ISD Programme 2010:

Artificial intelligence 2

Lecture 9, 2010-12-07

Ant colony optimization (ACO)

Krister Wolff, PhD, senior researcher, Chalmers University of Technology
e-mail: krister.wolff@chalmers.se url: www.am.chalmers.se/~wolff

Ant colony optimization

- Contents: **MW**, pp. 99-115.
  Appendix B.3.1, B.3.2

“Algorithms inspired by the behaviors of real ants”

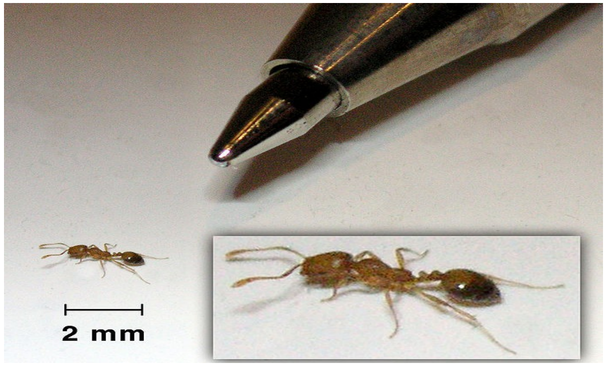
- Biological background
- Ant algorithms
- Applications of ACO

Biological background

- **Ants** are among the most widespread living organisms on our planet:

  They have colonized *almost every landmass on the planet,* except Antarctica, Greenland, Iceland and a few larger islands.

  More than **12000 species**(!) are known.



Pharaoh ants

Leaf-cutter ants

Biological background

- **Ants** display an amazing ability to *co-operate* in order to achieve goals far beyond the reach of an individual ant:


A group of Weaver ants bringing leaves together, using their own bodies as a dynamic bridge.

- http://www.youtube.com/watch?v=n71abhaadRs
- http://www.youtube.com/watch?v=IBTjQMtbViU

Biological background

- **Ants** can do collective transport of heavy objects.

  Coordination in collective transport of food objects is more efficient than single ants moving the pieces:

Ants doing collective transport

Biological background

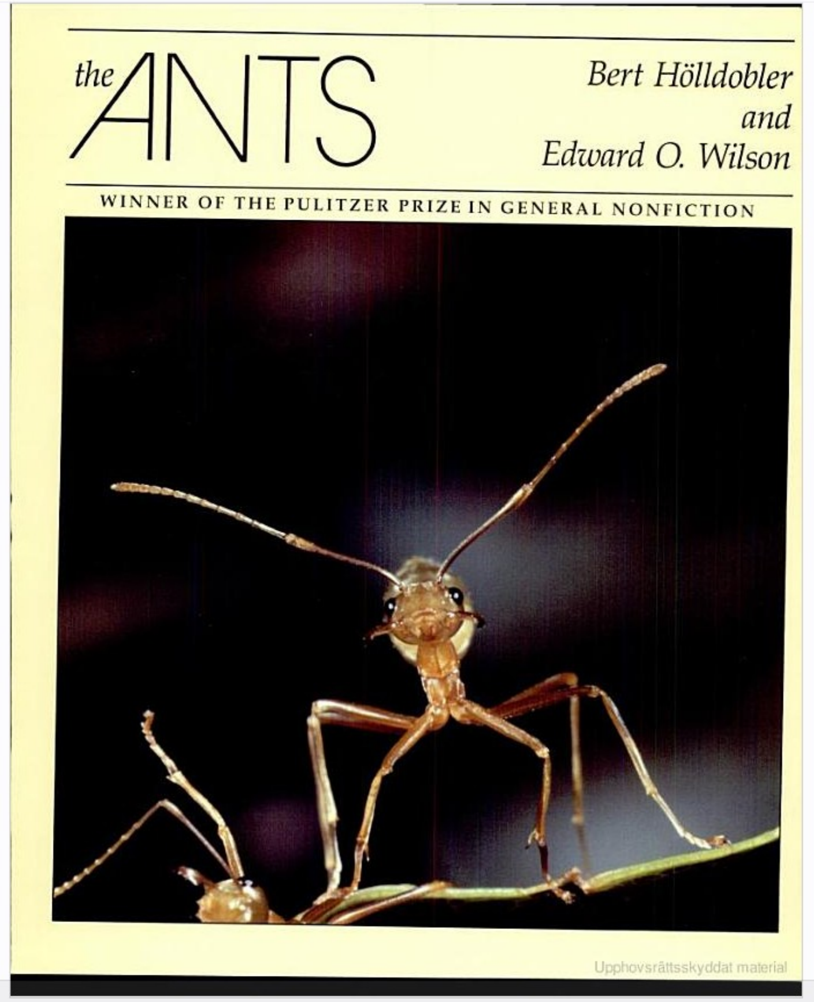
- Further reading:

  The ants

  by:

  Bert Hölldobler, and

  Edward O.Wilson



Biological background

- **Collective transport** of heavy objects is a complex task:

  (1) Discover the object.

