# Ant Colony Optimization

**Christian Blum**

Albcom, Lsi,
Universitat Politècnica De Catalunya
Barcelona, Spain
cblum@lsi.upc.edu

© Alex Wild (http://www.myrmecos.net)


*GECCO'11*, July 12–16, 2011, Dublin, Ireland.
ACM 978-1-4503-0690-4/11/07.

# Swarm Intelligence

## Short introduction and examples

# What is swarm intelligence

In a nutshell: AI discipline whose goal is designing intelligent multi-agent systems by taking inspiration from the collective behaviour of animal societies such as ant colonies, flocks of birds, or fish schools

© Ralf Müller

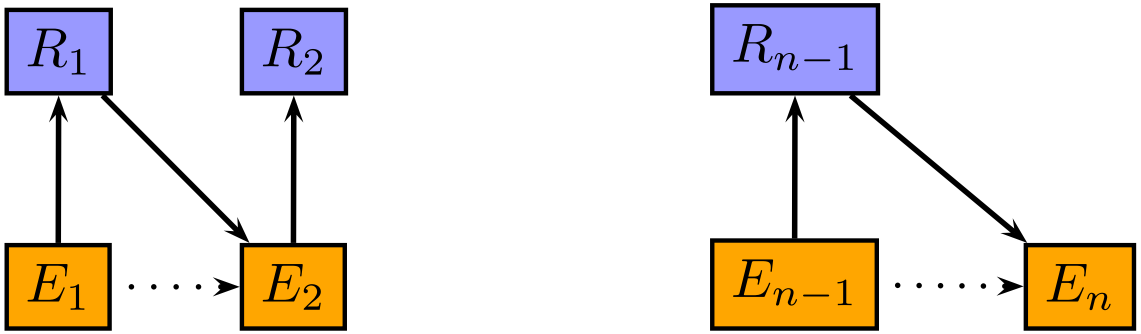
By curtesy of www.aerospaceweb.org

© Su Neko

# Swarm intelligence

Properties:

- Consist of a set of simple entities
- Distributedness: No global control
- Self-organization by:
  - **Direct communication:** visual, or chemical contact
  - **Indirect communication:** Stigmergy (Grassé, 1959)




Result: Complex tasks/behaviors can be accomplished/exhibited in cooperation

# Swarm intelligence

**Examples of social insects:**

- Ants
- Termites
- Some wasps and bees

© Velo Steve

© Alex Wild (http://www.myrmecos.net)

© Alex Wild

# Ant Colony Optimization

## A metaheuristics for optimization

# Inspiration of ACO (1)

**Communication strategies:**

- Direct communication: For example, recruitment
- Indirect communication: via chemical pheromone trails

© Alex Wild (http://www.myrmecos.net)

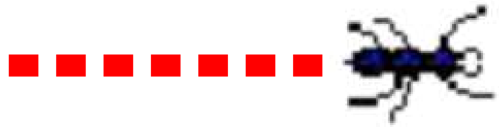
© Christian Blum

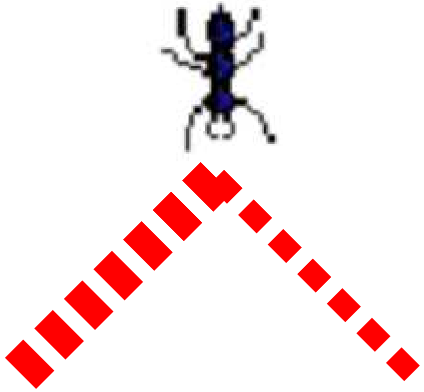
# Inspiration of ACO (2)

**Communication strategies:**

- Direct communication: For example, recruitment
- Indirect communication: via chemical pheromone trails

