Ameisen Systeme

Prof. Dr. Dagmar Monett Díaz
Dagmar.Monett-Diaz@hwr-berlin.de

Fachbereich Duales Studium, Fachrichtung Informatik

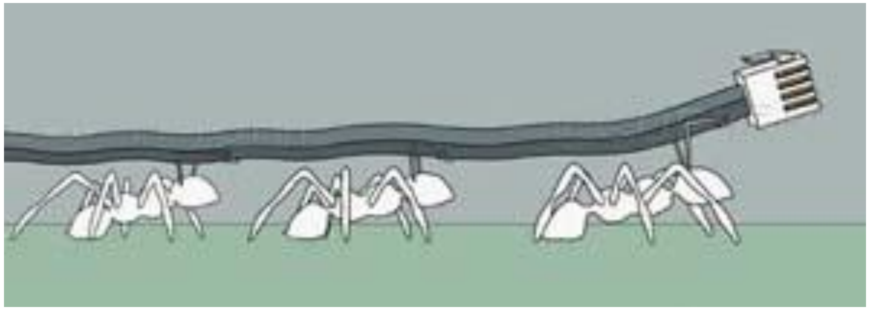
Hochschule für
Wirtschaft und Recht Berlin
Berlin School of Economics and Law

Ant Colony Systems

Ant Colony Optimization

Outline

- What is Swarm Intelligence? Motivation
- Ant Colonies
- Ant Colony Optimization
- Optimization problem
  - Characteristics
  - Algorithm
  - Examples
  - Modifications
  - Applications



What is Swarm Intelligence?

- "***Swarm Intelligence*** *is the property of a system whereby the collective behaviors of (unsophisticated) agents interacting locally with their environment cause coherent functional global patterns to emerge*."
- **Characteristics** of a swarm:
  - distributed, no central control or data source;
  - no (explicit) model of the environment;
  - perception of environment, i.e. sensing;
  - ability to change environment.

What is Swarm Intelligence?

- **Swarm systems** are examples of *behavior-based systems* exhibiting:
  - multiple lower level competences;
  - situated in environment;
  - limited time to act;
  - autonomous with no explicit control provided;
  - problem solving is emergent behavior;
  - strong emphasis on reaction and adaptation;
  - collective intelligence

Motivation

Motivation

Motivation

Motivation

Motivation

- **Robust** nature of <u>animal problem-solving</u>
  - *simple* creatures exhibit *complex behavior*;
  - behavior modified by *dynamic environment*.
- **Emergent behavior** observed in:
  - bacteria
  - *ants*
  - bees
  - ...

Motivation

- 10^{18} living insects (rough estimate)
- ~2% of all insects are social
- Social insects are:
  - All ants, all termites
  - Some beeps, some wasps
- 50% of all social insects are ants
- Avg weight of one ant between 1 and 5 mg
- Ants have colonized Earth for 100 million years, *Homo sapiens* for 50,000 years

Motivation

Each element of the swarm has its **own simple behaviour**, and a set of rules for interacting with its fellows, and with the environment.

Every element is the same – there is **no central controller**.

However, X **emerges** as a result of these local interactions.

E.g. ants finding food, termites building mounds, jellyfish.

Motivation

Ant colony size: from as few as 30 to millions of workers

Coordination of activities:

Set of dynamical mechanisns whereby *structure appears at the global level* as the result of *interactions among lower-level components*

The rules specifying the interactions among the system‘s constituent units are executed on the basis of *purely local information*, without reference to the global pattern, which is an *emergent property of the system* rather than a property imposed upon the system by an external ordering influence.

Stigmergy

Indirect communication via interaction with environment [Gassé, 59]

i.e. swarm behaviour **emerges** from the way individuals communicate through and affect their environment

Self-organization

- How do social insects achieve self-organization?
  - **Communication is necessary**
  - **Two types of communication:**

Direct: antennation, trophallaxis (food or liquid exchange), mandibular contact, visual contact, chemical contact, etc.

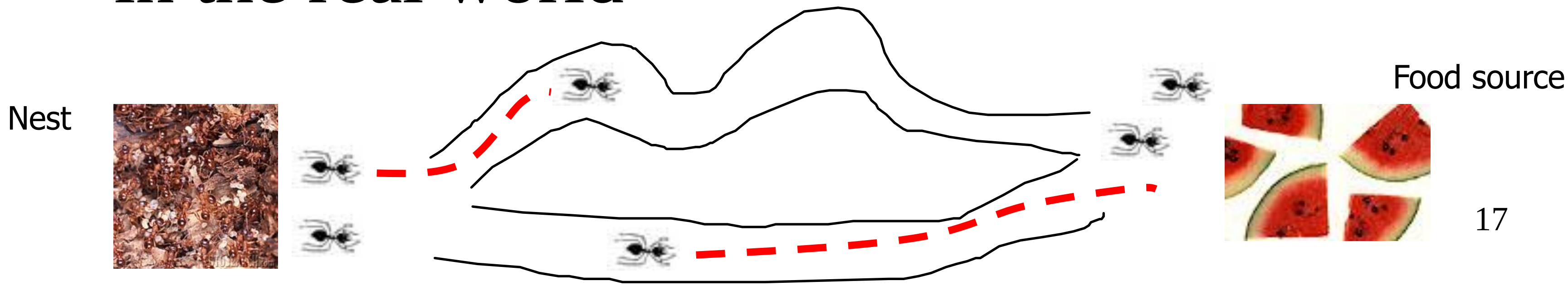
Indirect: two individuals interact indirectly when one of them modifies the environment and the other responds to the new environment at a later time (stigmergy!!)

Ant Colonies

- Ants are behaviorally unsophisticated; *collectively* perform complex tasks.
- Ants have highly developed sophisticated **sign-based stigmergy**
  - communicate using *pheromones*;
  - trails are laid that can be followed by other ants.

Pheromone Trails

- Species lay pheromone trails traveling from nest to nest, or possibly in both directions
- pheromones accumulate with multiple ants using path. Pheromones also evaporate
- helps in avoiding suboptimal solutions – local optima
- In ACO: may differ from how it takes places in the real world



Introduction ACO

- Ant Colony Optimization (**ACO**) algorithms attempt to imitate or to simulate the *process of collective ants behavior*;

  <u>Important</u>:

  – to understand how do **ant colonies** behave,

  – Mechanisms and strength of **stigmergy**

  – *collective intelligent behavior* and how to use it.