  (2) Realize that it is too heavy for a single ant.

  (3) Do recruitment of more ants.

  (4) Coordinate the transport.

  (5) Overcome dead locks.

  Note:

  There is no specific leader.

  Only short-range communication (*sound, pheromones*).

Biological background

- **Pheromones:** Chemical substances that ants deposit on the ground, which can be perceived by other ants.
- **Stigmergy:** *Indirect* communication through modification of the local environment, by e.g. deposit of pheromones.

  (1) Foraging ants deposit a trail of pheromones.

  (2) More ants are then likely to follow the same path.

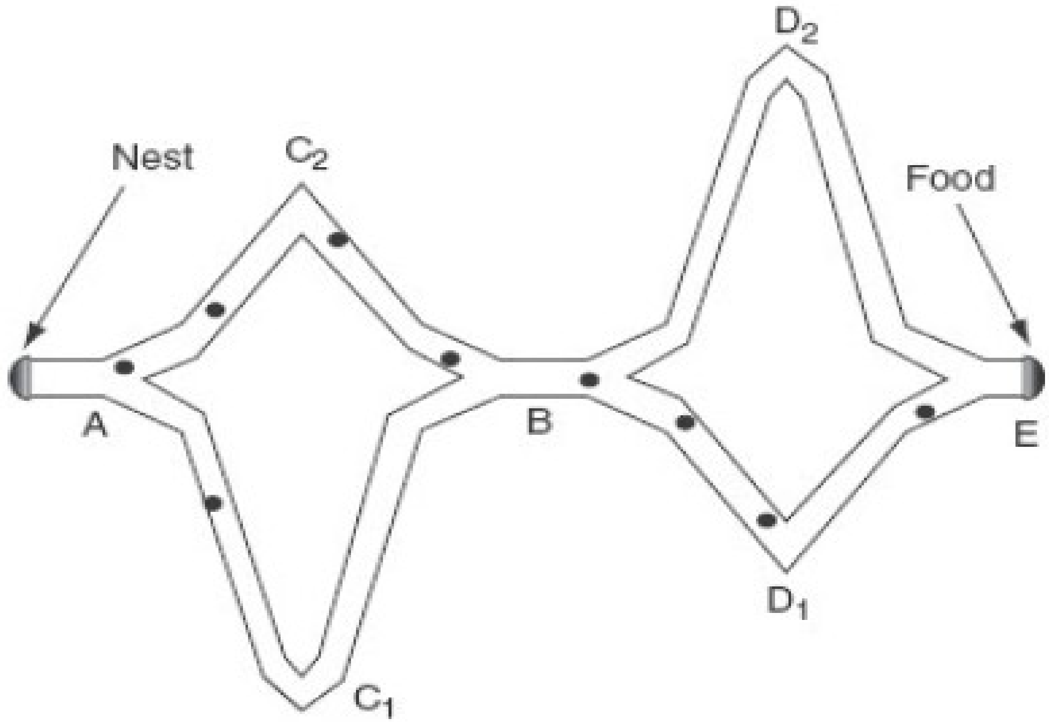
  (3) The (pheromone) trail then becomes reinforced.

  (4) Pheromones are volatile hydrocarbons.
  => evaporation
  => trail will eventually disappear

Krister Wolff, PhD, senior researcher, Chalmers University of Technology
e-mail: krister.wolff@chalmers.se url: www.am.chalmers.se/~wolff

Biological background

- Experiment by Deneubourg *etal* [15]: Study of real ants: *Linepithema humile* (Argentine ants).



