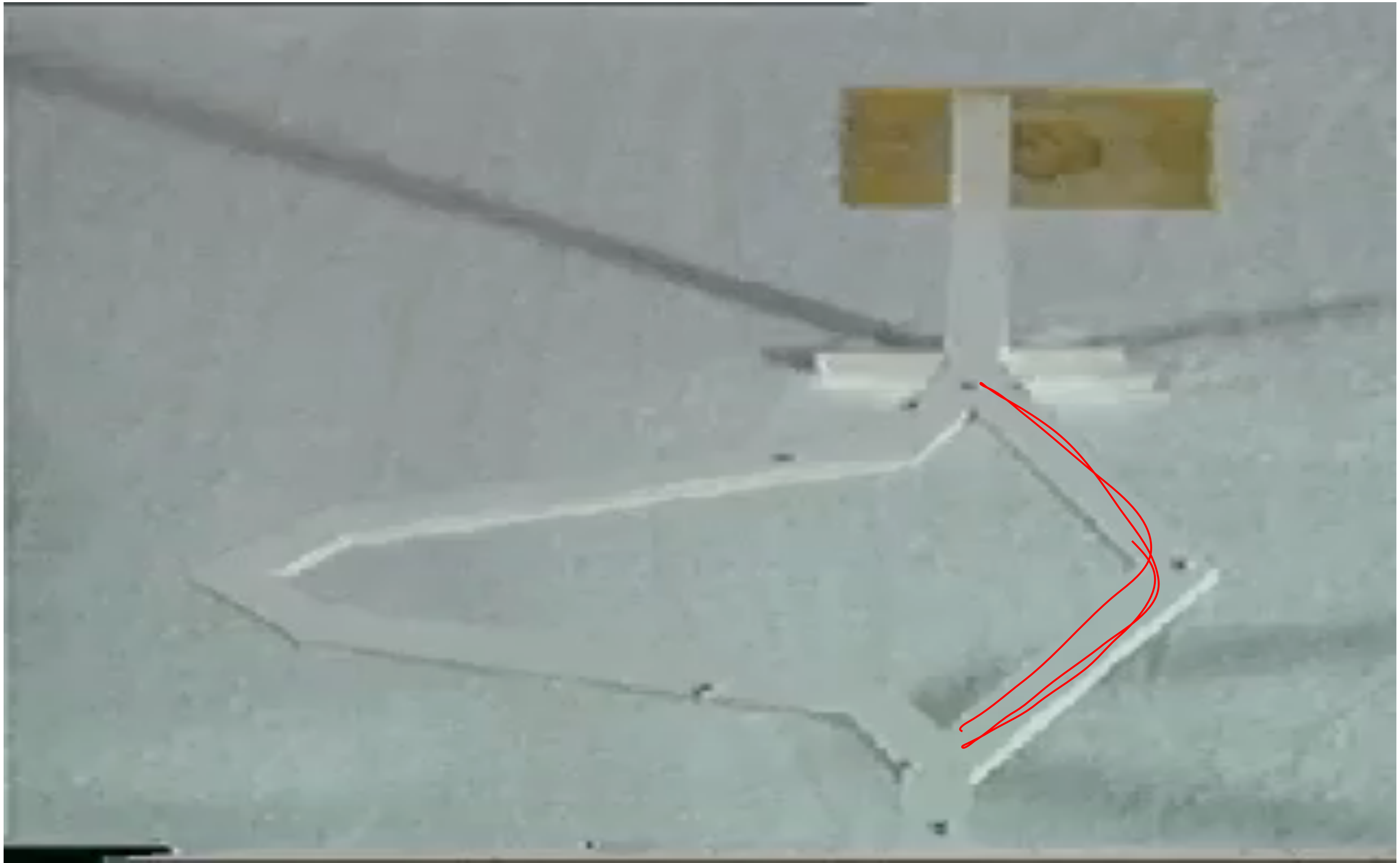
- The ants finally converged to the shorter path due to the pheromone depositing mechanism.



Courtesy of Prof. A. P. Englebrecht, author of text

S. Goss, S. Aron, J. L. Deneubourg and J. M. Pasteels."Self-organized shortcuts in the Argentine ant". Naturwissenschaften, 76, 579, 1989

Bridges with Non-equal Length



Ant Colony Optimization (ACO)

Algorithm

ACO
AS
Min-Max AS

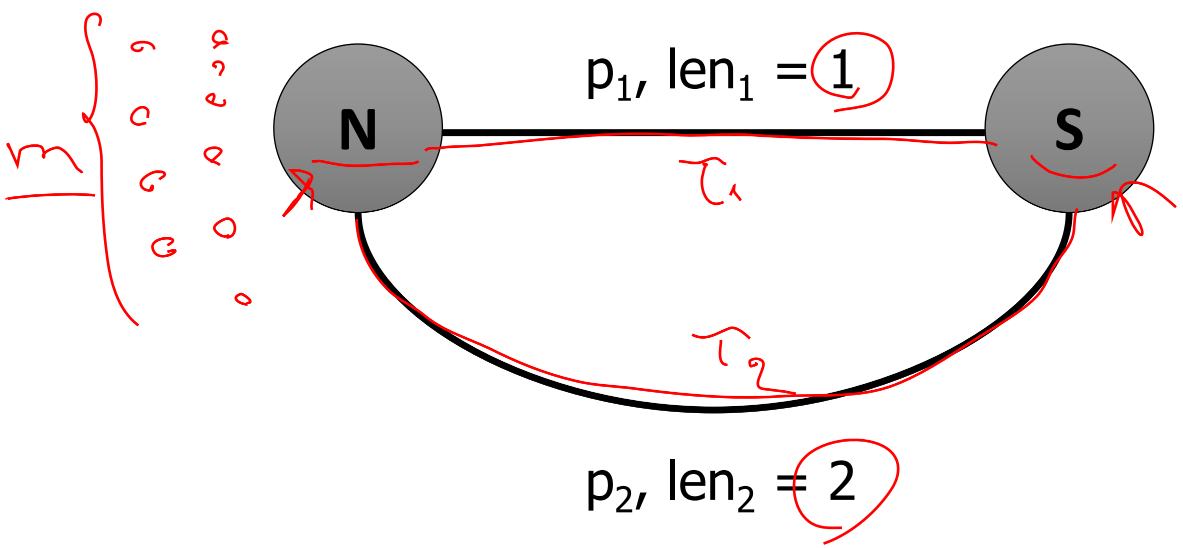
ACO - Introduction

- Introduced by M. Dorigo in 1992,
- Inspired by the foraging behaviour or real ants,
- Successfully applied in many application including:
  - Traveling Salesman Problem (TSP),
  - Packet Routing in Network Telecommunications,
  - Scheduling Problems,
  - Vehicle Routing Problems.

M. Dorigo."Optimization, Learning and Natural Algorithms". Ph.D. thesis, DEI, Politecnico di Milano, Italy, pp. 140, 1992.

ACO - Simulation

- To simulate the behaviour of ants: assume we have a nest and a food source connected through two paths with different lengths.



- Assign initial artificial pheromone values for every path, τ_1 and τ_2,
- Initially, both values are equal: $\tau_1 = \tau_2 = c > 0$

ACO - Simulation

- Place m ants at the nest,

- For each ant k,
  - Traverses path 1 with a probability:

احتمال انتخاب مسیر اول

$$p_1^k = \frac{\tau_1}{\tau_1 + \tau_2}$$

$\frac{1}{2}$

$\frac{c}{c+c} = \frac{1}{2}$

  - Traverses path 2 with a probability:

احتمال انتخاب مسیر دوم

$$p_2^k = 1 - p_1$$

ACO - Simulation

- An evaporation phase is applied:
  - To simulate the evaporation of real pheromone,
  - To avoid quick convergence to sub-optimal paths.

$$\tau_i = (1-\rho) \times \tau_i, \quad \rho \in (0,1)$$

ρ specifies the rate of evaporation, when equal to 1, the move becomes random.