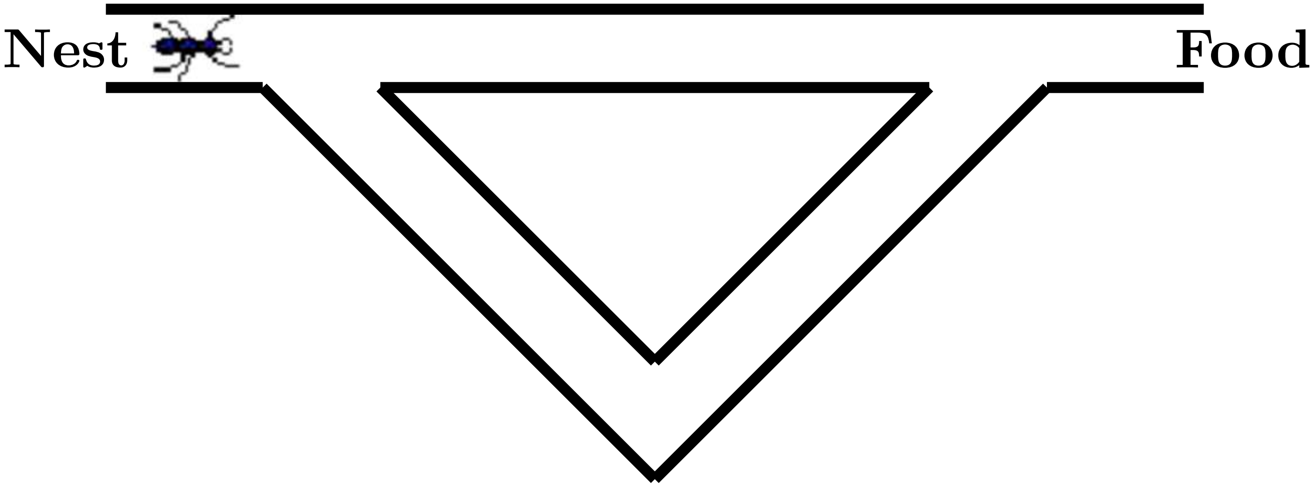
**Basic behaviour:**

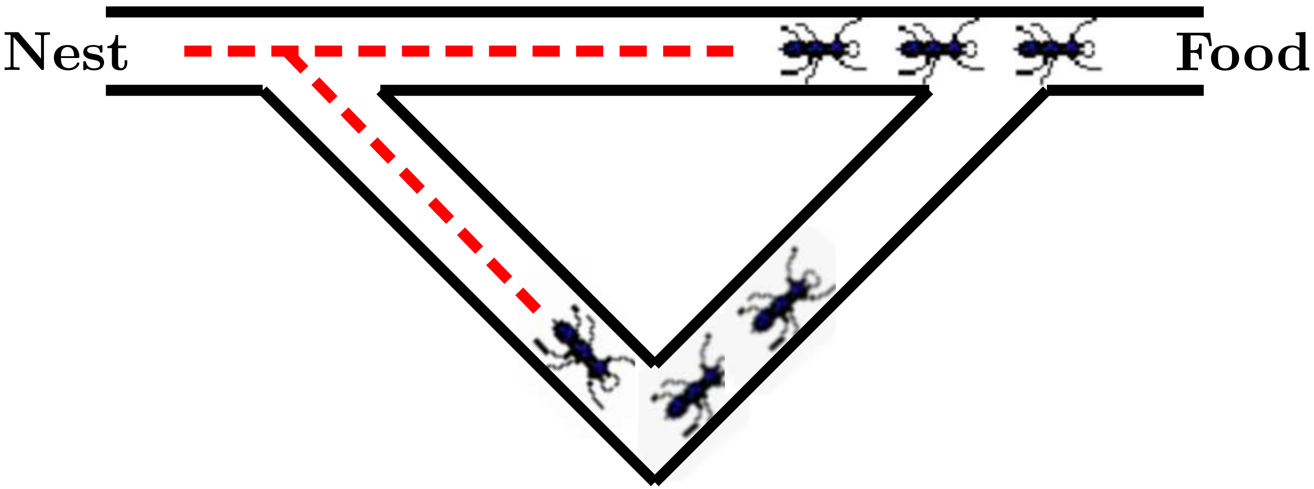




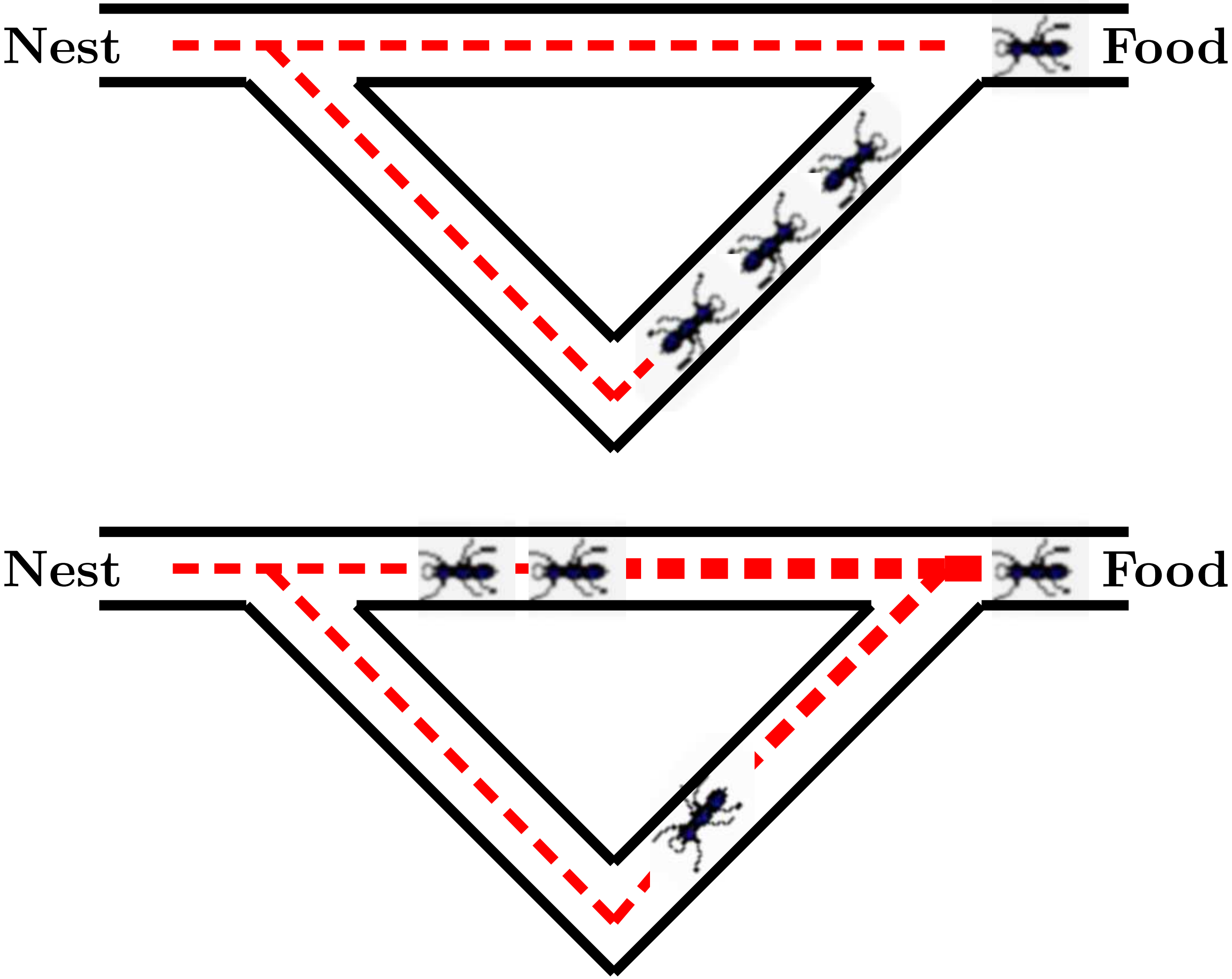
# Inspiration of ACO (3)

# Inspiration of ACO: double-bridge experiment (1)



# Inspiration of ACO: double-bridge experiment (2)

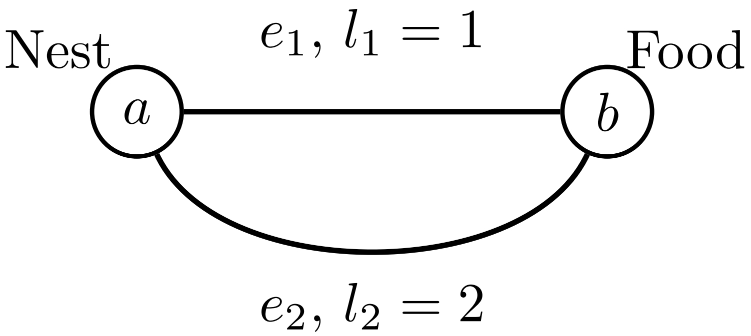


# The ant colony optimization metaheuristic

- Simulation of the foraging behaviour
- The ACO metaheuristic
- Example: traveling salesman problem (TSP)
- A closer look at algorithm components