History of Ant Algorithms

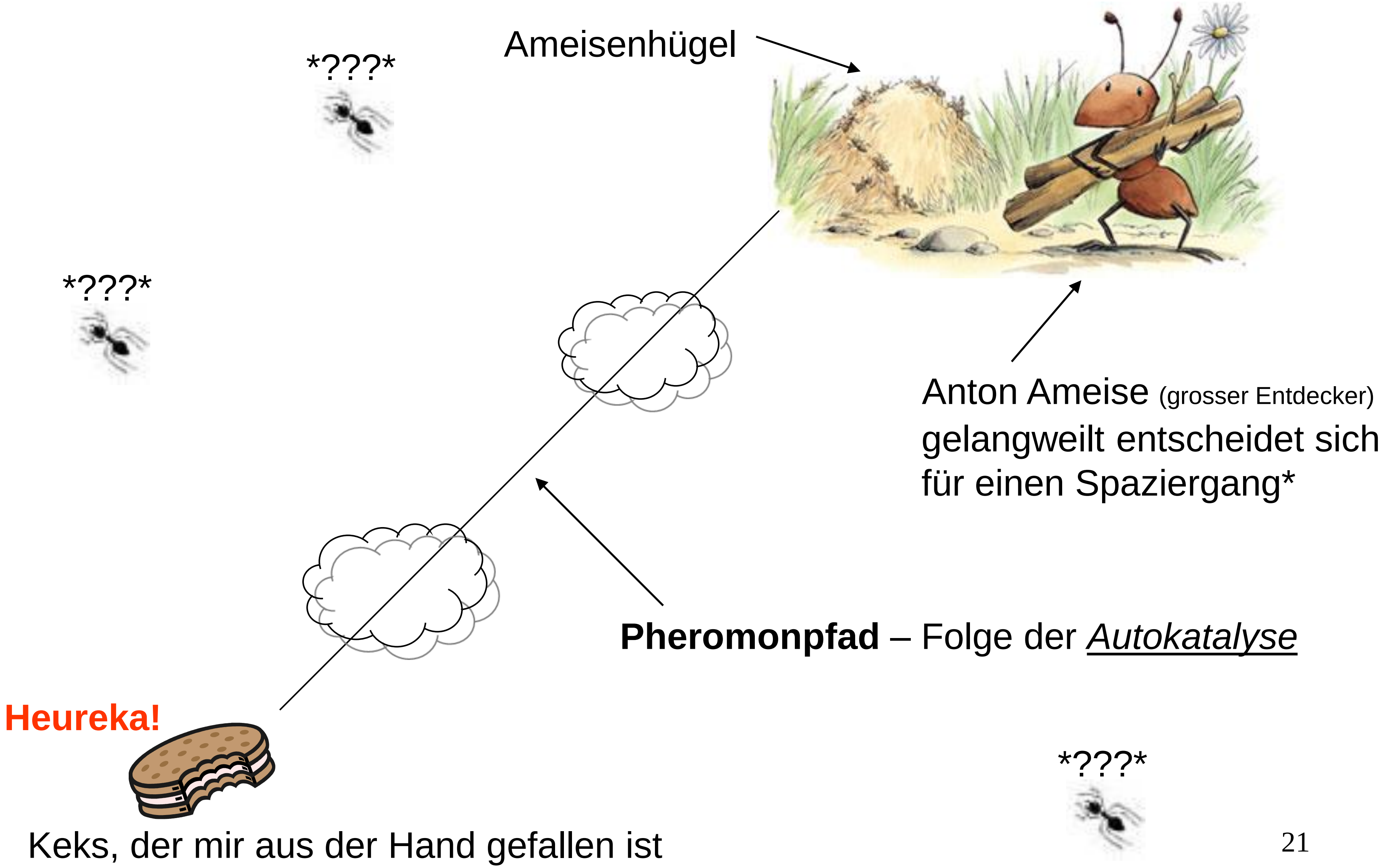
- Goss et al. 1989, Deneuborg et al. 1990, experiments with Argentine ants
- Dorigo et al. 1991, applications to shortest path problems
- *Now*: established method for various optimization problems

What are ant algorithms?

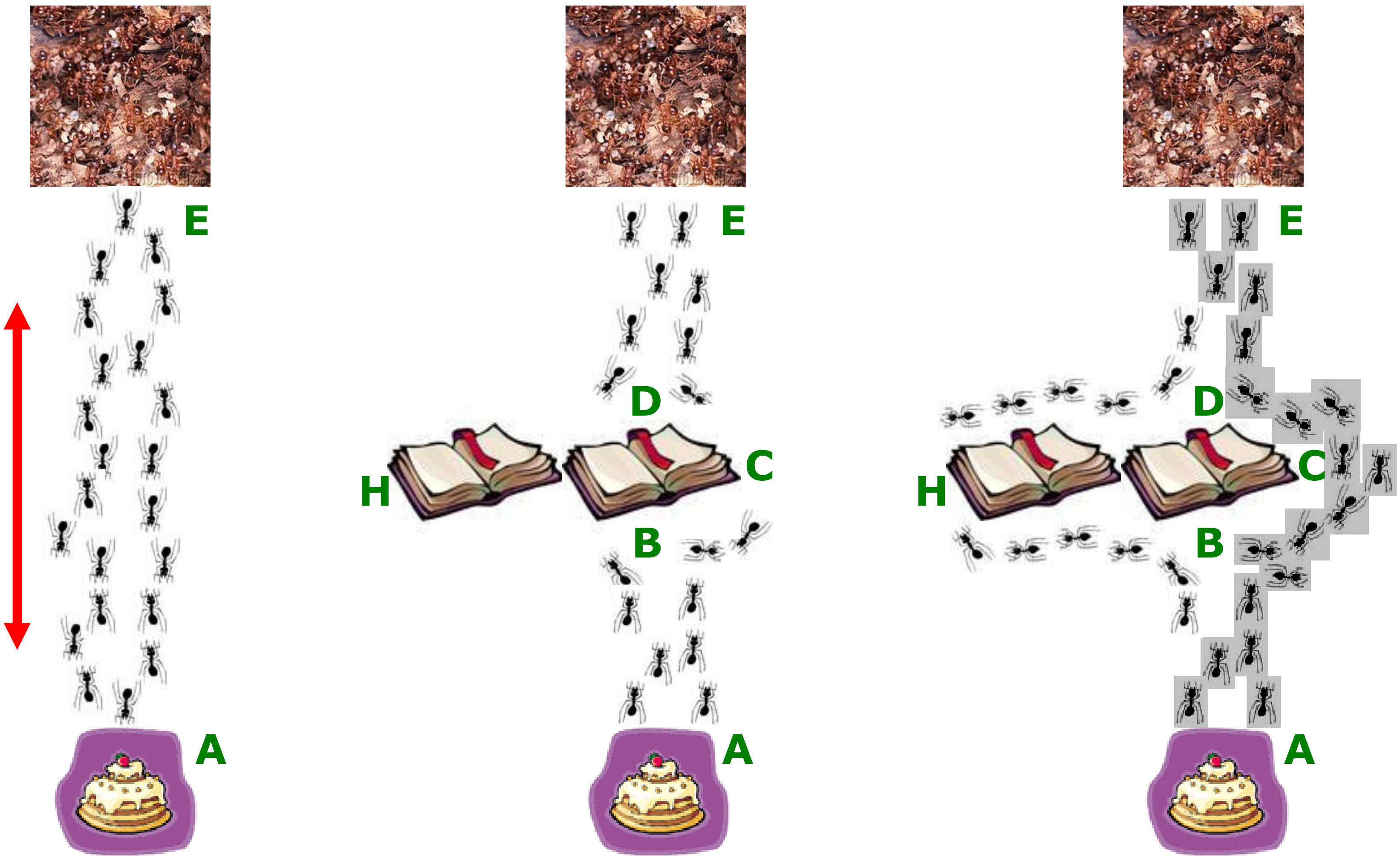
“**Ant algorithms** are **multi-agent systems** that exploit *artificial stigmergy* as a means for coordinating artificial ants for the solution of computational problems”