- Each ant leaves more pheromone on its traversed path:

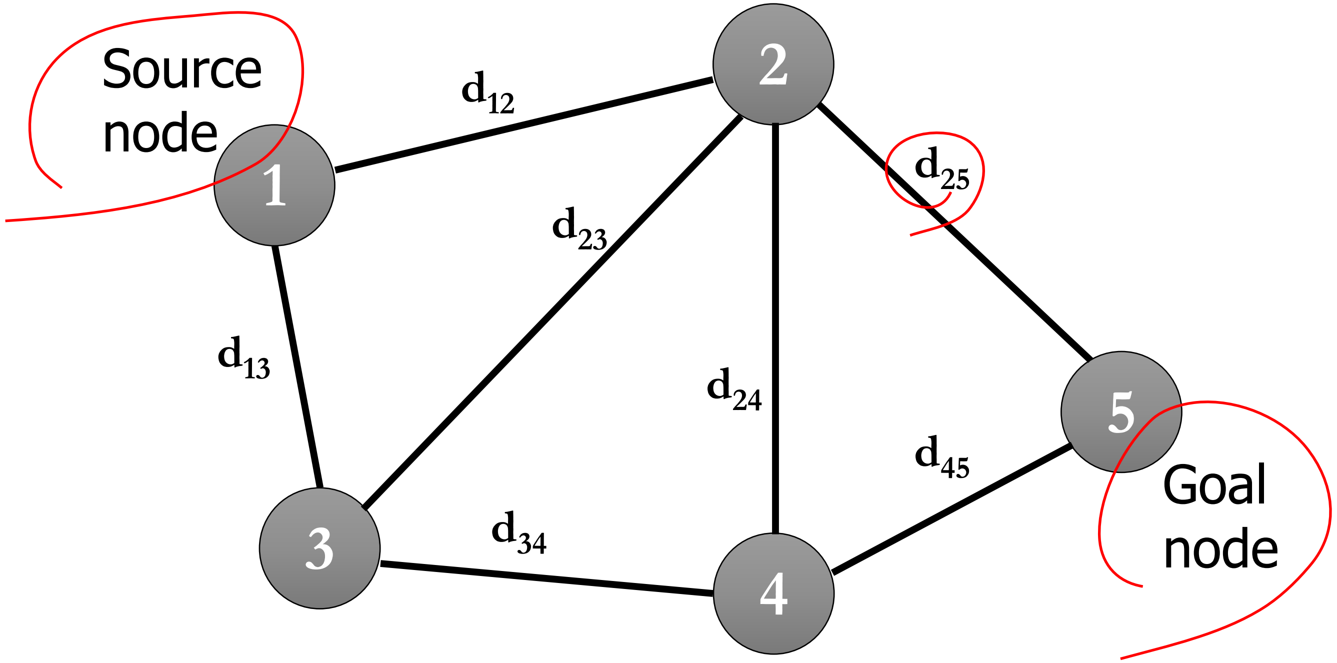
$$\tau_i = \tau_i + \frac{1}{len_i}$$

Real AC vs. Artificial AC

Real AC		Artificial AC
Real ant and pheromone	⟷	Artificial ant (agent) and pheromone (value) τ_i
Food	⟷	Solution
Continuous	⟷	Discrete
Pheromone update while moving	⟷	Pheromone update after traverse
Solutions are evaluated implicitly	⟷	Explicit function to evaluate solutions

ACO - Algorithm

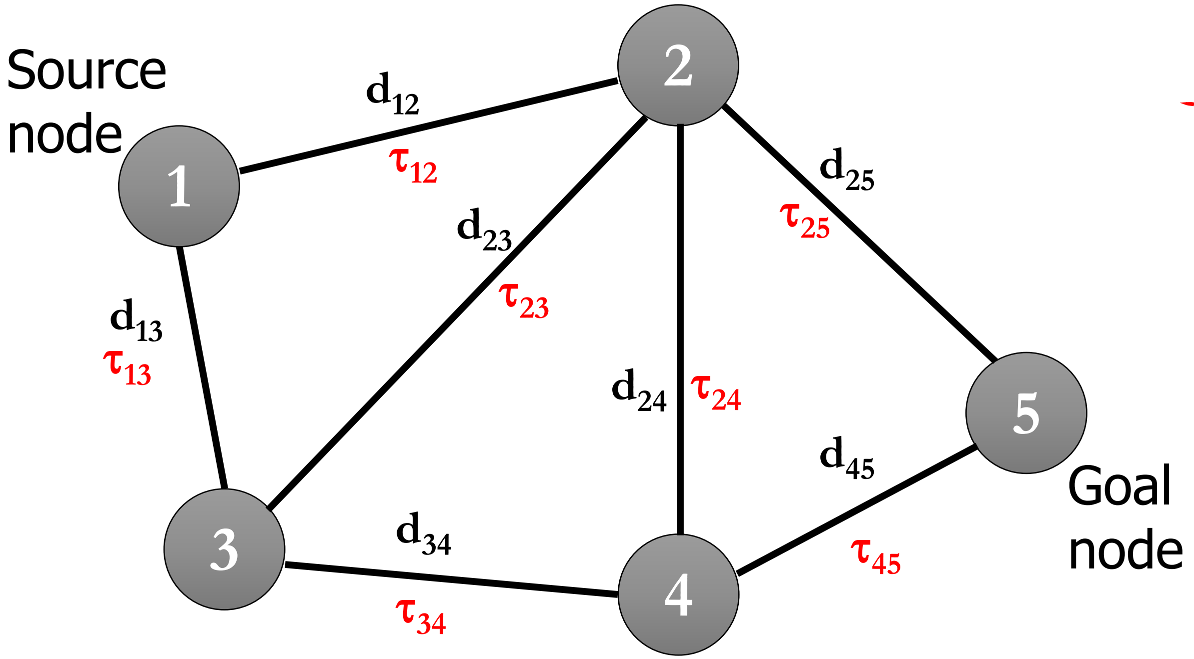
- Let $G=(N,E)$ denote a graph, where N is the set of nodes (vertices) and E is the set of arcs (edges),
- Each arc (i, j) is associated with a value d_{ij} denoting the ***distance*** between nodes i and j,



- A simple ant algorithm could be used to find the shortest path between two given nodes in the graph.

ACO - Algorithm

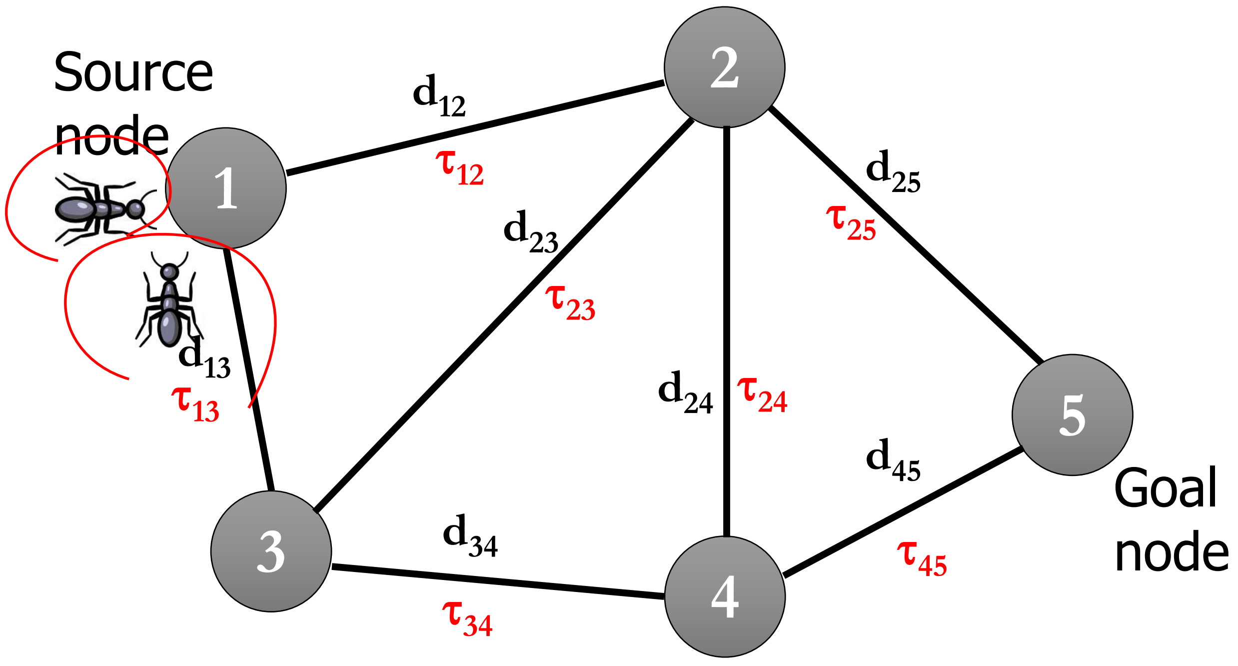
- Let each arc (i, j) be associated with a value τ_{ij} called the ***artificial pheromone***,
- At the beginning, the same small amount of artificial pheromone is placed on all arcs,



$\tau_{ij} = c \quad \forall ij$

ACO - Algorithm

- Place a group of ***m*** ants at the source node.