# Simulation of the foraging behaviour (1)

Technical simulation:




1. We introduce artificial pheromone parameters:

$$\mathcal{T}_1 \text{ for } e_1 \text{ and } \mathcal{T}_2 \text{ for } e_2$$

2. W initialize the phermomone values:

$$\tau_1 = \tau_2 = c > 0$$

# Simulation of the foraging behaviour (2)

Algorithm:

**Iterate:**

1. Place $n_a$ ants in node $a$.
2. Each of the $n_a$ ants traverses from $a$ to $b$ either
   - via $e_1$ with probability $\mathbf{p}_1 = \frac{\tau_1}{\tau_1 + \tau_2}$,
   - or via $e_2$ with probability $\mathbf{p}_2 = 1 - \mathbf{p}_1$.
3. Evaporate the artificial pheromone: $i = 1, 2$

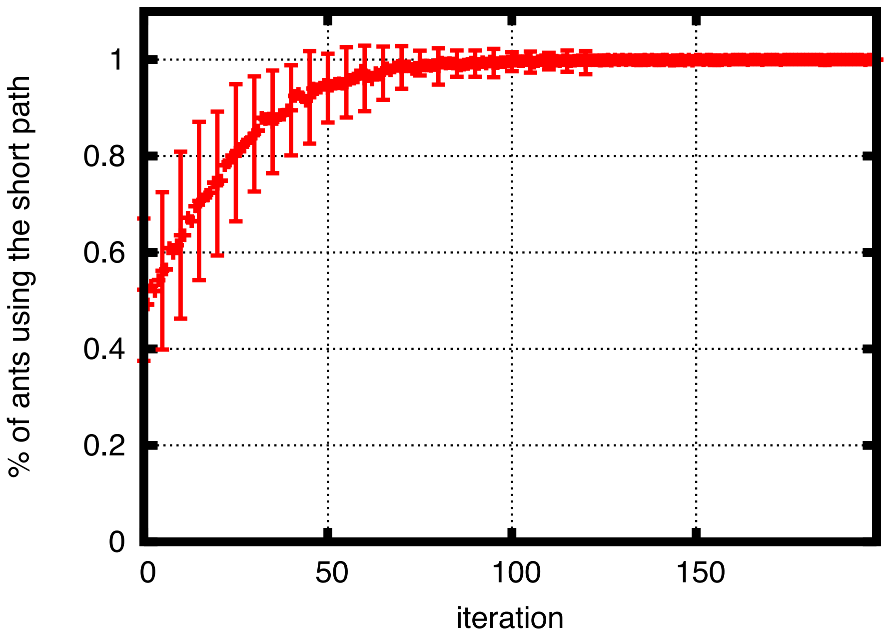
$$\tau_i \leftarrow (1 - \rho)\tau_i \;, \; \rho \in (0, 1]$$

4. Each ant leaves pheromone on its traversed edge $e_i$:

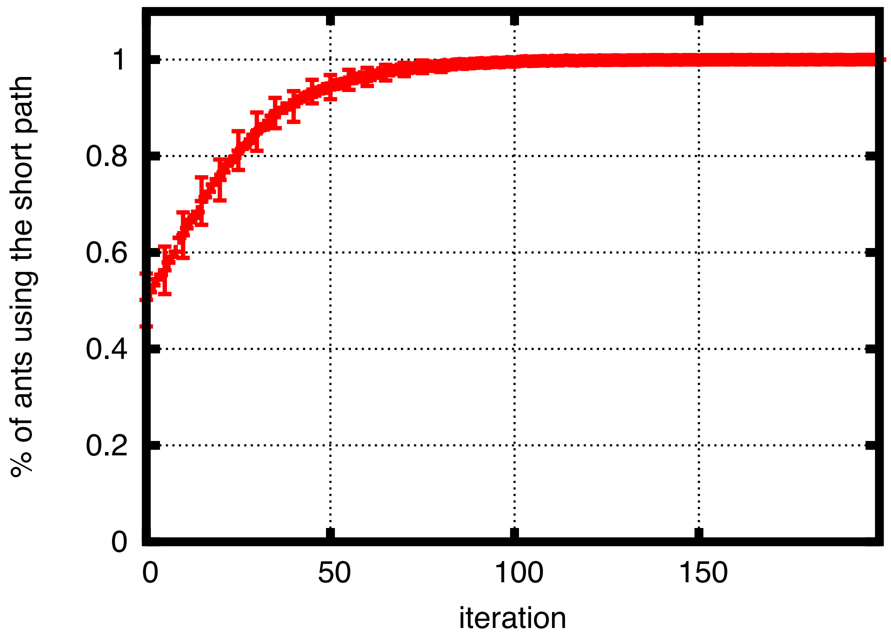
$$\tau_i \leftarrow \tau_i + \frac{1}{l_i}$$

# Simulation of the foraging behaviour (3)

**Simulation results:**




Colony size: 10 ants




Colony size 100 ants

**Observation:** Optimization capability is due to co-operation

# Simulation of the foraging behaviour (4)

Main differences between model and reality:

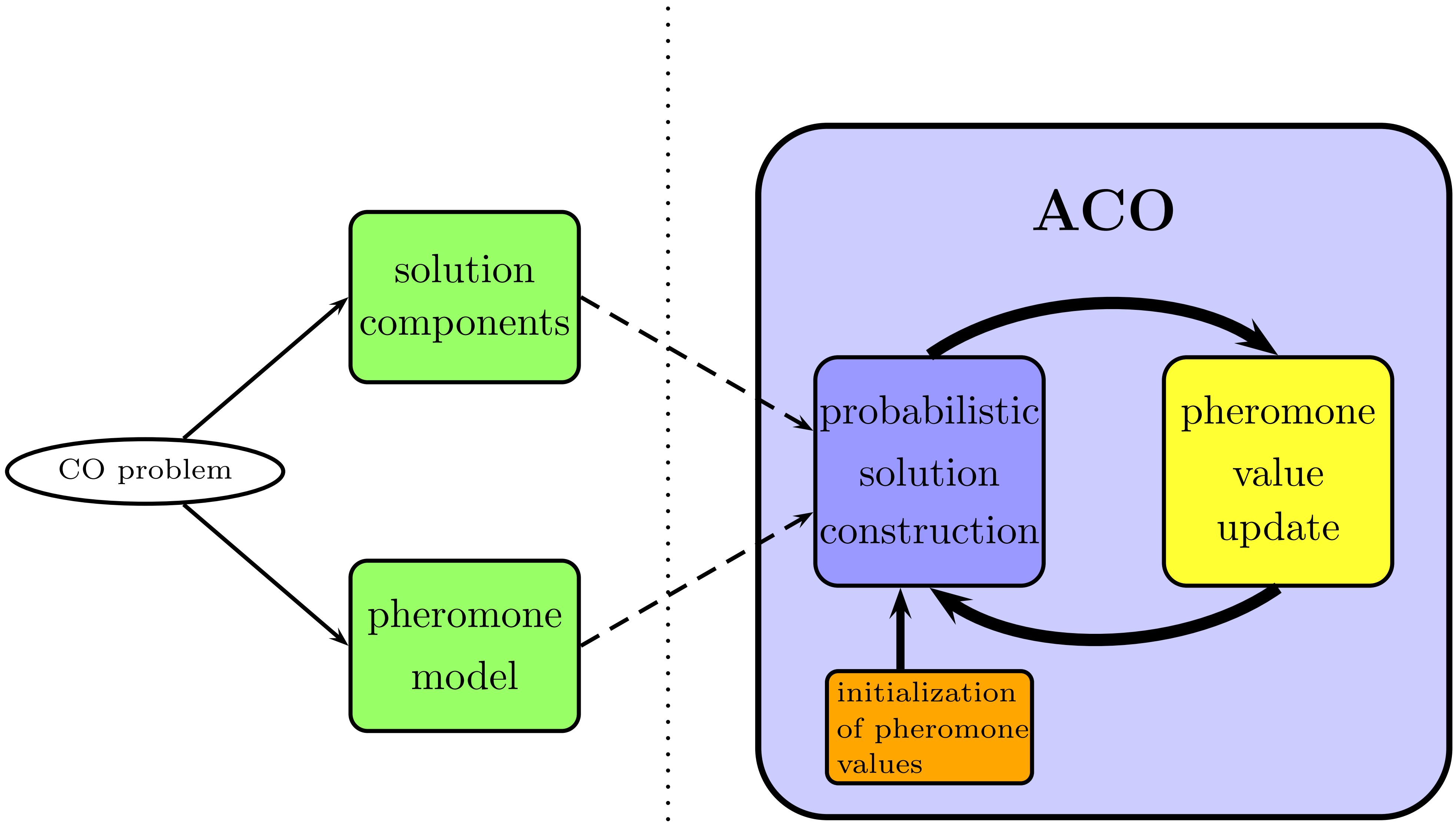
| | **Real ants** | **Simulated ants** |
|---|---|---|
| **Ants' movement** | asynchronous | synchronized |
| **Pheromone laying** | while moving | after the trip |
| **Solution evaluation** | implicitly | explicit quality measure |

Problem: In combinatorial optimization we want to find good solutions

# The ant colony optimization metaheuristic

- Simulation of the foraging behaviour
- The ACO metaheuristic
- Example: traveling salesman problem (TSP)
- A closer look at algorithm components

# The ACO framework



# The ACO pseudocode

**input:** An instance $P$ of a combinatorial problem $\mathcal{P}$.
InitializePheromoneValues($\mathcal{T}$)
**while** termination conditions not met **do**
  $S_{iter} \leftarrow \emptyset$
  **for** $j = 1, \ldots, n_a$ **do**
    $s \leftarrow$ ConstructSolution($\mathcal{T}$)
    $s \leftarrow$ LocalSearch($s$) — optional —
    $S_{iter} \leftarrow S_{iter} \cup \{s\}$
  **end for**
  ApplyPheromoneUpdate($\mathcal{T}$)
**end while**
**output:** The best solution found

# Metaheuristics: Timeline of their introduction

Metaheuristics:

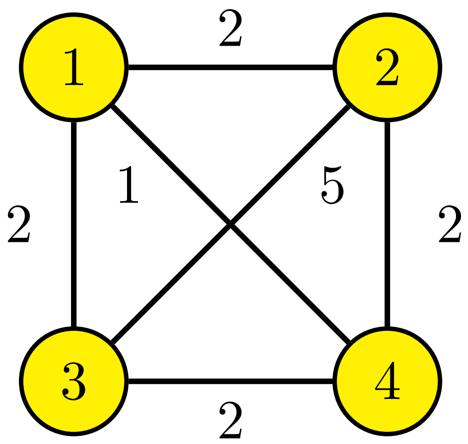
- Simulated Annealing (SA) [Kirkpatrick, 1983]
- Tabu Search (TS) [Glover, 1986]
- Genetic and Evolutionary Computation (EC) [Goldberg, 1989]
- **Ant Colony Optimization (ACO) [Dorigo, 1992]**
- Greedy Randomized Adaptive Search Procedure (GRASP) [Resende, 1995]
- Particle Swarm Optimization (PSO) [Kennedy, 1995]
- Guided Local Search (GLS) [Voudouris, 1997]
- Iterated Local Search (ILS) [Stützle, 1999]
- Variable Neighborhood Search (VNS) [Mladenović, 1999]

# The ant colony optimization metaheuristic

- Simulation of the foraging behaviour
- The ACO metaheuristic
- Example: traveling salesman problem (TSP)
- A closer look at algorithm components

# TSP: definition (1)

Example: Traveling salesman problem (TSP). Given a completely connected, undirected graph $G = (V, E)$ with edge-weights.