Ants in action

(The history of „Anton Ameise“)

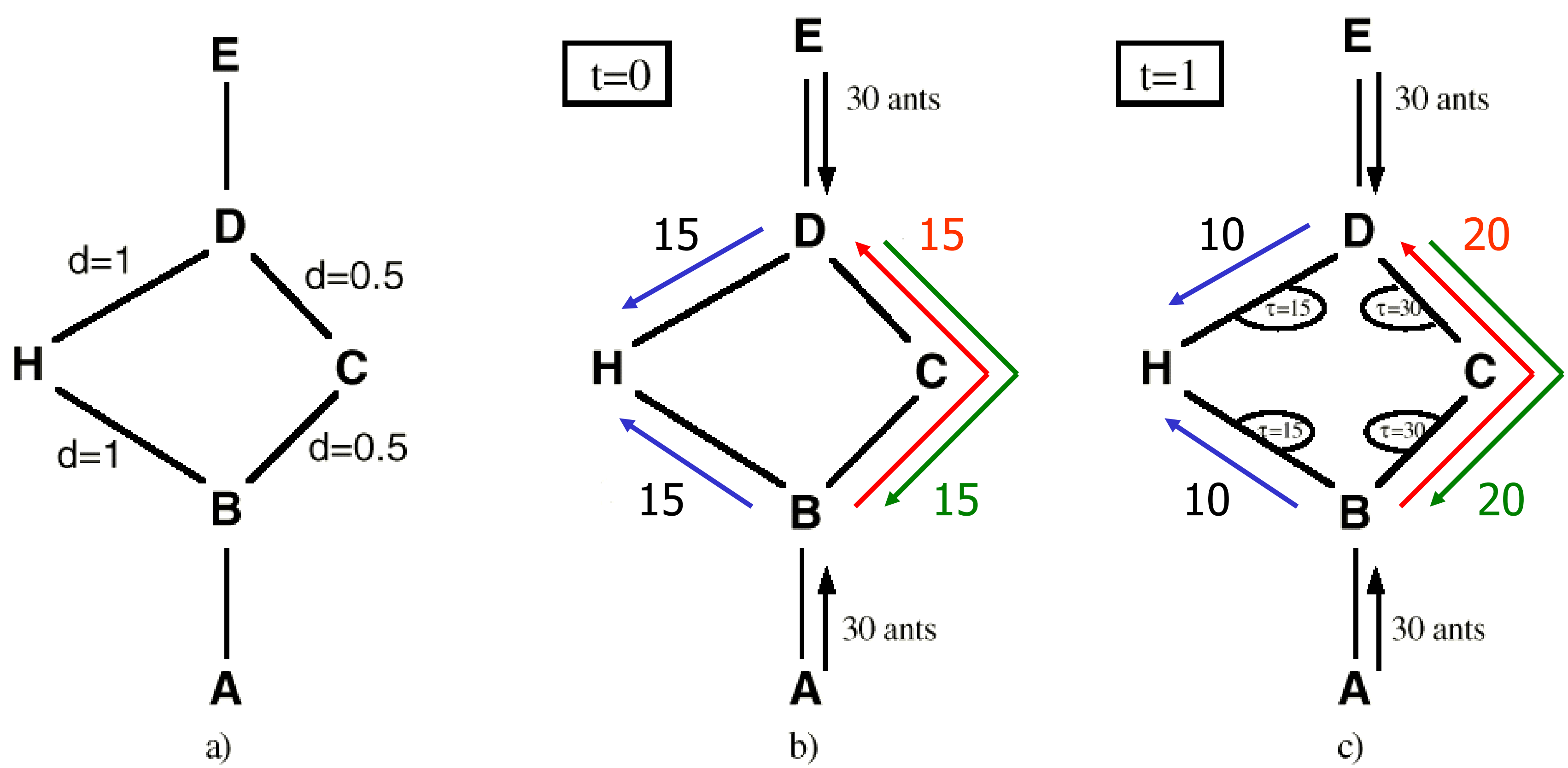


Ants in Action (Stigmergy)



Use of tour A-B-C-D-E increase, A-B-H-D-E decline with time

Ants in Action (discrete)



Assumptions: discrete time intervals, at t=0 no pheromone on edges
30 ants both A->B & E->D
ants move at 1 unit per timestep
strong visited/marked routes synonym with smallest path

ACO

- **ACO** is a meta-heuristic that uses strategies of real ants to solve optimization problems
- ACO was initially proposed by Colorni, Dorigo and Maniezzo
- The *main underlying idea* was that of **parallelizing search** over several constructive computational threads, all based on a *dynamic memory structure* incorporating information on the effectiveness of previously obtained results and in which the behavior of each single agent is inspired by the behavior of real ants

Optimization problem for ACO

- **Optimization problem in general:**
  - given: X, f:X → ℝ, f:X → {True, False}
  - find: $x \in X$, so that f(x) minimal (or maximal) and c(x) feasible.