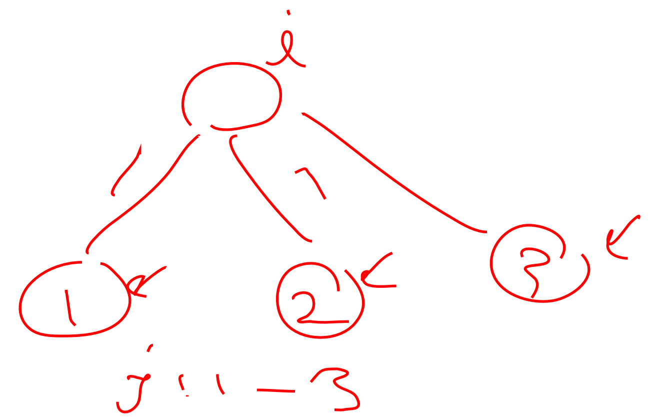


ACO - Algorithm

- At each node ***i***, the ant has a choice to move to any of the ***j*** nodes connected to it,

- Each node ***j*** $\in$ ***Ni*** connected to ***i*** has a probability to be selected by ant ***K*** equal to:

$$p_{ij}^{k} = \begin{cases} \dfrac{\tau_{ij}^{\alpha} / d_{ij}^{\beta}}{\sum_{n \in N_i} \tau_{in}^{\alpha} / d_{in}^{\beta}} & \text{if} \quad j \in N_i \\ 0 & \text{if} \quad j \notin N_i \end{cases}$$



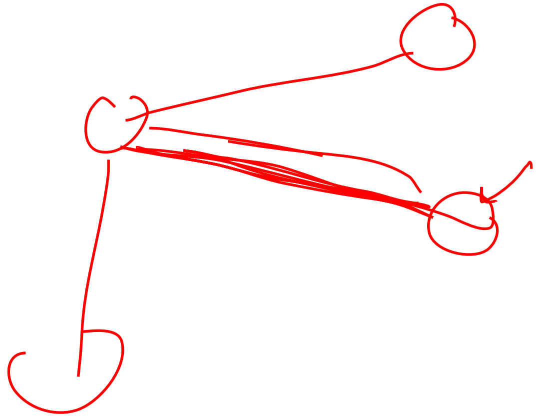
- α and β are chosen to balance the ***local*** vs. the ***global*** search ability of the ant.

ACO - Algorithm

- Evaporate the artificial pheromone:

$$\tau_i = (1-\rho)\times\tau_i, \quad \rho \in (0,1]$$

- The ant deposits extra pheromone on the arc it chooses:



$$\tau_{ij} = \tau_{ij} + \Delta\tau$$

- This will increase the probability of a subsequent ant choosing the same arc and referred to as ***online step_by_step*** pheromone update.

ACO

ACO - Algorithm

- Different approaches exist for choosing the value of $\Delta\tau$:
  - ***Ant density model***, adding ***Q***, a constant value, hence the final pheromone added to the edge will be proportional to the number of ants choosing it. This doesn't take the edge length into account,
  - ***Ant quantity model***, adding Q/d_{ij}, taking the edge length into account, hence enforcing the ant local search ability.

online
step-by-step

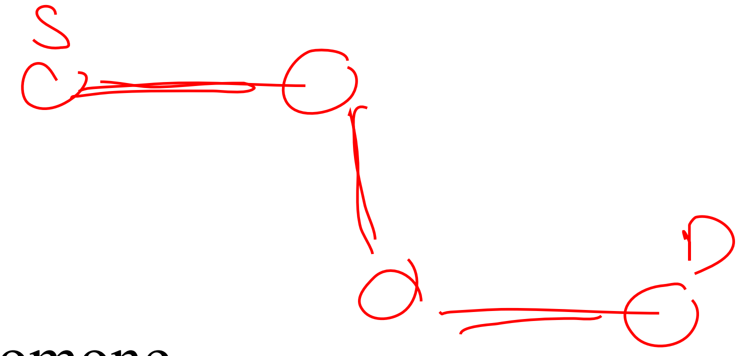
  - ***Online delayed*** pheromone update, sometimes referred to as the ***ant cycle model***,
    - After the ant builds the solution, it traces the path backward and updates the pheromone trails on the visited arcs according to the solution quality.

ACO - Algorithm

- The online delayed pheromone update is done by adding:

$$\Delta\tau_{ij} = \frac{Q}{L^k}$$

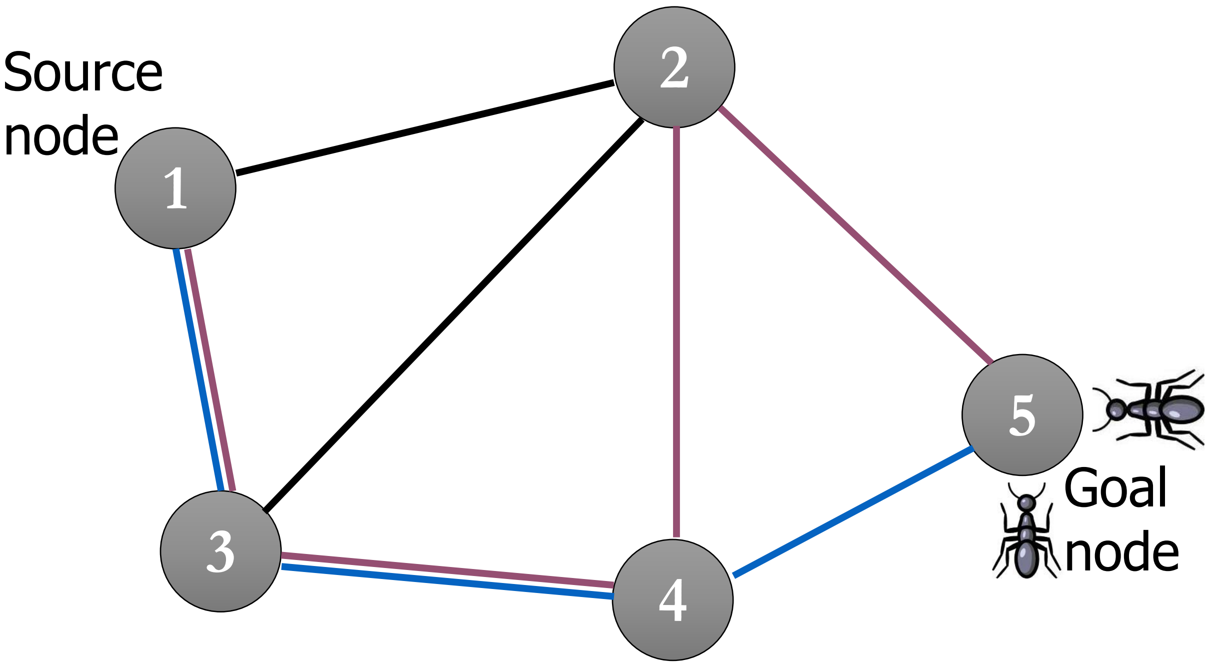
for every arc (i, j) on the path, where $\boldsymbol{L^k}$ is the length of the path found by ant $\boldsymbol{k}$.



- Done after the evaporation phase

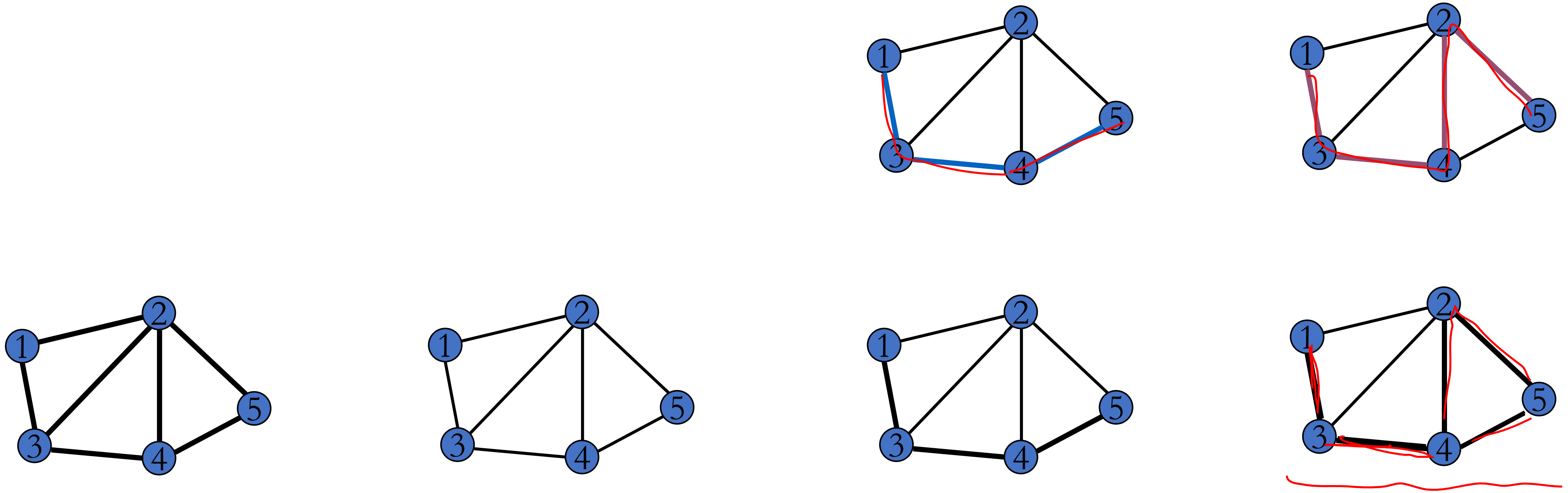
- Repeat the process for different initialization of pheromone