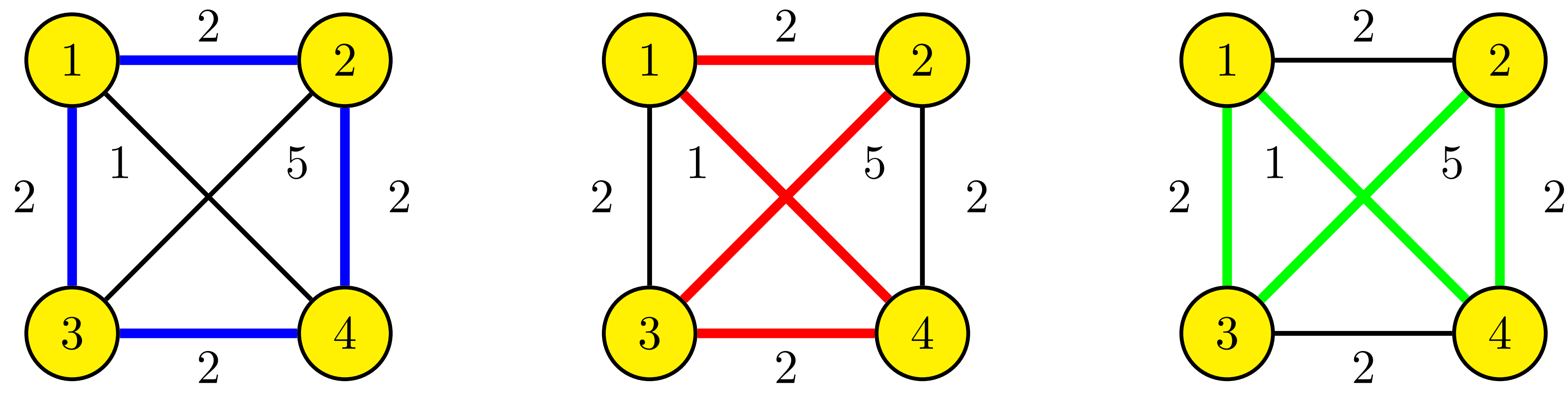



Goal: Find a tour (a Hamiltonian cycle) in $G$ with minimal sum of edge weights.

# TSP definition (2)

TSP in terms of a combinatorial optimization problem $\mathcal{P} = (\mathcal{S}, f)$:

- $\mathcal{S}$ consists of all possible Hamiltonian cycles in $G$.
- Objetive function $f : \mathcal{S} \mapsto \mathbb{R}^+$: $s \in \mathcal{S}$ is defined as the sum of the edge-weights of the edges that are in $s$.