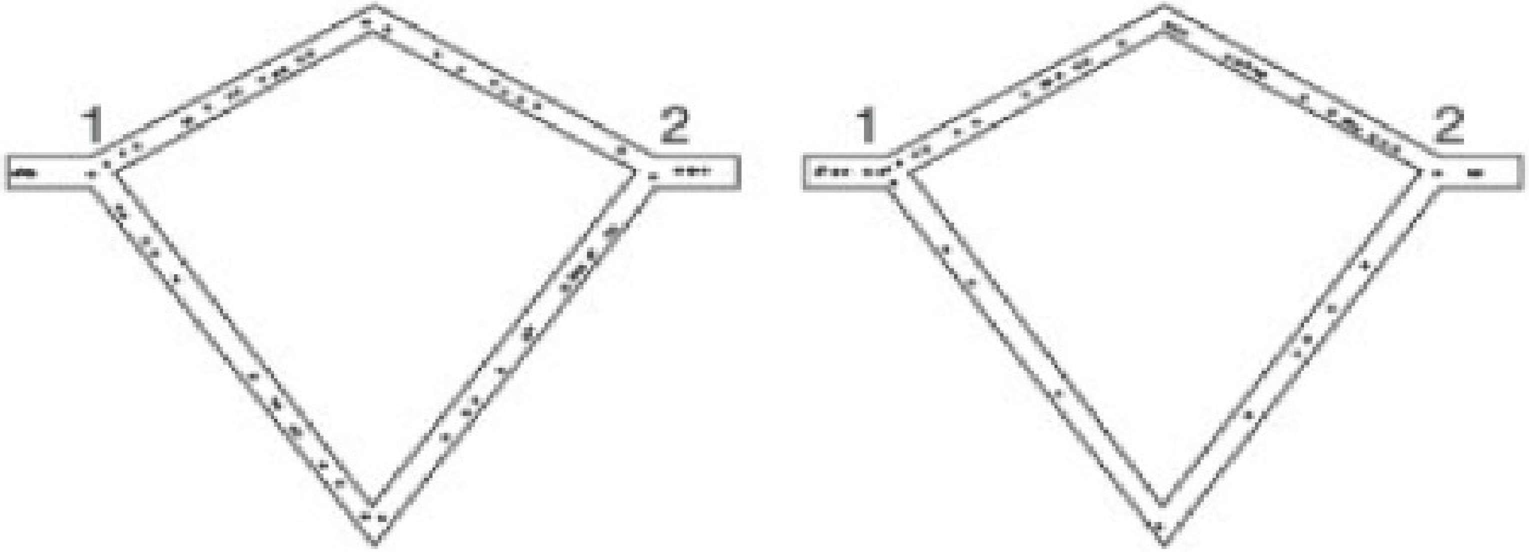
Biological background

- **Numerical model** developed by Deneubourg *etal*:

  Empirical, based on the previous study.

  Simple path; *single* decision point.

  Argentine ants deposit pheromones both on the outbound *and* the inbound part of the motion.



Biological background

- **Deneubourg's empirical, numerical model:**

  (1) outbound decision point 1 (towards food)

  Probability that an ant chooses the short path:

$$p_1^S = \frac{(C+S_1)^m}{(C+S_1)^m + (C+L_1)^m}$$

S_1 = amount of pheromone on the *short* path.

L_1 = amount of pheromone on the *long* path.

C = constant (=20), m = constant (=2)

Probability of selecting the long path: $p_1^L = 1 - p_1^S$

Biological background

- **Deneubourg's empirical, numerical model:**
  (2) inbound decision point 2 (towards nest)
  Probability that an ant chooses the short path:

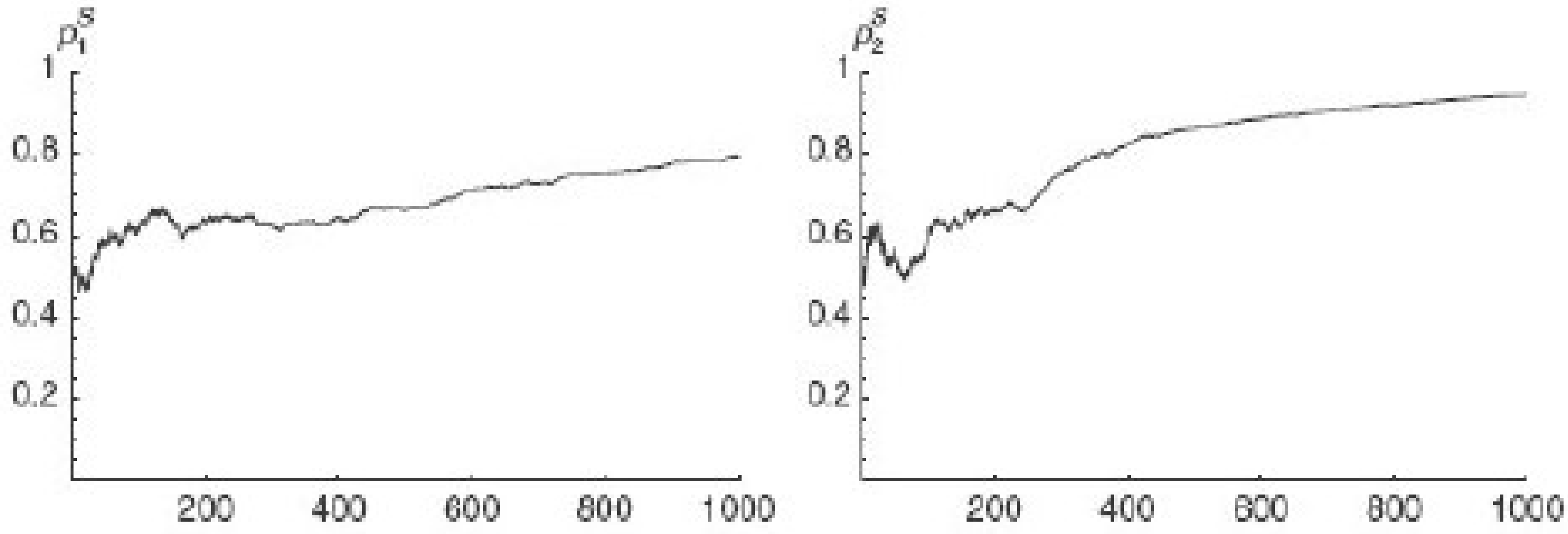
$$p_2^S = \frac{(C+S_2)^m}{(C+S_2)^m + (C+L_2)^m}$$