- **Optimization problem for ACO:**
  - find
    - ***basic components*** C = $\{c_1,\ldots,c_n\}$, so that
    - ***partial solution*** subsets S are in C,
    - ***feasible (partial) solution*** F are in C,
    - ***solution*** s in C
    - ***cost function*** f.
  - then
    - iterative extend (feasible) partial solutions with basic components in order to find a solution s, so that f(s) is minimal (or maximal).
    - ***Pheromone*** deposit on each component c_i to control the search

TSP: The problem

A salesman must visit **n** cities, passing through each city only once, beginning from one of them which is considered as his base, and returning to it.

The cost of the transportation among the cities (whichever combination possible) is given.

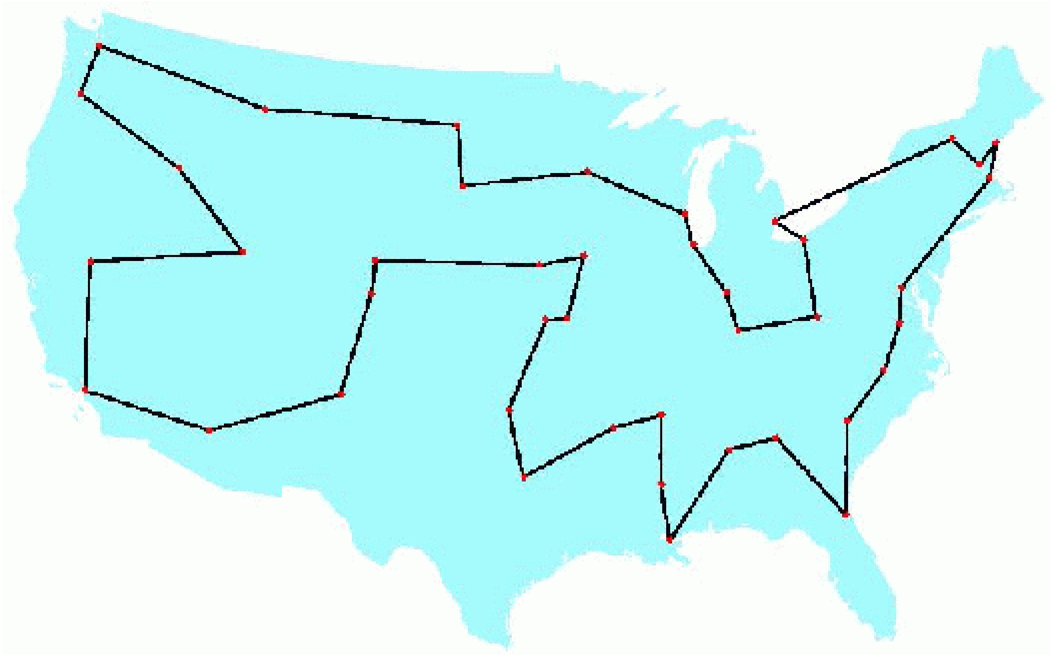
The program of the journey is requested, that is the order of visiting the cities **in such a way that the cost is the minimum**.

TSP: Examples

From http://www.tsp.gatech.edu

TSP: Examples

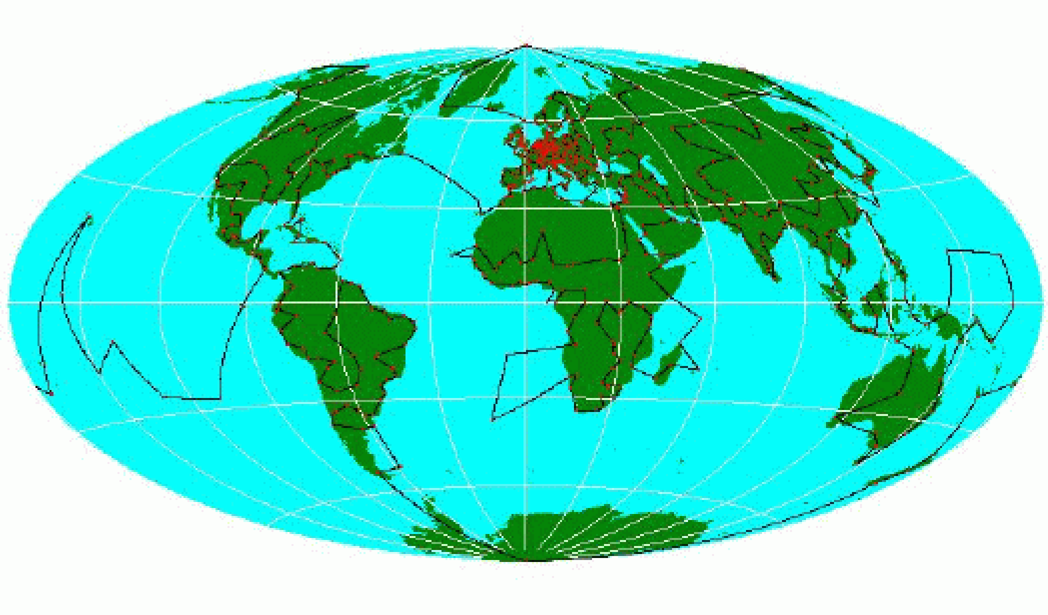
By George Dantzig, Ray Fulkerson, and Selmer Johnson (1954)
Original instance = 49 cities (one city from each of the 48 states in the U.S.A. and adding Washington, D.C.). Costs of travel = road distances
Solved instance: 42-city problem obtained by removing Baltimore, Wilmington, Philadelphia, Newark, New York, Hartford, and Providence.