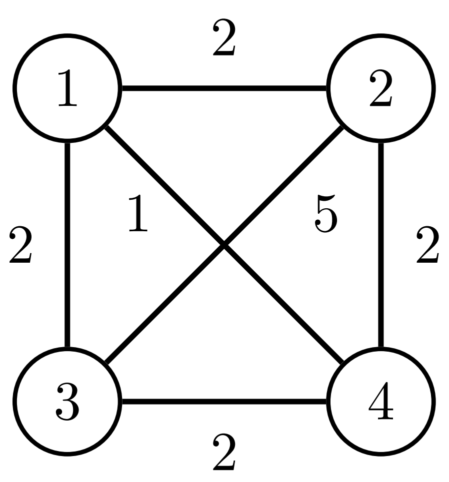

obj. function value: 8    obj. function value: 10    obj. function value: 10

# Applying ACO to the TSP

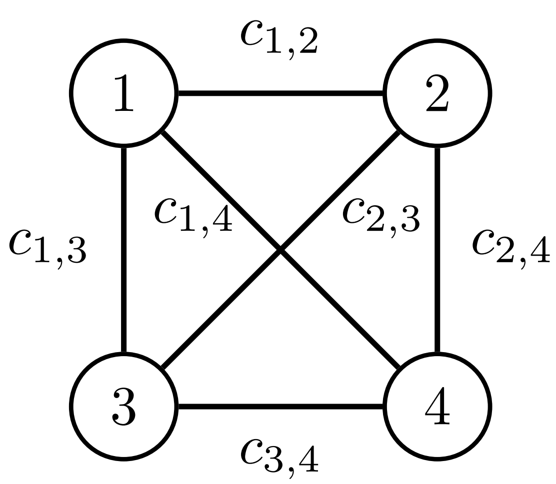
**Preliminary step:** Definition of the

- solution components
- pheromone model

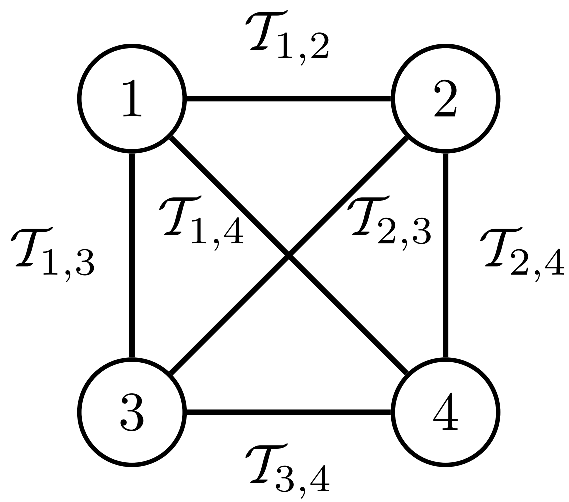
**example instance**




**solution components**



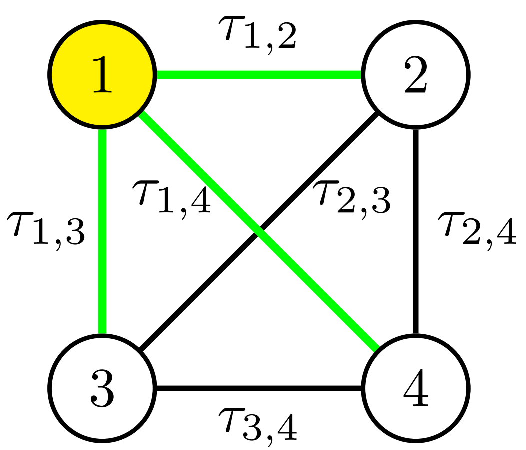

**pheromone model**



# TSP: solution construction

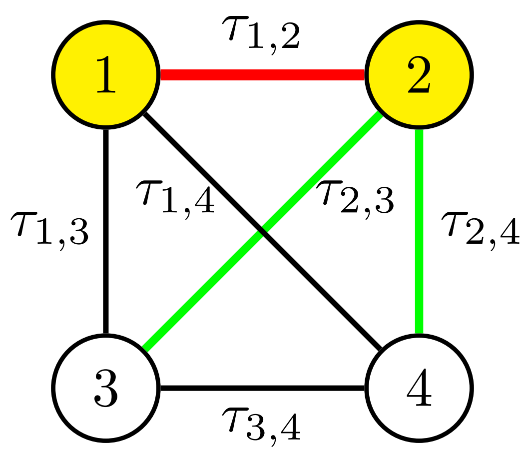
**Tour construction:**

**Step 1**



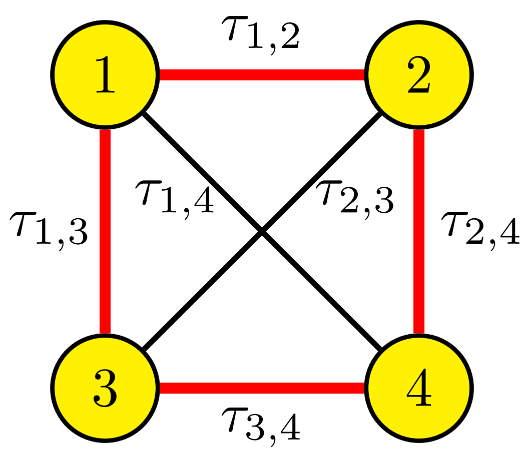

$$\mathbf{p}(c_{i,j}) = \frac{\tau_{i,j}}{\tau_{1,2} + \tau_{1,3} + \tau_{1,4}}$$

**Step 2**




$$\mathbf{p}(c_{i,j}) = \frac{\tau_{i,j}}{\tau_{2,3} + \tau_{2,4}}$$

**Finished**



# TSP: pheromone update (1)

Pheromone update: For example with the Ant System (AS) update rule

**Pheromone evaporation**

$$\tau_{i,j} \leftarrow (1 - \rho) \cdot \tau_{i,j}$$

**Reinforcement**

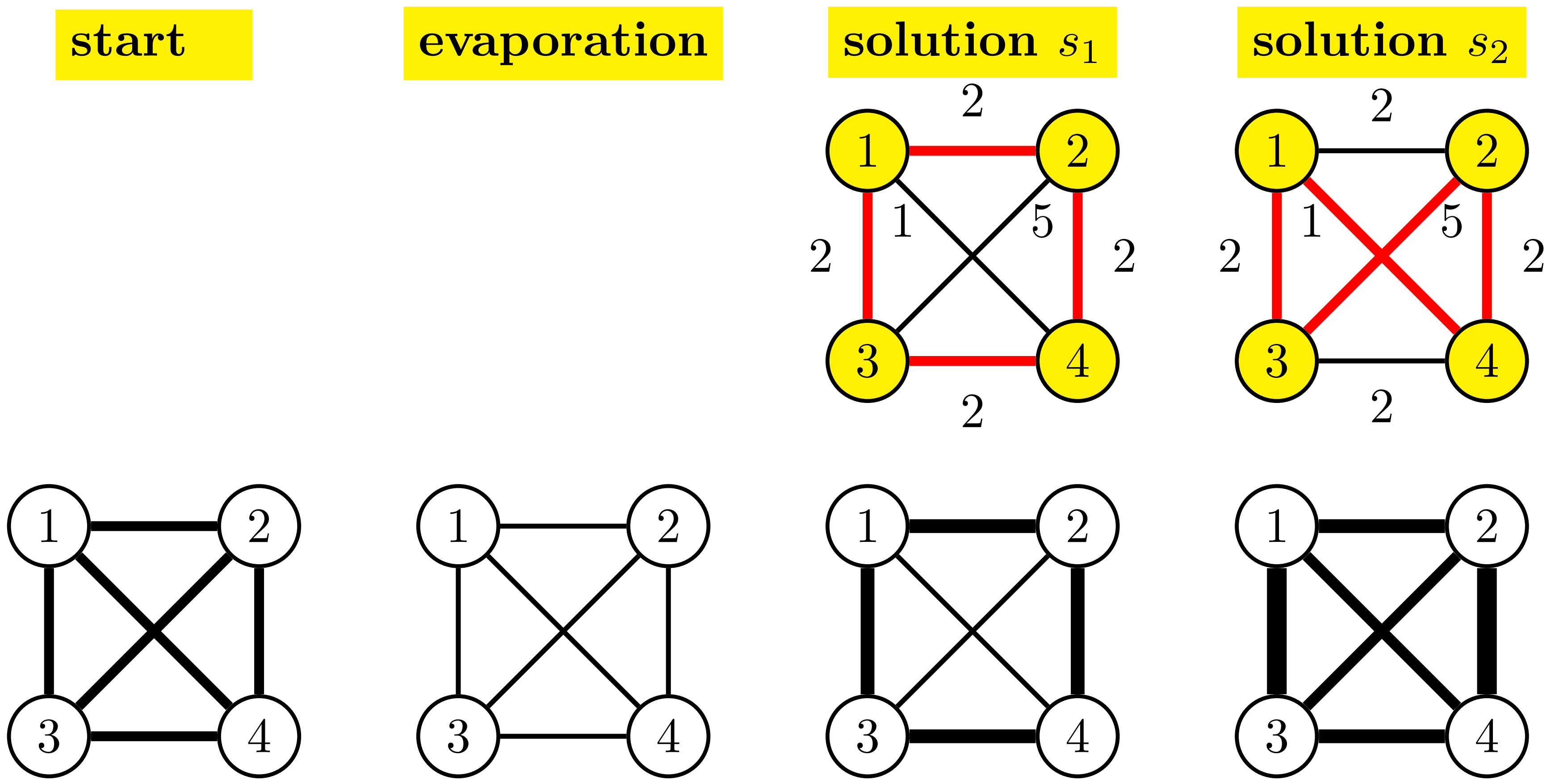
$$\tau_{i,j} \leftarrow \tau_{i,j} + \rho \cdot \sum_{\{s \in S_{iter} | c_{i,j} \in s\}} F(s)$$

where

- evaporation rate $\rho \in (0, 1]$
- $S_{iter}$ is the set of solutions generated in the current iteration
- quality function $F : S \mapsto \mathbb{R}^+$. We use $F(\cdot) = \frac{1}{f(\cdot)}$