ACO - Algorithm



ACO - Algorithm

Start | **Evaporation** | **Solution 1** | **Solution 2**



ACO Metaheuristic

- Algorithm Ant colony optimization metaheuristic
- Set parameters, initialize pheromone trails α β ρ m Q $\tau_{i,j}$
- while termination conditions not met do
    - ConstructAntSolutions ρ
    - ApplyLocalSearch (optional)
    - UpdatePheromones
- end while

ACO algorithm

- Termination conditions:
  - Max number of iterations reached
  - Acceptable solution reached
  - All ants (or most of them) follow the same path, i.e stagnation.

Parameters

- **Number of ants**:
  - More ants more computations, but also more exploration.
- **Max number of iterations**: has to be enough to allow convergence.
- **Initial pheromone**: constant, random
- **Pheromone decay parameter** ρ

α β

Components

- **Transition rule:** probability of selection for the ant
- **Pheromone evaporation rule**
- **Pheromone update rule**
- **Problem heuristic if used**
- **Quality of solution measure**
- **Termination Criteria**

Ant System Algorithm

- ***Ant System Algorithm (AS)*:**
  - Proposed by Dorigo et al.,
  - Uses the same basic steps outlined in ACO,
  - The online delayed pheromone update is adopted using all the solutions of the current iteration.

probability with which ant k in city r chooses to move to the city s

$$p_k(r,s) = \begin{cases} \dfrac{[\tau(r,s)]\cdot[\eta(r,s)]^{\beta}}{\sum\limits_{u\in J_k(r)} [\tau(r,u)]\cdot[\eta(r,u)]^{\beta}}, & \text{if } s \in J_k(r) \\ 0, & \text{otherwise} \end{cases} \qquad (1)$$

where τ is the pheromone, $\eta = 1/\delta$ is the inverse of the distance $\delta(r,s)$, $J_k(r)$ is the set of cities that remain to be visited by ant k positioned on city r (to make the solution feasible), and β is a parameter which determines the relative importance of pheromone versus distance ($\beta > 0$).

$$\tau(r,s) \leftarrow (1-\alpha)\cdot\tau(r,s) + \sum_{k=1}^{m} \Delta\tau_k(r,s) \qquad (2)$$

where

$$\Delta\tau_k(r,s) = \begin{cases} \dfrac{1}{L_k}, & \text{if } (r,s) \in \text{tour done by ant } k \\ 0, & \text{otherwise} \end{cases}$$

- M. Dorigo.”Optimization, Learning and Natural Algorithms”. Ph.D. thesis, DEI, Politecnico di Milano, Italy, pp. 140, 1992.
- M. Dorigo, V. Maniezzo and A. Colorni.” Ant System: Optimization by a Colony of Cooperating Agents”. IEEE Trans. Syst., Man, and Cybern. Part B, vol. 26, no. 1, 1996.

Ant Colony System Algorithm

- ***<u>Ant Colony System Algorithm (ACS)</u>*:**
  - Proposed by Gambardella and Dorigo,

  - Based on AS but is different in:
    - Transition rule based on Elitist strategy (balances exploitation vs exploration)
    - Pheromone update rule
    - Local pheromone update

- Informally, the ACS works as follows: ants are initially positioned on cities chosen according to some initialization rule (e.g., randomly). Each ant builds a tour (i.e., a feasible solution to the TSP) by repeatedly applying a stochastic greedy rule (the 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. Once all ants have terminated their tour, the amount of pheromone on edges is modified again (by applying the global updating rule). As was the case in ant system, ants are guided, in building their tours, by both heuristic information (they prefer to choose short edges) and by pheromone information. An edge with a high amount of pheromone is a very desirable choice. The pheromone updating rules are designed so that they tend to give more pheromone to edges which should be visited by ants

M. Dorigo and L. M. Gambardella."Ant Colony System: a cooperative learning

ACS

- If $q < q_o$ (q is a random number, q_o is in(0,1)):

$$j = arg\,max_{j \in N_i^k} \left\{ \tau_{ij}(t) \eta_{ij}^{\beta} \right\}$$

- Else use:

$$p_{ij}^k(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha} [\eta_{ij}]^{\beta}}{\sum_{k \in allowed_k} [\tau_{ik}(t)]^{\alpha} [\eta_{ik}]^{\beta}} & \text{if } j \in \text{allowed}_k \\ 0 & \text{otherwise} \end{cases}$$

- J element of N^k_i is a randomly selected node give probability P^k_{ij}

- This creates bias towards choices of better quality, q_o controls this bias.

ACS

- Uses the Ant density model update, where all the ants update their last traversed edge:

$$\tau_{ij} = (1 - \rho_2)\tau_{ij} + \rho_2 \tau_0$$

- Introduced a new pheromone update rule (***using the best solution (ant) only***).

- delayed update: $$\tau_{ij}(t+1) = (1 - \rho_1)\tau_{ij}(t) + \rho_1 \Delta\tau_{ij}^{best}(t)$$

- Best here can be ***iteration best or global best***

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

Max-Min Ant System Algorithm

- ***Max-Min Ant System Algorithm (MMAS)*:**
  - Proposed by Stutzle and Hoos to overcome stagnation.
  - The update is done using the best solution (best ant) in the current iteration or the best solution over all, also decay in the update of the pheromone

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

  - The values of the pheromone are restricted between τ_{min} and τ_{max}, allows high exploration in the beginning (τ_{max}) and more intensification later.
  - The values of τ_{min} and τ_{max} are chosen experimentally, although they could be calculated analytically if the optimal solution is known,
  - Improved the performance significantly over AS.

T. Stutzle and H. H. Hoos."*MAX-MIN* Ant System". *Future Generation Comput. Syst.*, vol. 16, no 8, 2000.

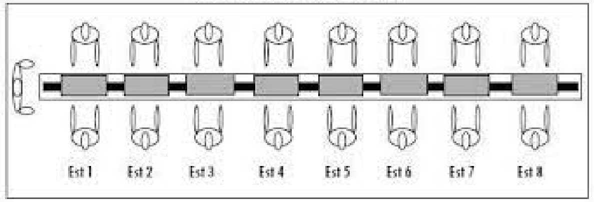
Ant Colony Optimization

Applications

ACO Applications

- Travelling Salesman Problem (TSP),
- Assembly Line Balancing (ALB), assigning tasks to workstations in a serial production system