TSP: Examples

By Groetschel and Holland (1987)

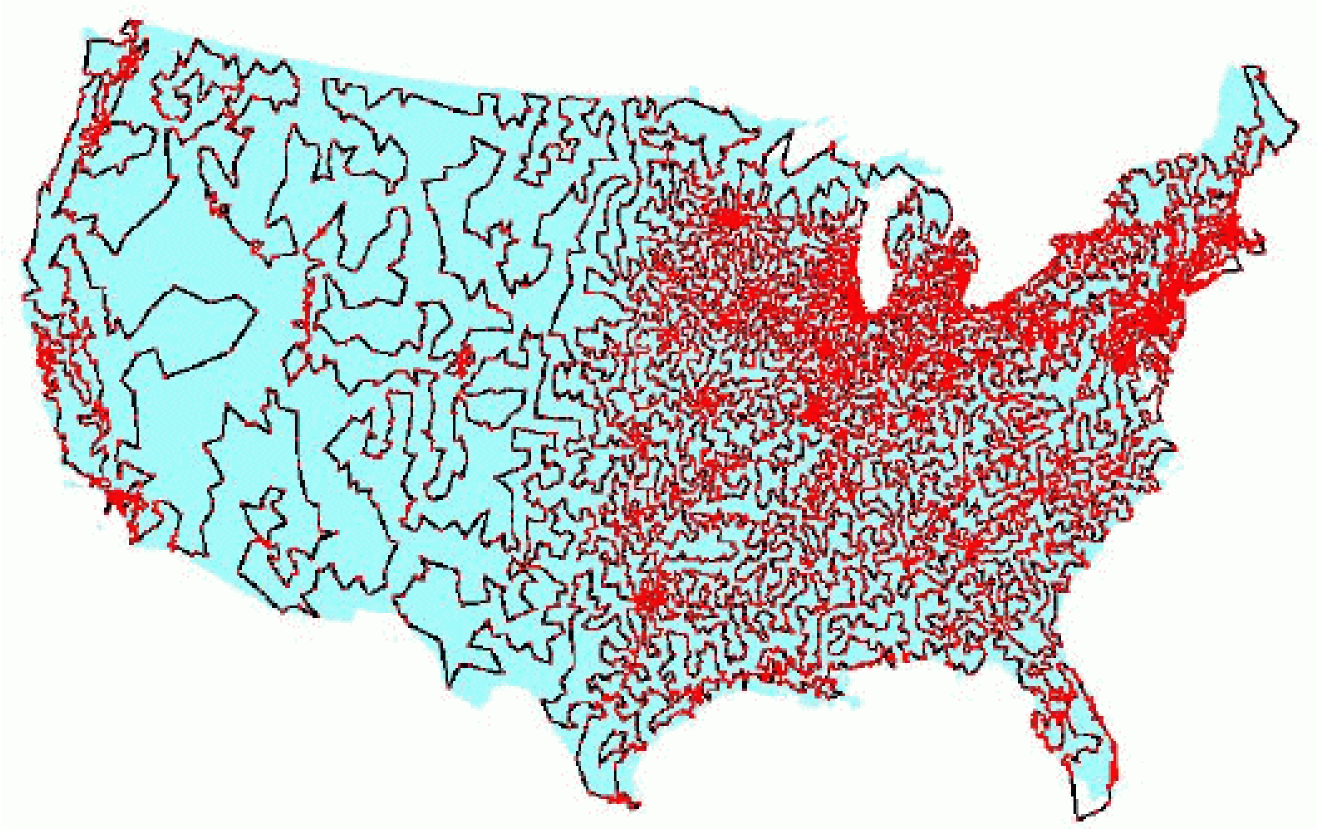
<u>Solved instance</u>: 666 interesting places in the world



TSP: Examples

By Applegate, Bixby, Chvátal, and Cook (1988)

<u>Solved instance</u>: 13,509 city locations in U.S.A. having populations of at least 500



TSP: Examples

By Applegate, Bixby, Chvátal, and Cook (2001)

<u>Solved instance</u>:
15,112 German cities

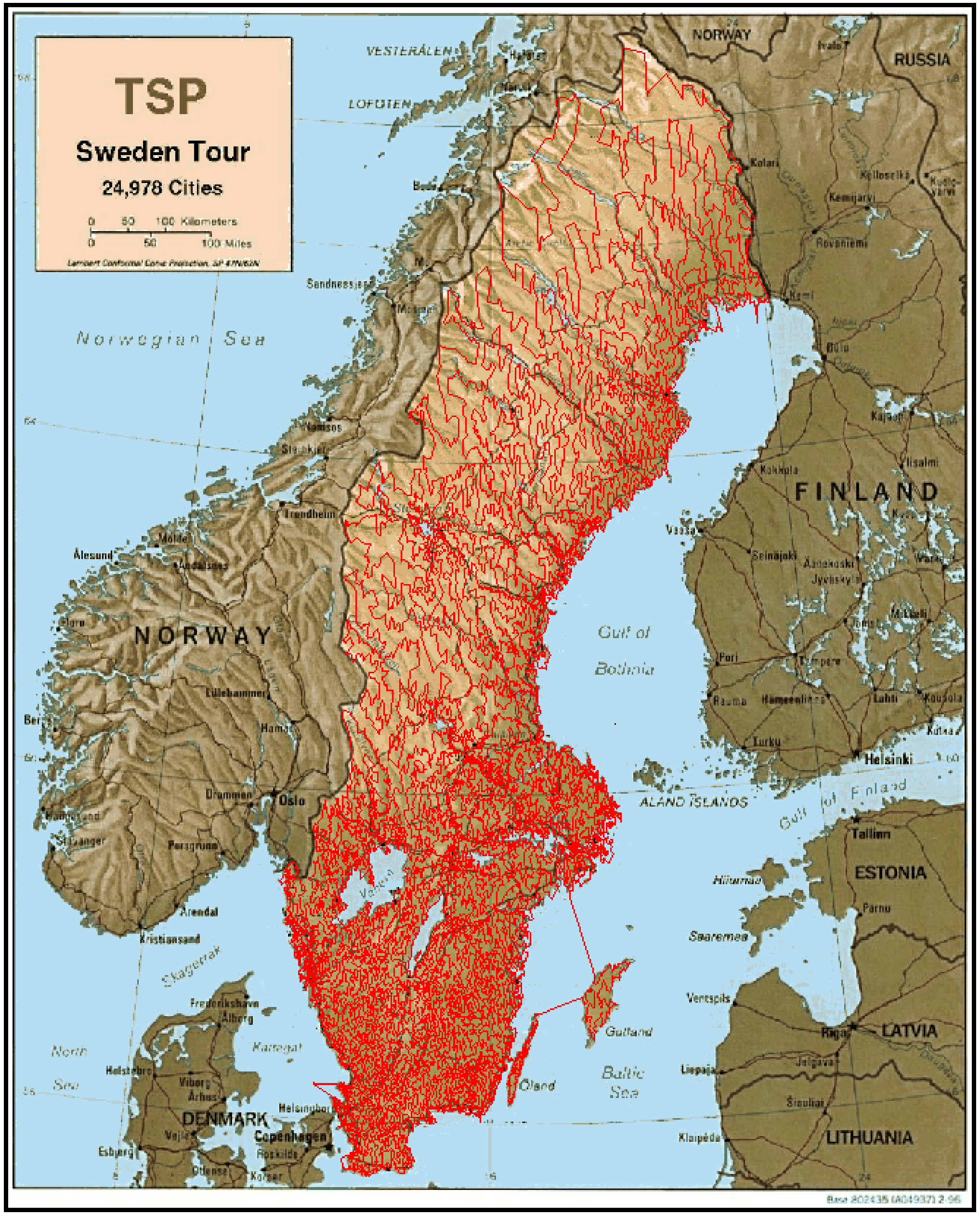
The computation was carried out on a network of 110 processors located at Rice University and at Princeton University. The total computer time used in the computation was 22.6 years, scaled to a Compaq EV6 Alpha processor running at 500 MHz.