# TSP: pheromone update (2)

Pheromone update: For example with the Ant System (AS) update rule

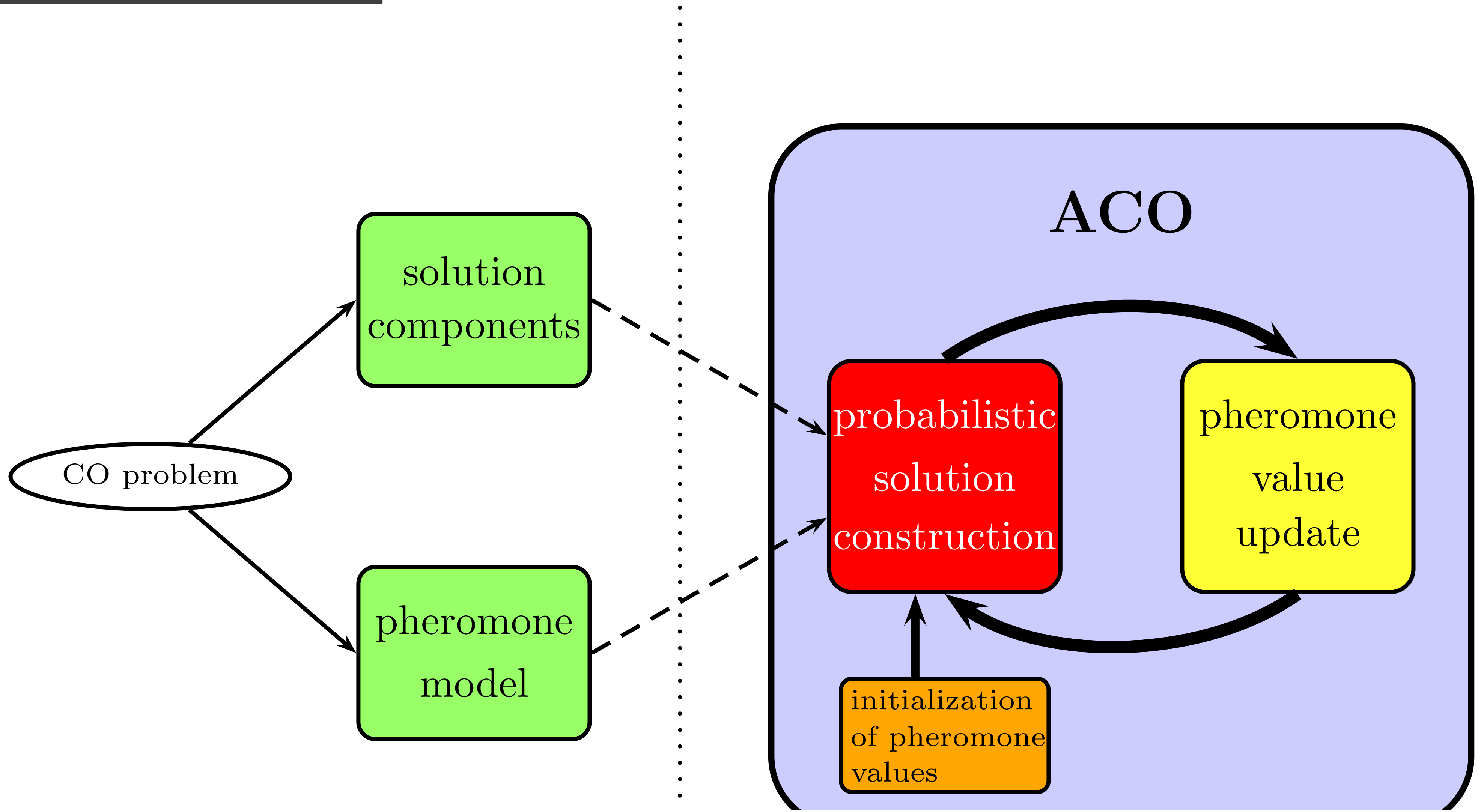


# The ant colony optimization metaheuristic

- Simulation of the foraging behaviour
- The ACO metaheuristic
- Example: traveling salesman problem (TSP)
- A closer look at algorithm components

# Solution construction (1)

Solution construction: A closer look



# Solution construction (2)

**A general constructive heuristic:**

- $s^p = \langle\rangle$
- Determine $N(s^p)$
- **while** $N(s^p) \neq \emptyset$
  - $c \leftarrow \mathsf{ChooseFrom}(N(s^p))$
  - $s^p \leftarrow$ extend $s^p$ by adding solution component $c$
  - Determine $N(s^p)$
- **end while**

**Problem:** How to implement function $\mathsf{ChooseFrom}(N(s^p))$?

# Solution construction (3)

Possibilities for implementing ChooseFrom($N(s^p)$):

- Greedy algorithms:

$$c^* = \mathrm{argmax}_{c_{i,j} \in N(s^p)} \eta(c_{i,j}) \ ,$$

where $\eta : C \mapsto I\!R^+$ is a Greedy function

Examples for Greedy functions:

- TSP: Inverse distance between nodes (i.e., cities)
- SALB: $t_i/C$

# Solution construction (4)

Possibilities for implementing ChooseFrom($N(s^p)$):

- Ant colony optimization:

$$\mathbf{p}(c_{i,j} \mid s^p) = \frac{[\tau_{i,j}]^{\alpha} \cdot [\eta(c_{i,j})]^{\beta}}{\sum_{c_{k,l} \in N(s^p)} [\tau_{k,l}]^{\alpha} \cdot [\eta(c_{k,l})]^{\beta}} \ , \quad \forall \ c_{i,j} \in N(s^p) \ ,$$