Figure 1. Two-Sided Assembly Line



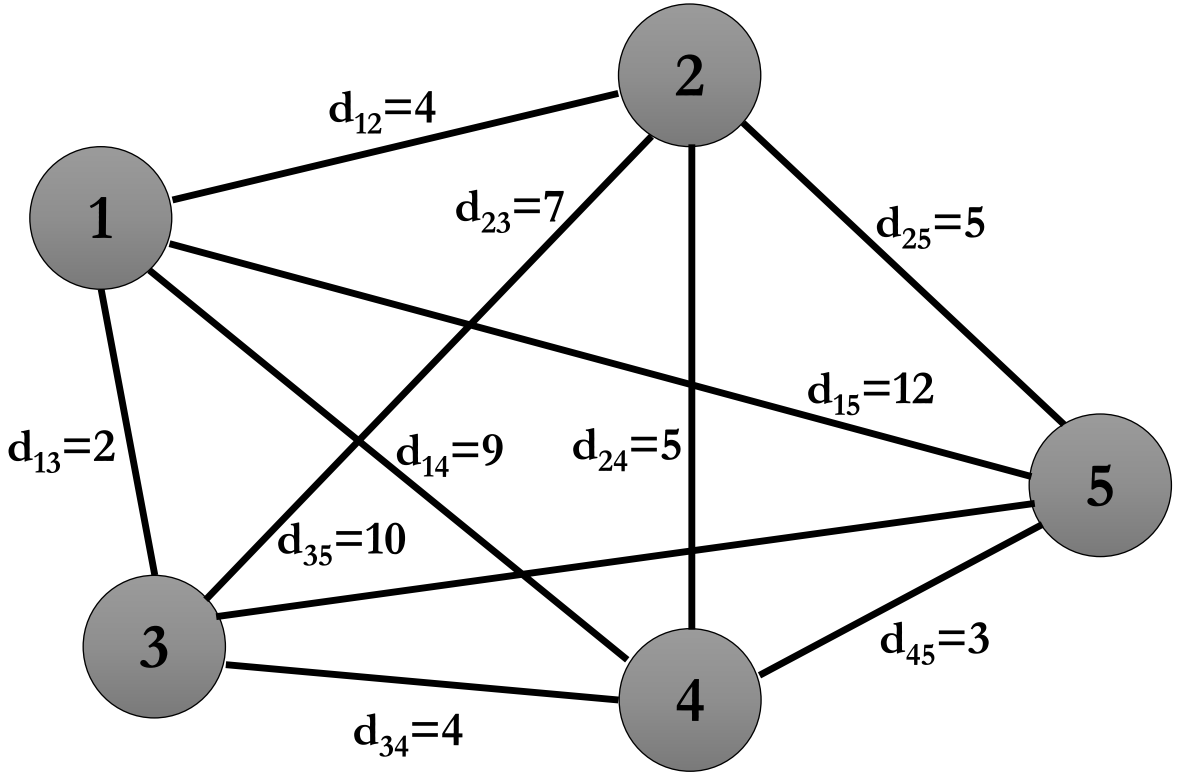
Source: Jurado y Taborda, 2006.

- Cell Assignment in PCS Networks (CA).
- Other

ACO Applications

Travelling Salesman Problem

ACO for TSP - Example



ACO for TSP - Example

- A small artificial pheromone value is placed on all the edges,
- ***M*** ants are placed on the graph divided among all the nodes,
- ***M*** (number of ants) roundtrips are created in ***n*** (number of cities) steps.

ACO Algorithm Overview

- Initialize the pheromone trails,
- While termination condition not met
  - ConstructAntSolutions,
  - ApplyLocalSearch *(Optional),*
  - UpdatePheromones,
- end

ACO for TSP - Example

- The ConstructAntSolutions step:

- For each ant:
  - A new node is randomly chosen according to the selection probability,

  - This node is added to the ants’ ***visited list*** (so as not to be re-selected because every city is visited once).

  - After the completion of the roundtrips, the visited lists are emptied,

ACO Algorithm Overview

- Initialize the pheromone trails,
- While termination condition not met
    - ConstructAntSolutions,
    - ApplyLocalSearch *(Optional),*
    - UpdatePheromones,
- end

ACO for TSP - Example

- Usually, the online step-by-step update rule is ***not adopted***,

- The ApplyLocalSearch step:
  - These are optional actions referred to as ***daemon actions***,

  - One action is to apply a local search method in order to improve the constructed solution,

  - Another action is to add extra pheromone (***delayed update***) based on the collection of some global information.

ACO Algorithm Overview

- Initialize the pheromone trails,
- While termination condition not met
  - ConstructAntSolutions,
  - ApplyLocalSearch *(Optional),*
  - UpdatePheromones,
- end

ACO for TSP - Example

- The UpdatePheromones step:
  - Apply pheromone evaporation,
  - Update the pheromone trails using a chosen set of good solutions, different approaches use (***Ant cycle model***):
    - All the solutions found in the current iteration,
    - The best solution found in the current iteration,
    - The best solution found so far,

ACO for TSP - Example

- A simple iteration:
  - Assume that an ant ***k*** was placed at node 1,
  - The neighbours of node 1 are :

$$N_1 = \{2, 3, 4, 5\}$$

  - Initially, all the edges have the same pheromone value (assume $\boldsymbol{\tau = 1}$).

ACO for TSP - Example

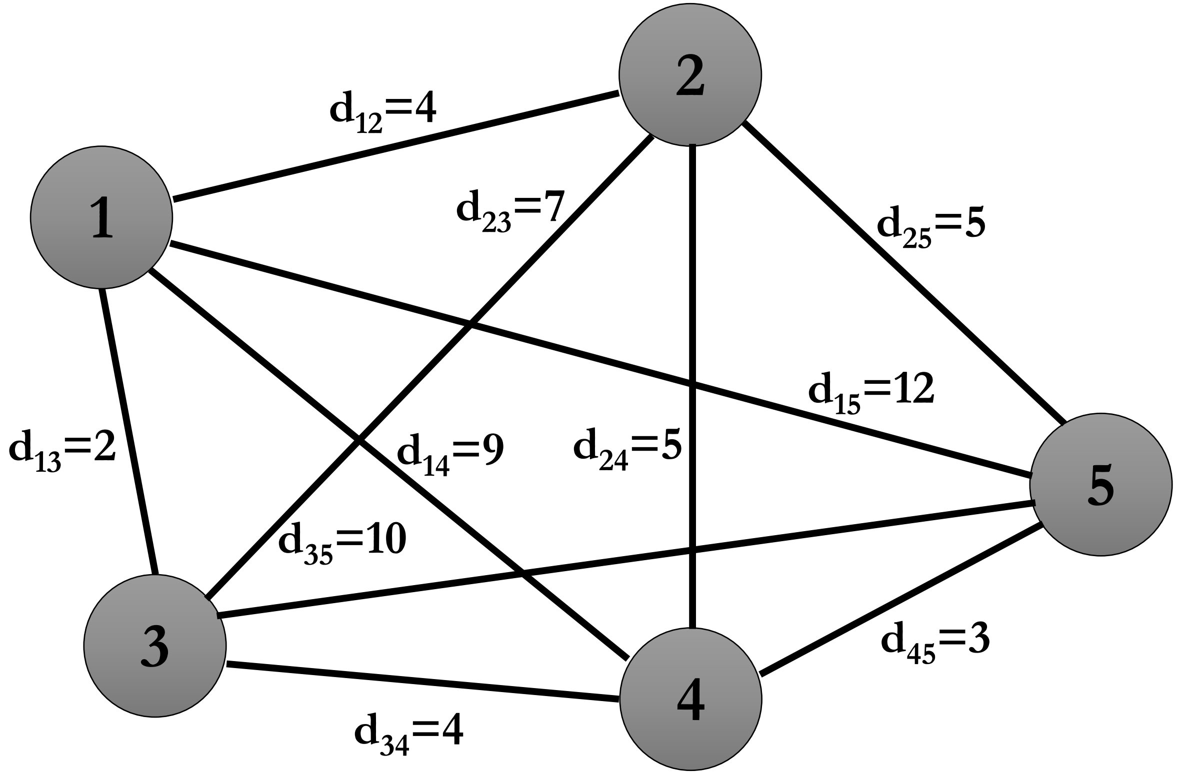
- Assuming α and β are both equal to 1,

- Then:

$$\sum_{n\in N_1} \tau_{1n}^{\alpha} / d_{1n}^{\beta} = \frac{1}{4} + \frac{1}{2} + \frac{1}{9} + \frac{1}{12} \cong 0.95$$

$$p_{12}^{k} = \frac{0.25}{0.95} \cong 0.26, p_{13}^{k} = \frac{0.5}{0.95} \cong 0.53$$

$$p_{14}^{k} = \frac{0.11}{0.95} \cong 0.12, p_{15}^{k} = \frac{0.08}{0.95} \cong 0.08$$