The pheromone levels S_i and L_i changes over time since the ants deposit pheromones.

With real ants, the *short* path took ~20 s,
the *long* path took ~20*r* s,
r = length ratio, is in [1,2].

Biological background

- **Deneubourg's empirical, numerical model:** Shows good fit with experimental data.



The probabilities p_1^s and p_2^s.

Ant algorithms

- **ACO** algorithms operate by **searching for paths** in a graph.

  A ***solution candidate*** to the problem at hand is represented by ***a path in the graph***.

  In order to solve a problem using a ACO it has to be formulated as ***finding the shortest path*** in a graph.

  Construction graph $G:(N,\varepsilon)$

  N = set of nodes (vertices)

  ε = set of edges (connections between nodes)

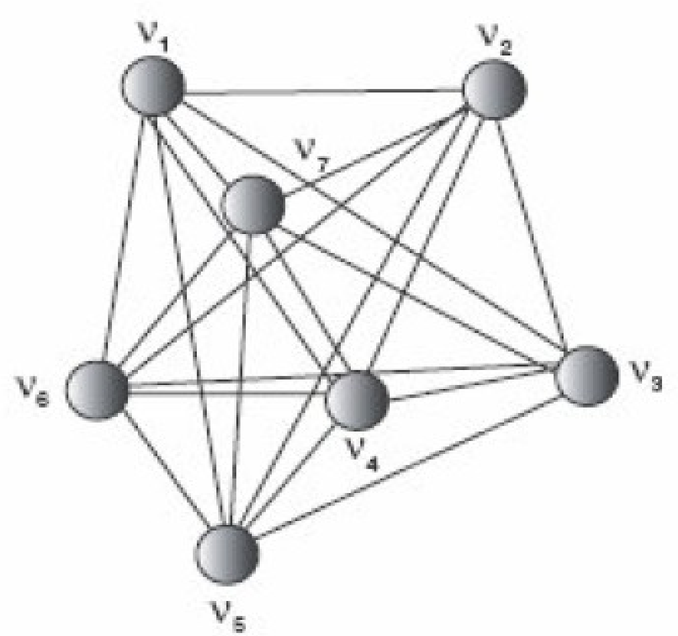
Ant algorithms

- **Construction graph** for ACO:

  Artificial ants are released onto the graph *G* and move according to *probabilistic rules*.

  As they move, they deposit (artificial) pheromone on the edges.

- Edge e_{ij}, connecting node v_j to node v_i is associated with a pheromone level τ_{ij}



- ***Symmetric*** graph is assumed here ($\tau_{ij} = \tau_{ji}$).
- Asymmetric graph is a *directed* graph (τ_{ij} *is not* = τ_{ji})

Ant algorithms

- We shall consider the **Travelling salesperson problem** (TSP) for describing ACO:

  The ***aim in TSP*** is to find the *shortest possible path* that visits each city once (and only once!), and in the final step returns to the city of origin.

  v_i = nodes that represent the cities.

  e_{ij} = straight-line paths connecting city *i* to city *j*.

  n = number of cities (nodes)

  The number of *possible* paths is: $\frac{1}{2}(n-1)!$

  That number grows rapidly with n.