where $\alpha$ and $\beta$ are positive values

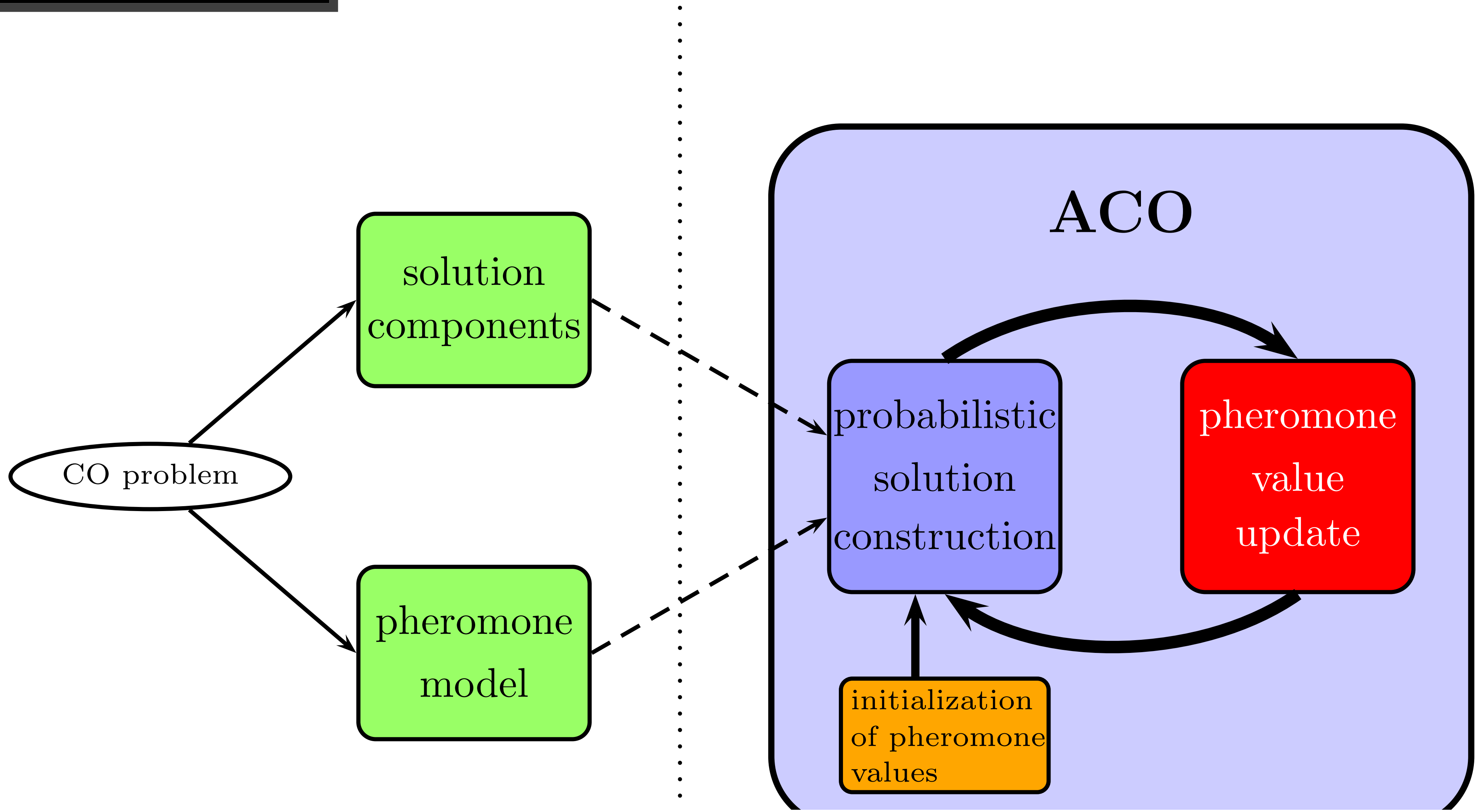
Note: $\alpha$ and $\beta$ balance between pheromone information and Greedy function

Observations:

- ACO can be applied if a constructive heuristic exists!
- ACO can be seen as an iterative, adaptive Greedy algorithm

# Pheromone update (1)

Pheromone update: A closer look



# Pheromone update (2)

A general update rule:

$$\tau_{i,j} \leftarrow (1-\rho)\cdot\tau_{i,j} + \rho \cdot \sum_{\{s\in S_{upd} | c_{i,j}\in s\}} w_s \cdot F(s) \quad ,$$

where

- evaporation rate $\rho \in (0, 1]$
- $S_{upd}$ is the set of solutions used for the update
- quality function $F : S \mapsto \mathbb{R}^+$. We use $F(\cdot) = \frac{1}{f(\cdot)}$
- $w_s$ is the weight of solution $s$

Question: Which solutions should be used for updating?

# Pheromone update (3)

**ACO update variants:**

| | |
|---|---|
| AS-update | $S_{upd} \leftarrow S_{iter}$<br>weights: $w_s = 1 \; \forall \; s \in S_{upd}$ |
| elitist AS-update | $S_{upd} \leftarrow S_{iter} \cup \{s_{bs}\}$ ($s_{bs}$ is best found solution)<br>weights: $w_s = 1 \; \forall \; s \in S_{iter}$, $w_{s_{bs}} = e \geq 1$ |
| rank-based AS-update | $S_{upd} \leftarrow$ best $m-1$ solutions of $S_{iter} \cup \{s_{bs}\}$ (ranked)<br>weights: $w_s = m - r$ for solutions from $S_{iter}$, $w_{s_{bs}} = m$ |
| IB-update: | $S_{upd} \leftarrow \text{argmax}\{F(s) \mid s \in S_{iter}\}$<br>weight 1 |
| BS-update: | $S_{upd} \leftarrow \{s_{bs}\}$<br>weight 1 |

# Successful ACO variants

- Ant Colony System(ACS) [Dorigo, Gambardella, 1997]

  M. Dorigo and L. M. Gambardella. **Ant colony system: a cooperative learning approach to the traveling salesman problem**. *IEEE Trans. Evolutionary Computation*, 1(1), 53–66, 1997

- $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System($\mathcal{MM}$AS) [Stützle, Hoos, 2000]

  T. Stützle and H. H. Hoos. **MAX-MIN Ant System**. *Future Generation Computer Systems*, 16(8), 889–914, 2000

- The hyper-cube framework (HCF) for ACO [Blum, Dorigo, 2004]

  C. Blum and M. Dorigo. **The hyper-cube framework for ant colony optimization**. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 34(2), 1161–1172, 2004

- Population-based ACO (P-ACO) [Guntsch, Middendorf, 2002]

  M. Guntsch and M. Middendorf. **A population based approach for ACO**. In: *Proceedings of EvoWorkshops 2002*, Springer LNCS, pages 71–80, 2002