ACO for TSP - Example

- Node 3 has the highest probability of being selected,
- If node 3 gets selected, ant ***k*** moves to that node and continues with the next iteration where:

$$N_3 = \{2, 4, 5\}$$

ACO for TSP - Example

- Hence:

$$\sum_{n \in N_3} \tau_{1n}^{\alpha} / d_{1n}^{\beta} = \frac{1}{7} + \frac{1}{4} + \frac{1}{10} \cong 0.49$$

$$p_{32}^{k} = \frac{0.14}{0.49} \cong 0.29, p_{34}^{k} = \frac{0.25}{0.49} \cong 0.51$$

$$p_{35}^{k} = \frac{0.1}{0.49} \cong 0.20$$

- Another node is selected, and so on …

ACO for TSP - Example

- If ant *k* completes the tour:

$$Tour = \{1, 3, 2, 4, 5\}$$
$$Cost = 29$$

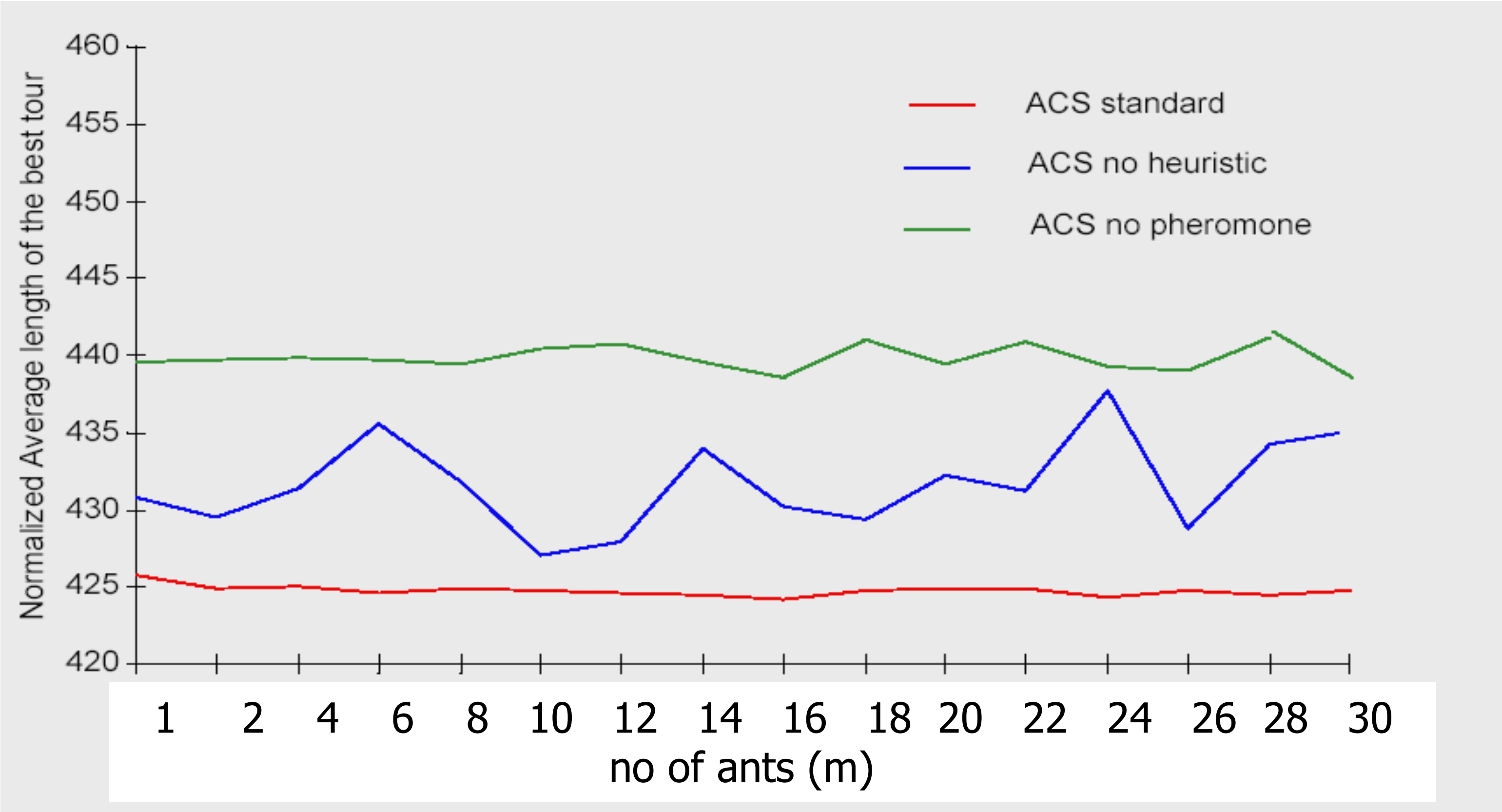
- First, all the pheromone trails get evaporated,

- And then, if this ant is selected to update the pheromone trails, it enforces the edges

$$\{1, 3\}, \{3, 2\}, \{2, 4\},$$
$$\{4, 5\}, \{5, 1\}$$

with the value Q/29.

Pheromone and the heuristic function



Comparison between ACS standard, ACS with no heuristic, and ACS in which ants neither sense nor deposit pheromone. Problem: Oliver30. Averaged over 30 trials

M. Dorigo and L. M. Gambardella."Ant Colony System: a cooperative learning approach to the traveling salesman problem". IEEE Trans. Evolutionary Computation, vol. 1, no. 1, 1997.

Pheromone and the heuristic function

- The results show that not using pheromone deteriorates the performance,

- ACS without heuristics performs better than ACS without pheromone. This may be due that ACS with pheromone and no heuristics is still guided by the global update rule (reflecting the importance of the solution).

- ACS without pheromone reduces to a stochastic multi-greedy algorithm

- ACS with both is better confirming the role of cooperation.

M. Dorigo and L. M. Gambardella."Ant Colony System: a cooperative learning approach to the traveling salesman problem". IEEE Trans. Evolutionary Computation, vol. 1, no. 1, 1997.

Comparison Results of ACO Systems

Number of ants = n
Number of iterations = 10000

Benchmark	Optimal	AS	MMAS	ACS
eil51	**426**	437.3	**427.6**	428.1
kroa100	**21282**	22471.4	**21320.3**	21420.0
d198	**15780**	16702.1	**15972.5**	16054.0

T. Stutzle and H. H. Hoos."*MAX-MIN* Ant System". *Future Generation Comput. Syst.*, vol. 16, no 8, 2000.

Comparison Results: ACO and Genetic Algorithms

- Comparison of ACS with the genetic algorithm (GA), evolutionary programming (EP), simulated annealing (SA), and the annealing-genetic algorithm (AG), a combination of genetic algorithm and simulated annealing.

- The best integer tour length, the best real tour length (in parentheses) and the number of tours required to find the best integer tour length (in square brackets).

- The best result for each problem is in boldface.

M. Dorigo and L. M. Gambardella."Ant Colonies for the travelling salesman problem", BioSystems, 1997.

Comparison Results: ACO and Genetic Algorithms

Problem name	ACS	GA	EP	SA	AG	Optimum
Oliver30 (30-city problem)	**420** **(423.74)** **[830]**	421 (N/A) [3,200]	**420** **(423.74)** [40,000]	424 (N/A) [24,617]	**420** (N/A) [12,620]	**420** **(423.74)**
Eil50 (50-city problem)	**425** (427.96) **[1,830]**	428 (N/A) [25,000]	426 **(427.86)** [100,000]	443 (N/A) [68,512]	436 (N/A) [28,111]	**425** (N/A)
Eil75 (75-city problem)	**535** **(542.31)** **[3,480]**	545 (N/A) [80,000]	**542** (549.18) [325,000]	580 (N/A) [173,250]	561 (N/A) [95,506]	**535** (N/A)
KroA100 (100-city problem)	**21,282** **(21,285.44)** **[4,820]**	21,761 (N/A) [103,000]	N/A (N/A) [N/A]	N/A (N/A) [N/A]	N/A (N/A) [N/A]	**21,282** (N/A)

It is clear that ACS and EP greatly outperform GA, SA, and AG.
N/A means not available in the literature

M. Dorigo and L. M. Gambardella.”Ant Colonies for the travelling salesman problem”, BioSystems, 1997.

ACS on larger TSP problems

ACS performance for some bigger geometric problems (over 15 trials). We report the integer length of the shortest tour found, the number of tours required to find it, the average integer length, the standard deviation, the optimal solution (for fl1577 we give, in square brackets, the known lower and upper bounds, given that the optimal solution is not known), and the relative error of ACS.

Problem name	ACS best integer length (1)	ACS number of tours generated to best	ACS average integer length	Standard deviation	Optimum (2)	Relative error $\frac{(1)-(2)}{(2)} * 100$
d198 (198-city problem)	15,888	585,000	16,054	71	15,780	0.68 %
pcb442 (442-city problem)	51,268	595,000	51,690	188	50,779	0.96 %
att532 (532-city problem)	28,147	830,658	28,523	275	27,686	1.67 %
rat783 (783-city problem)	9,015	991,276	9,066	28	8,806	2.37 %
fl1577 (1577-city problem)	22,977	942,000	23,163	116	[22,204 – 22,249]	3.27÷3.48 %

ACO Applications

Cell Assignment in PCS Networks

Cell Assignment in PCS Networks (CA)

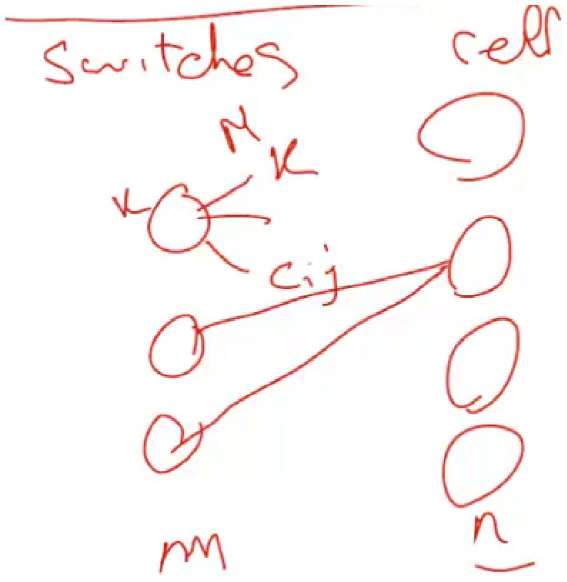
- The cell assignment problem is a challenging problem in PCS (Personal Communication Services) networks,
- In PCS networks, each cell has an antenna that is used to communicate with subscribers over some pre-assigned radio frequencies,
- Groups of cells are connected to a switch, through which the cells are then routed to the satellite networks.