Ant algorithms

- **Ant system** (AS) algorithm: Solve the TSP over *n* nodes using a set of *N* artificial ants, which are released onto the construction graph G:

  Solution candidate = *a path*, e.g. (2, 5, 3, 4, 1).

  Starts with an *empty list* of nodes: S=Ø

  For each movement the index of the current node (city) is added to the list of nodes.

  Tabu list $L_T(S)$: the indices of the already visited nodes.

  In every step an ant chooses its move *probabilistically,* based on:

  (1) pheromone level τ_{ij}, and

  (2) distances between current node v_j and the potential target nodes.

Ant algorithms: AS

- Probability of an ant taking a step from node *j* to node *i*:

$$p(e_{ij}|S) = \frac{\tau_{ij}^{\alpha}\eta_{ij}^{\beta}}{\sum_{\nu_l \notin L_T(S)} \tau_{lj}^{\alpha}\eta_{lj}^{\beta}}$$

where $\eta_{ij} = 1/d_{ij}$, d_{ij} = Euclidean distance

α and β are constants.

- In each step, the current node must be added to the tabu list $L_T(S)$.
- When *all* nodes have been visited *once*, the tour is completed by a return to the city of the origin.

Ant algorithms: AS

- After one iteration, i.e. all the N ants have been evaluated, the **pheromone levels are updated.**

  Pheromone level on edge e_{ij} deposited by ant k:

$$\Delta\tau_{ij}^{[k]} = \begin{cases} \frac{1}{D_k} \text{ if ant } k \text{ traversed } e_{ij} \\ 0 \text{ otherwise} \end{cases}$$

  where D_k = length of the tour generated by ant k.

  Total increase of pheromone level on edge e_{ij}:

$$\Delta\tau_{ij} = \sum_{k=1}^{N} \Delta\tau_{ij}^{[k]}$$

Ant algorithms: AS

- **Complete updating rule,** *including* evaporation:

$$\tau_{ij} \leftarrow (1-\rho)\,\tau_{ij}\,\Delta\tau_{ij}$$

ρ = evaporation rate]0, 1]

Note that *all* edges in the graph is updated simultaneously!

Thus, edges that are *not visited* will only evaporate.

Krister Wolff, PhD, senior researcher, Chalmers University of Technology
e-mail: krister.wolff@chalmers.se url: www.am.chalmers.se/~wolff

Ant algorithms: AS

- The free parameters in AS:

  $\alpha = 1$

  β in [2, 5]

  $\rho = 0.5$

  $N = n$ (i.e. the number of *ants* equals the number of *nodes* in the graph)

  Empirically known to give good performance over a large range of problems.

Krister Wolff, PhD, senior researcher, Chalmers University of Technology
e-mail: krister.wolff@chalmers.se url: www.am.chalmers.se/~wolff

Ant algorithms: AS

- **Initialization** of pheromone trails:

$$\tau_{ij} = \tau_0 = N \frac{1}{D^{nn}}$$

N = The number of ants.

D^{nn} = nearest-neighbor tour (starting at a random node and at each step move to the nearest *unvisited* node).

See Algorithm 4.1 for a summary of AS.

Krister Wolff, PhD, senior researcher, Chalmers University of Technology
e-mail: krister.wolff@chalmers.se url: www.am.chalmers.se/~wolff

Ant algorithms: TSP example

- **Example 4.2**: Computer simulation of AS applied to the TSP:

  50 cities, 50 ants.

  Several parameter combinations were tried:

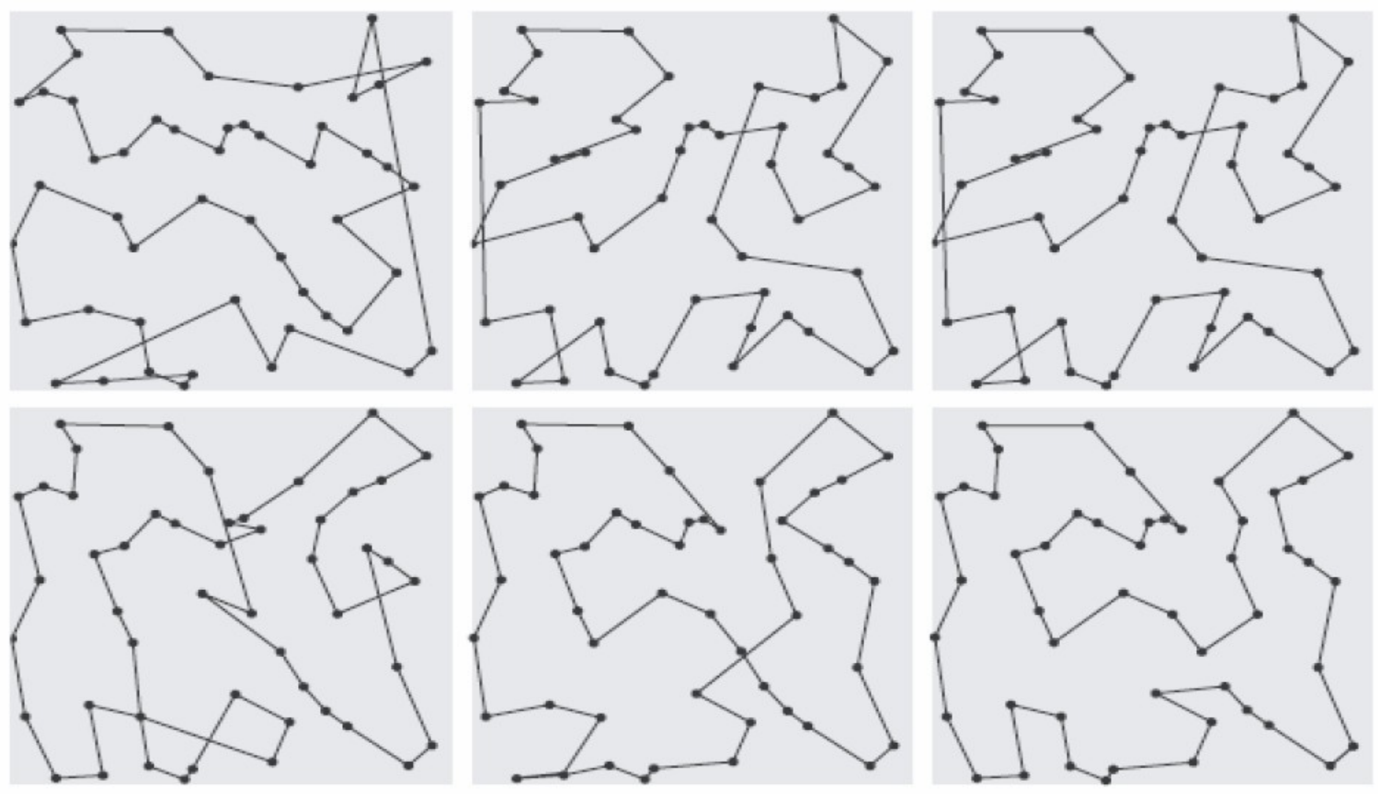
  100 runs for each combination.

  200 iterations for each run

  10000 evaluated paths for each run

Ant algorithms: TSP example

- An instance of the **TSP**. Iterations 1, 2, 3, 5, 10, 100.



Ant algorithms: TSP example

- The algorithm is rather *insensitive* to the choice of parameters:

α	β	ρ	τ_0	$\overline{D}$	s
1	2	0.5	0.3	132.94	1.12
1	2	0.5	0.1	132.95	1.11
1	5	0.5	0.1	132.50	1.03
1	2	0.8	0.1	133.31	1.70
1	2	0.2	0.1	133.14	1.03
1	2	0.5	0.01	132.90	1.16

Best path found so far: D* = 127.28

Ant algorithms: MMAS

- **Max-min ant system** (MMAS):

  Similar to the Ant systems algorithm, but exploit good candidate solutions more strongly:

  Only the “best solution” ant allowed to deposit pheromone, either defined as:

  (1) ***best in current iteration***, or

  (2) ***best so far***.

Ant algorithms: MMAS

- Let D_b denote the length of the *current best tour*. The change in pheromone levels are then given by:

$$\Delta\tau_{ij} = \Delta\tau_{ij}^{[b]}$$

$$\Delta\tau_{ij}^{[b]} = \begin{cases} \frac{1}{D_b} \text{ if the best ant traversed edge } e_{ij} \\ 0 \text{ otherwise} \end{cases}$$

Note that *all* edges are updated!

Only those visited by the *best ant* receive positive contributions.

Ant algorithms: MMAS

- Strong *exploitation* of good solutions may lead to *stagnation*.