TSP: Examples

By Applegate, Bixby, Chvátal, Cook, and Helsgaun (2004)

<u>Solved instance</u>:

24,978 cities in Sweden

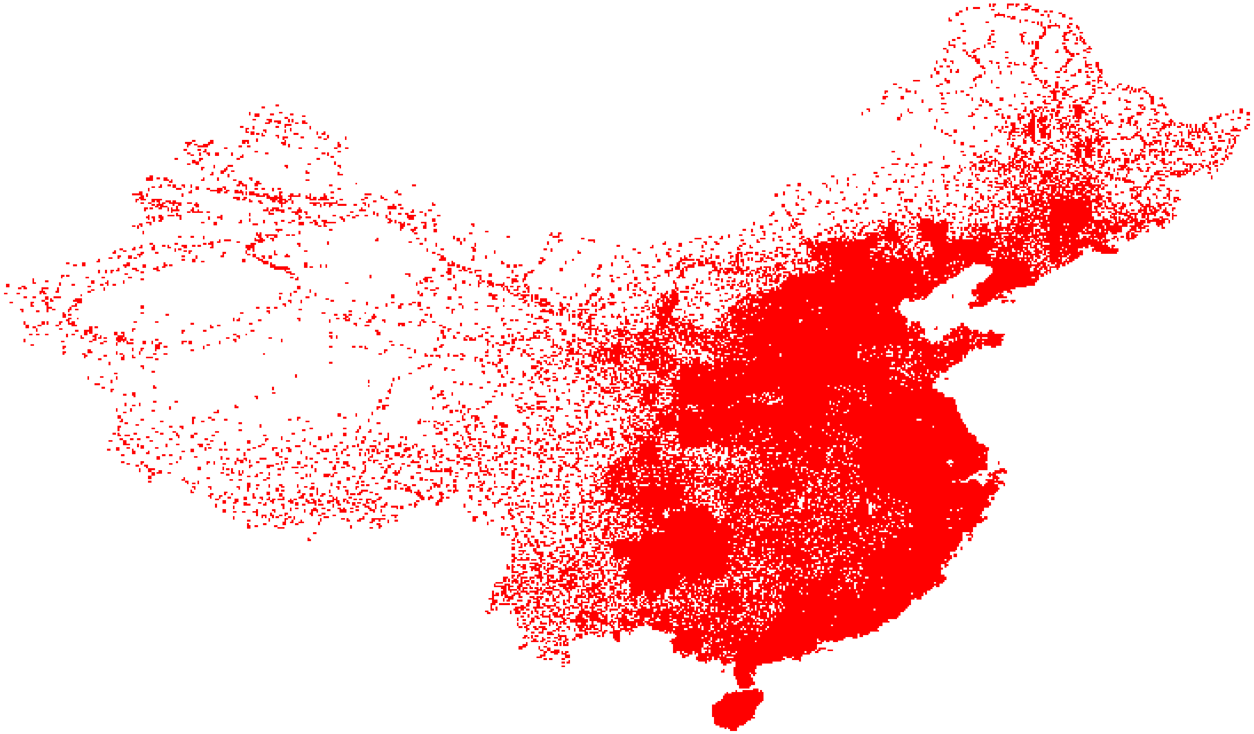


TSP: Examples

By Hung Dinh Nguyen (2003)

<u>Solved instance</u>:

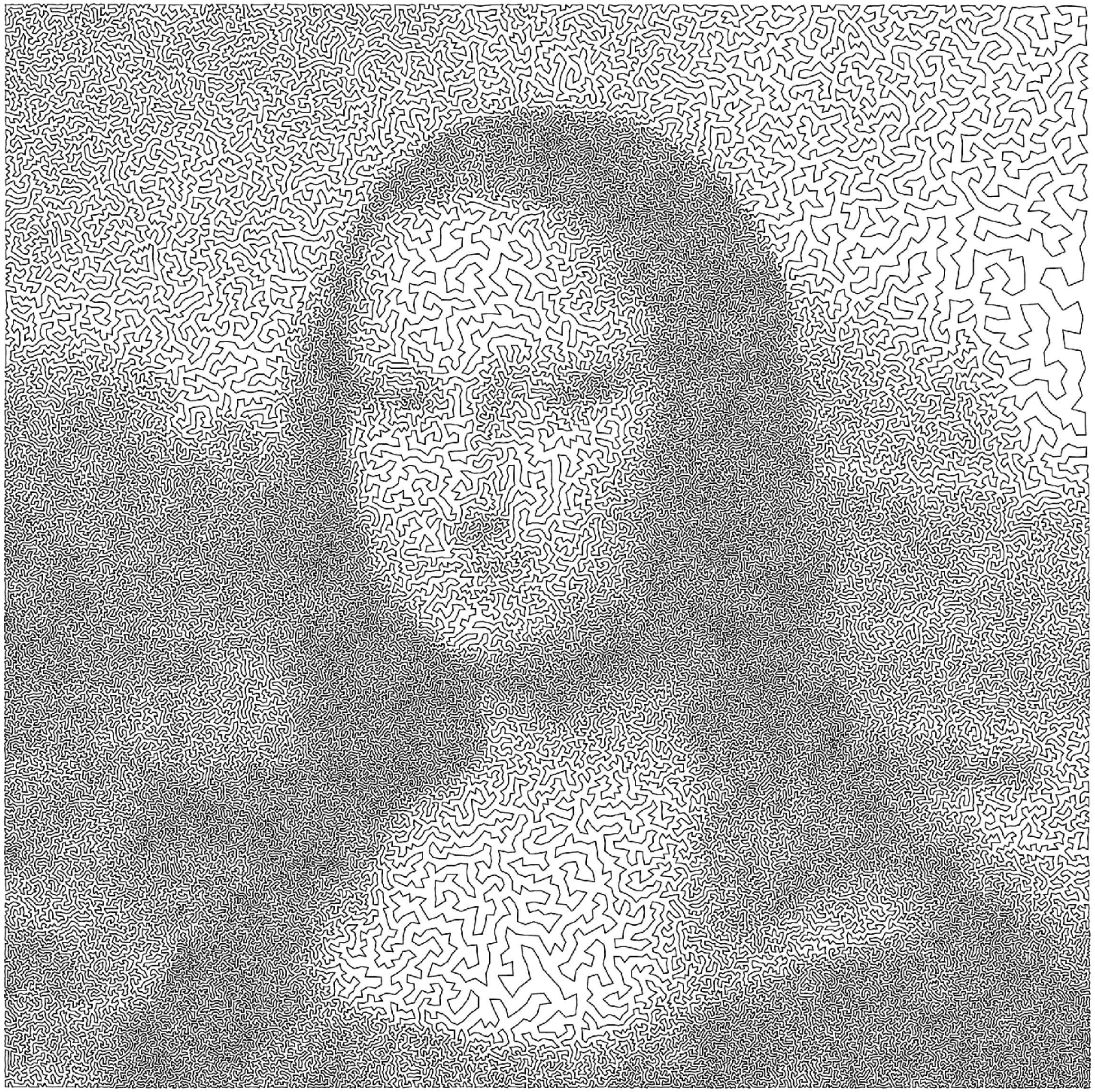
71,009 cities in China



TSP: Examples

Current best:
by Yuichi Nagata (2009)

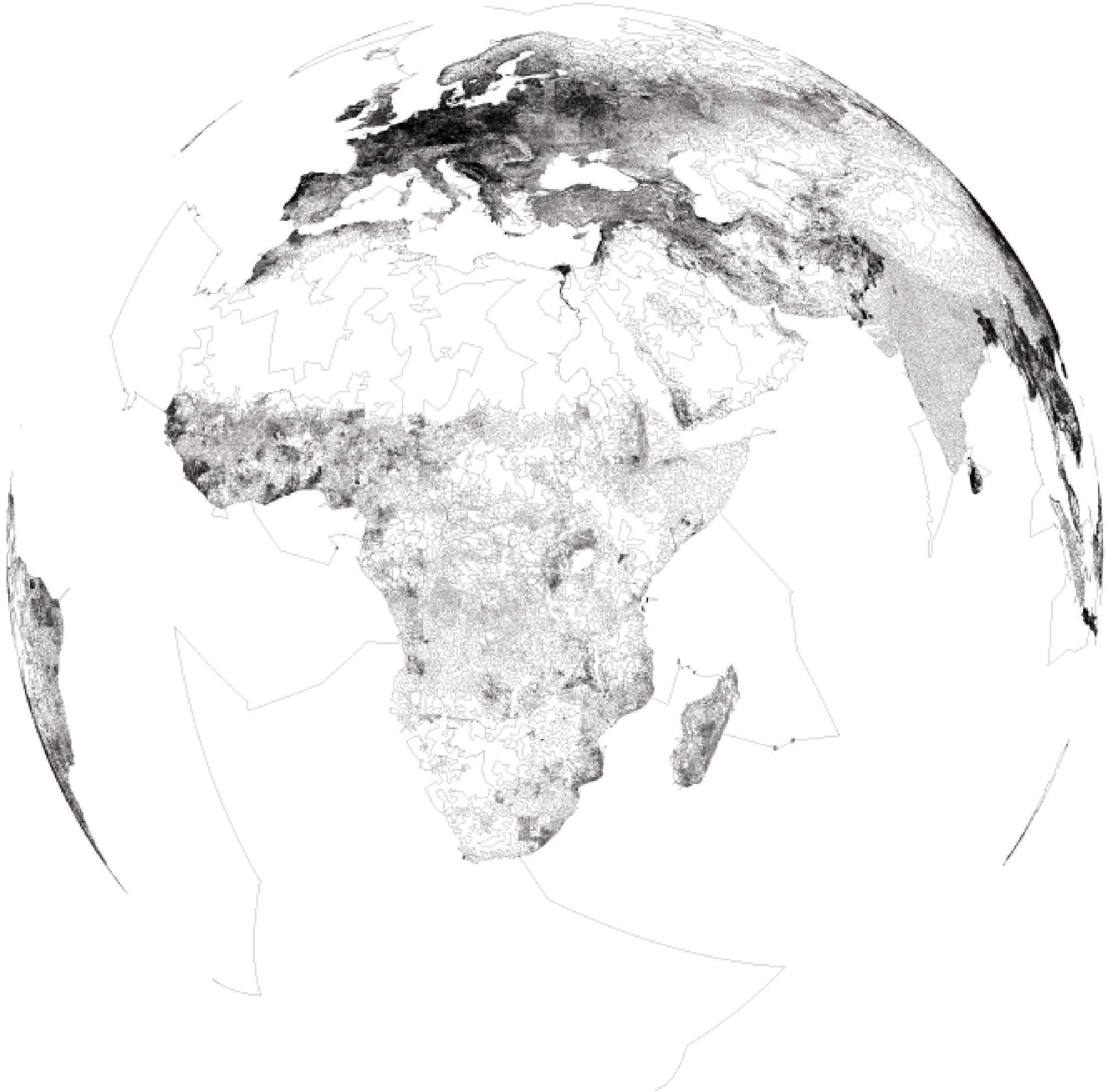
<u>Solved instance</u>:
100,000 cities
(Mona Lisa TSP)



TSP: Examples

By Helsgaun (2009)

Solved instance:
1,904,711 cities
(World TSP)