S. J. Shyu, B. M. T. Lin and T. S. Hsiao.“An Ant Algorithm for Cell Assignment in PCS Netwroks”. IEEE International Conference on Networking, Sensing and Control, pp. 1081-1086, 2004.
J. R. L. Fournier and S. Pierre.“Assigning Cells to Switches in Mobile Networks using an Ant Colony Optimization Heuristic”.Computer Communications, vol. 28, pp. 65-73, 2005.

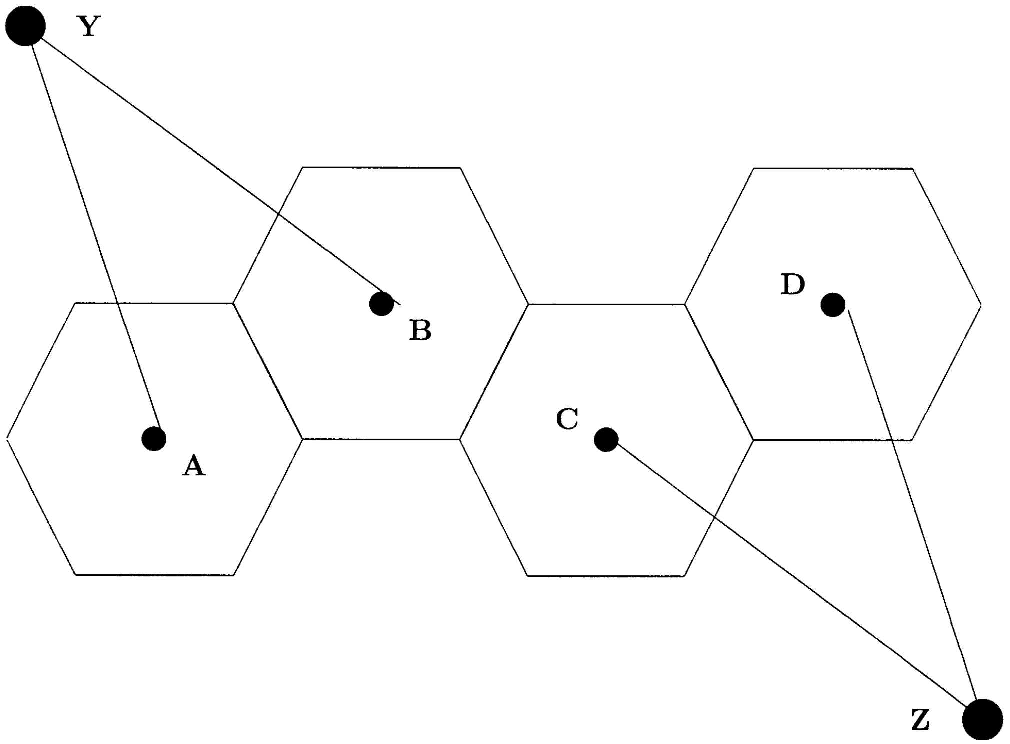
Cell Assignment in PCS Networks (CA)

- Assignment of channels to cells
  - Assign frequency channels to cells so as to minimize interference (close cells shouldn't have close frequency ranges) and maximize utilization of channels (reuse)

- Given a set of n cells to be assigned to m switches, each switch k has a capacity M_k. The objective is to have an assignment that minimizes the cost (link between cells and switches and handoff cost)



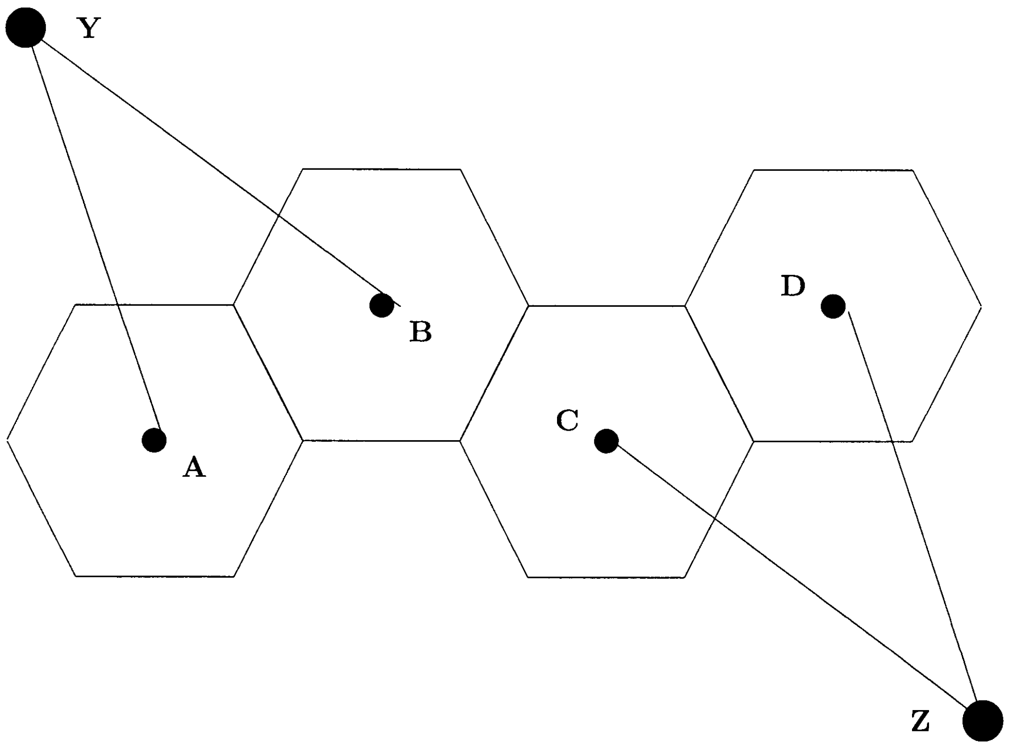
Cell Assignment in PCS Networks (CA)



For instance:

- Assuming a situation where cells **A** and **B** are assigned to switch **Y** and cells **C** and **D** are assigned to switch **Z**,
- Suppose that a subscriber is currently talking to someone and this call is transmitted through cell **B** and switch **Y**,

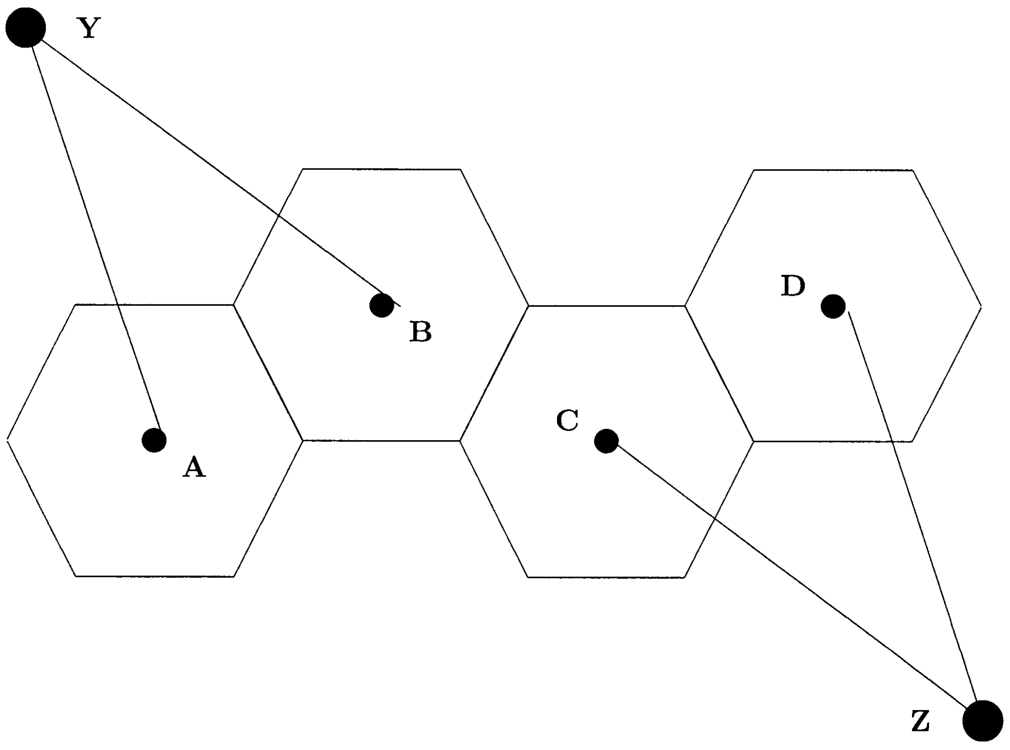
Cell Assignment in PCS Networks (CA)



If the subscriber moves from cell **B** to cell **A,** switch **Y** will perform a *handoff* for the call:

- This call does not trigger any location update in the database that records the position of the subscriber,
- The handoff does not entail any network entity other than switch **Y**.