  => After update: $\tau_{ij} \leftarrow (1-\rho)\tau_{ij} \Delta \tau_{ij}$

  the levels are modified according to:

$$\text{if } \tau_{ij} > \tau_{\max} \Rightarrow \tau_{ij} \leftarrow \tau_{\max}$$

$$\text{if } \tau_{ij} < \tau_{\min} \Rightarrow \tau_{ij} \leftarrow \tau_{\min}$$

  $\tau_{\min}$ defines the *lower bound* of the probability of visiting edge e_{ij}

  => Convergence theorem B.3.2

Krister Wolff, PhD, senior researcher, Chalmers University of Technology
e-mail: krister.wolff@chalmers.se url: www.am.chalmers.se/~wolff

Ant algorithms: MMAS

- **Pheromone initialization**:

$$\tau_{ij} = \tau_{\max} \; \forall \, ij \in [1, n]$$

=> greater degree of *exploration* in the early stages.

- What is the value of $\tau_{\max}$?

Theoretical <u>upper limit</u> of pheromone level, for any edge e_{ij} [B.3.1]:

$$\frac{1}{\rho D^*}$$

D^* = length of the *optimal* tour.

Ant algorithms: MMAS

- During optimization D^* is not known...

Therefore set: $$\tau_{\max} = \frac{1}{\rho D_b}$$

D_b = length of the *current best tour*.

Initially, take: $$\tau_{ij} = \tau_{\max} = \frac{1}{\rho D^{nn}}$$

Whenever a new best tour is found, update $\tau_{\max}$ using D_b. => $\tau_{\max}$ *changes dynamically.*

Krister Wolff, PhD, senior researcher, Chalmers University of Technology
e-mail: krister.wolff@chalmers.se url: www.am.chalmers.se/~wolff

Ant algorithms: MMAS

- τ_{min} is set empirically:

$$\tau_{\text{min}} = \frac{\tau_{\text{max}}\left(1 - \sqrt[n]{0.05}\right)}{\left(\frac{n}{2} - 1\right)\sqrt[n]{0.05}}$$

for TSP with n>>1.

Ant algorithms: MMAS

- The MMAS can get stuck in a local optima:

  Restart the algorithm using the most recent estimate of D^*.

- Better results for the TSP can be obtained by *alternating* the definition of the *best solution*:

  (1) ***best in current iteration***, or

  (2) ***best so far***.

Ant algorithms: MMAS

- **Example 4.3**. Max-min ant system applied to the same instance of the TSP as in example 4.2:

α	β	ρ	τ_{max}	τ_{min}	$\overline{D}$	s
1	2	0.5	0.0133	3.43×10^{-5}	136.11	3.38
1	5	0.5	0.0133	3.43×10^{-5}	128.87	1.69
1	2	0.8	0.0083	2.14×10^{-5}	129.19	1.82
1	2	0.2	0.0333	8.57×10^{-5}	128.66	1.57

Generally *wider distribution* of results than for AS, but in several cases also *better results* than for AS.

Applications of ACO

(1) Problems involving ***routing***: Telecomunication networks, TSP,

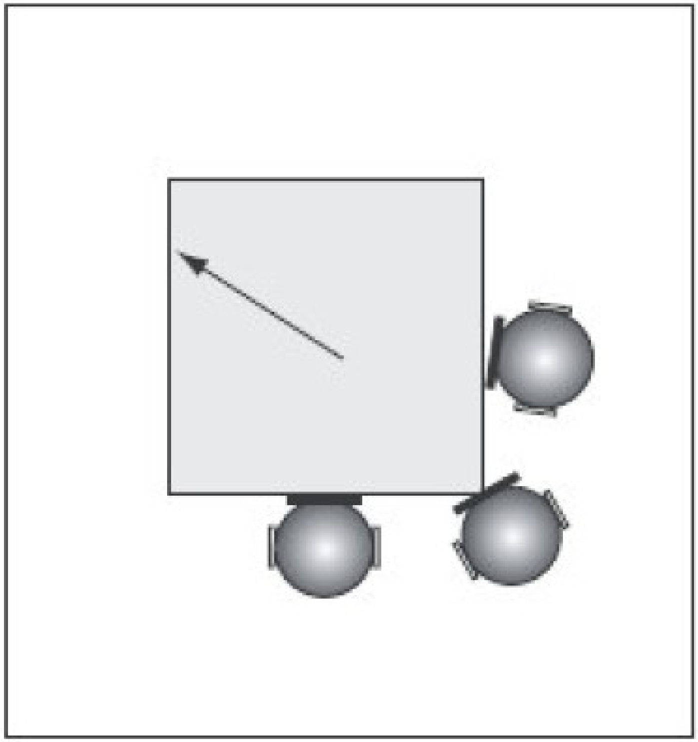
(2) Job shop **scheduling**:

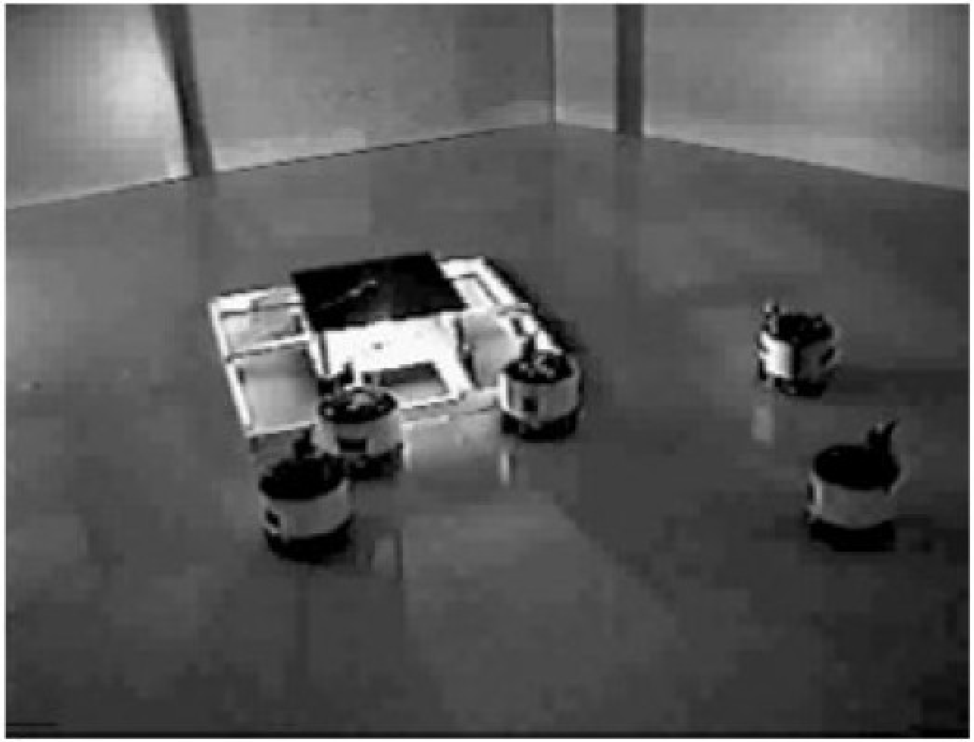
n jobs, consisting of a *finite* sequence of operations are to be processed in *m* machines, in the shortest possible time. Constraints regarding the order of precedence between the operations.

(3) Applications that are *inspired* by the behaviors of real ants are common in **swarm robotics**:
(3.1) Co-operative transport of heavy objects using autonomous robots.

(3.2) Dynamic bridge-building

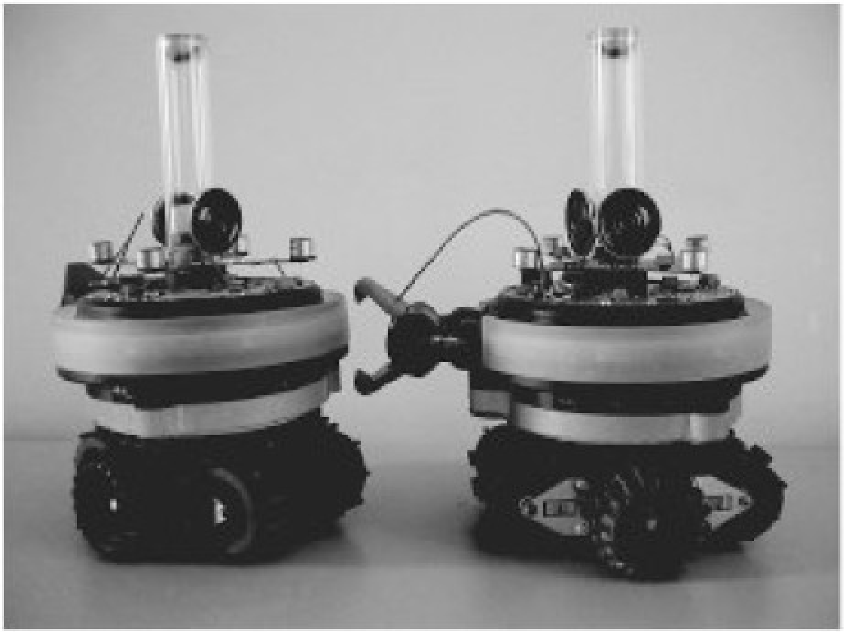
Applications of ACO





- Co-operative transportation: Box-pushing task

Applications of ACO



- Swarm-bot project: Dynamic bridge-building
- Cooperative transport:
- http://www.youtube.com/watch?v=CJOubyiITsE