Cell Assignment in PCS Networks (CA)



On the other hand, if the subscriber moves from cell **B** to cell **C**, then the *handoff* involves:

- The modification of the location of the subscriber in the database,
- The execution of a fairly complicated protocol between switches **Y** and **Z**.

Cell Assignment in PCS Networks (CA)

- Costs associated with the assignment include:
  - Cable (link) costs between cells and switches,
  - Handoff costs between different cells:
    - Simple or no costs (involving only one switch),
    - Complex costs (involving two switches).

- Each switch has a certain capacity (in terms of calls volume) that should not be exceeded.

- The objective is to have an assignment that minimizes the cost.

- NP-hard problem

Mathematical Formulation

- Let c_{ik} be the cost of the link between cell i and switch *k*, *i=1,…,n* ; *k=1,…,m*

- Let x_{ik} is 1 if cell i assigned to switch k, 0 otherwise;

- Let y_{ij} = 1 if cell i and cell j are assigned to the same switch.

- Let M_k be the capacity of switch k

- Let h_{ij} is handoff cost between cell i and j when they are assigned to different switches;

- Let d_i is the number of calls allowed for cell i

Mathematical Formulation

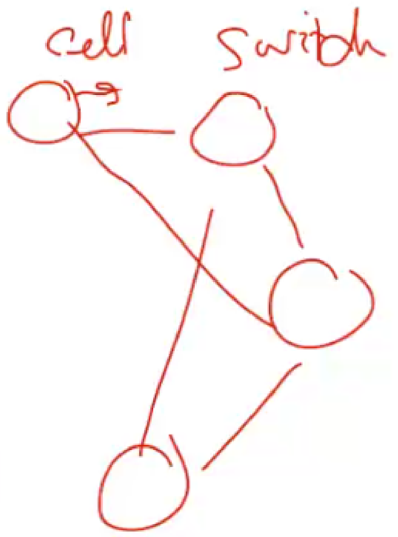
- The objective is

$$\text{Minimize} \quad f = \sum_{i=1}^{n}\sum_{k=1}^{m} c_{ik}x_{ik} + \sum_{i=1}^{n}\sum_{j=1, i\neq j}^{n} h_{ij}(1-y_{ij})$$

$$\text{subject to} \quad \sum_{i=1}^{n} d_i x_{ik} \leq M_k, k = 1,\ldots,m$$

ACO for CA

- In this problem, each ant has to make two different decisions per iteration:
    - Choosing the next cell to be assigned,
    - Choosing the switch that the chosen cell to be assigned to.
    - ➢A pheromone trail τ_{ij} is associated with every node i and j



cell – S – C – S – C – S

- Select max number of iterations
- Select number of ants
- Decide on transition rule
- Decide on pheromone update rules

ACO-algorithm

- Select max no of iterations, itmax
- Select number of Ants, antmax
- While number of iterations <itmax
  - While current ant number< antmax
    - Initialize ant parameters (pheromone, update,..etc)
    - While (number of assigned cells is less than n)
      - Select next node (cell/switch) to be assigned using transition rule considering the capacity constraint of the selected switch.
      - Update problem data (capacity, cost,..)
    - End
    - Update Pheromone trail using update rules and evaporation rule
    - Evaluate quality of the Solution using the objective function
  - Retain best solution of all ants
  - Update pheromone trail based on best solution (if desired)
- Return best solution

ACO for CA

- Transition rules

- Let T be the set of nodes not selected yet

  - Simple ratio

$$p_{ij} = \begin{cases} \dfrac{\tau_{ij}^{\alpha}}{\sum_{j \in T} \tau_{ij}^{\alpha}} & if \quad j \in T \\ 0 & if \quad j \notin T \end{cases}$$

  - Using heuristics

$$p_{ij} = \begin{cases} \dfrac{[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{l \in T} [\tau_{lk}]^{\alpha}[\eta_{lk}]^{\beta}} & \text{if } j \in \text{T} \\ 0 & \text{otherwise} \end{cases}$$

ACO for CA

- Heuristic can be related to utilization

$$\eta_{ij} = (1/\text{cost of assigning cell i to switch j})$$

- Pheromone update and evaporation can be simple

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

- Or

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

- Where $\Delta\tau_{ij}$ can be density, quantity (using local cost) or online delayed using solution quality

ACO for CA

- Or as in ACS, two updates

$$\tau_{ij} = (1 - \rho_2)\tau_{ij} + \rho_2 \tau_0$$

and

$$\tau_{ij} = (1 - \rho_1)\tau_{ij} + \rho_1 \Delta \tau_{ij}^{best}$$

References

Text: Fundamentals of Computational Swarm Intelligence by A. P. Engelbrecht, Wiely, 2005. Chapter 1, 22, and 23.

Papers:

- ID. Couzin, J. Krause, R. James, GD. Ruxton, NR. Franks.”Collective Memory and Spatial Sorting in Animal Groups”. Journal of Theoretical Biology, 218, pp. 1–11, 2002.

- C. Grosan, A. Abraham and C. Monica.”Swarm Intelligence in Data Mining”. In Swarm Intelligence in Data Mining, A. Abraham, C. Grosan and V. Ramos (Eds), Springer, pp. 1–16, 2006.

- MM. Millonas .“Swarms, phase transitions, and collective intelligence”. In CG. Langton Ed., Artificial Life III, Addison Wesley, Reading, MA, 1994.

References

- P. P. Grasse.“Recherches sur la biologie des termites champignonnistes (*Macrotermitina*)”. Ann. Sc. Nat., Zool. Biol. anim., 6, 97, 1944.

- P. P. Grasse.”La reconstruction du nid et les coordinations interindividu- elles chez *bellicositermes natalensis* et *cubitermes* sp. La theorie de la stig- mergie: essai d'interpretation du comportement des termites constructeurs”. Insectes Sociaux, 6, 41, 1959.

- J. L. Deneubourg, S. Aron, S. Goss and J. M. Pasteels.”The self- organizing exploratory pattern of the Argentine ant”. Journal of Insect Behaviour, 3, 159, 1990.

- S. Goss, S. Aron, J. L. Deneubourg and J. M. Pasteels.”Self-organized shortcuts in the Argentine ant ”. Naturwissenschaften, 76, 579, 1989.

- M. Dorigo.”Optimization, Learning and Natural Algorithms”. Ph.D. thesis, DEI, Politecnico di Milano, Italy, pp. 140, 1992.

References

- M. Dorigo, V. Maniezzo and A. Colorni.” Ant System: Optimization by a Colony of Cooperating Agents”. IEEE Trans. Syst., Man, and Cybern. Part B, vol. 26, no. 1, 1996.

- T. Stutzle and H. H. Hoos.”*MAX-MIN* Ant System”. *Future Generation Comput. Syst.*, vol. 16, no 8, 2000.

- M. Dorigo and L. M. Gambardella.”Ant Colony System: a cooperative learning approach to the traveling salesman problem”. IEEE Trans. Evolutionary Computation, vol. 1, no. 1, 1997.

- J. Bautista and J. Pereira. “Ant Algorithms for A Time and Space Constrained Assembly Line Balancing Problem”. European Journal of Operation Research, 177, pp. 2016-2032, 2002.

- S. J. Shyu, B. M. T. Lin and T. S. Hsiao.“An Ant Algorithm for Cell Assignment in PCS Netwroks”. IEEE International Conference on Networking, Sensing and Control, pp. 1081-1086, 2004.