TSP: Examples

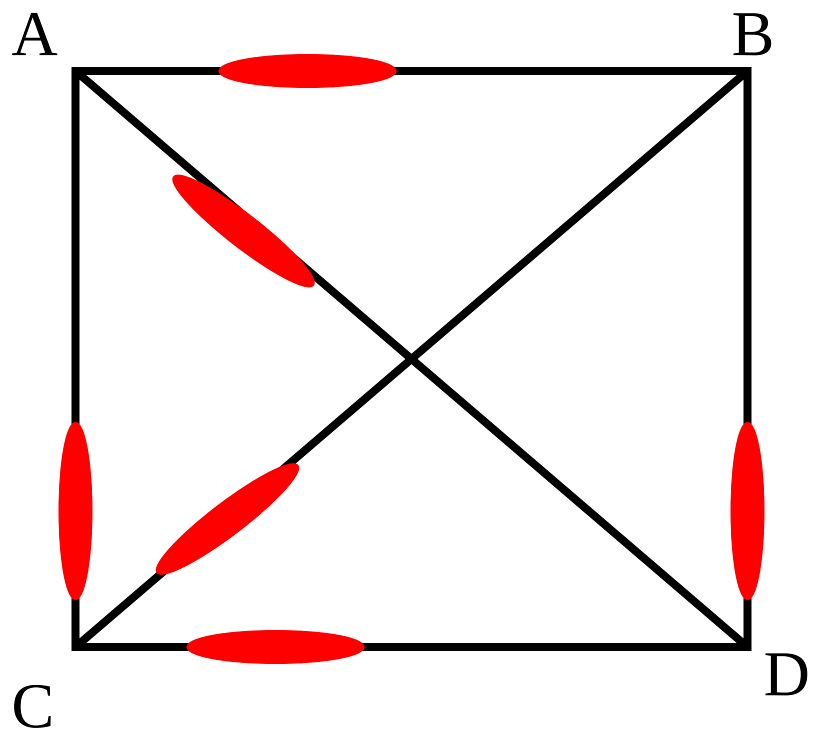
By Groetschel

Solved instance: 52 locations in Berlin

(See berlin52:
- TSPLIB95
- berlin52.tsp
- optimal tour)

E.g. A 4-city TSP

Initially, random levels of pheromone are scattered on the edges



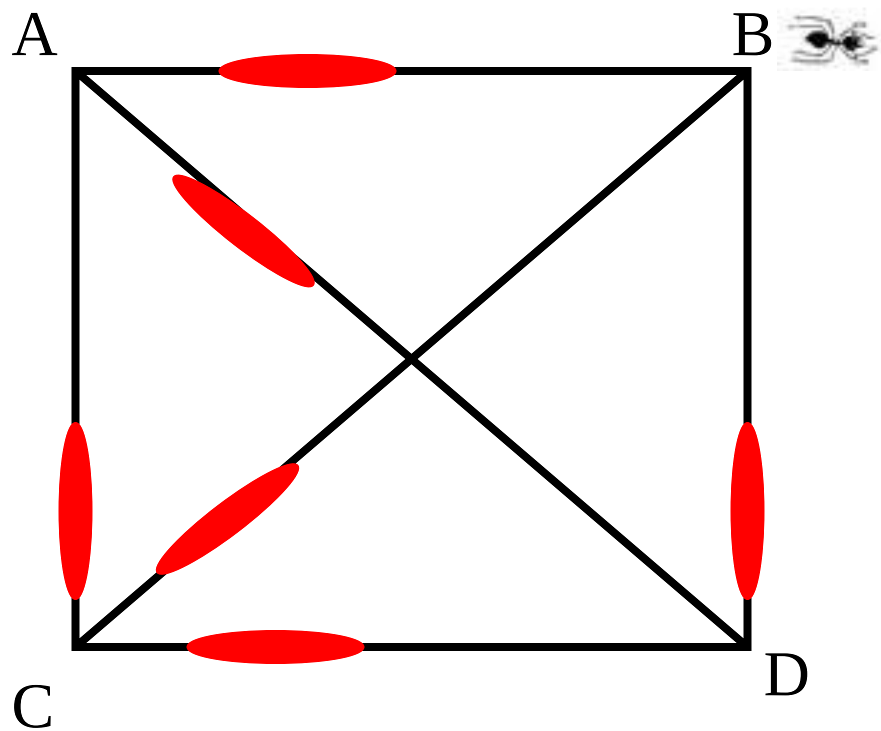
Pheromone

Ant

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

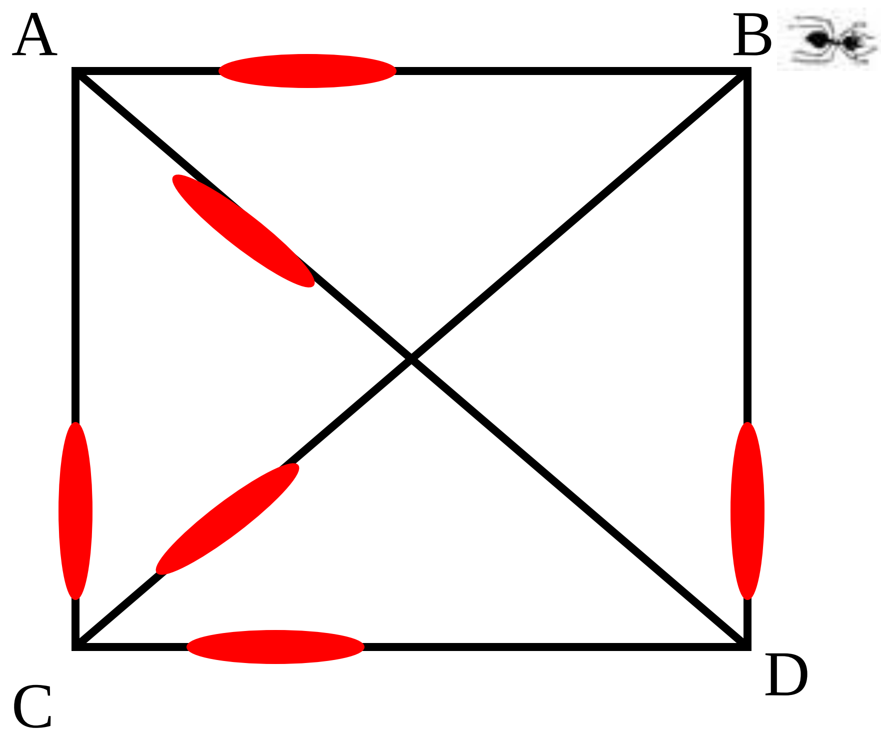
AB: 10, AC: 10, AD, 30, BC, 40, CD 20

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



 Pheromone

 Ant

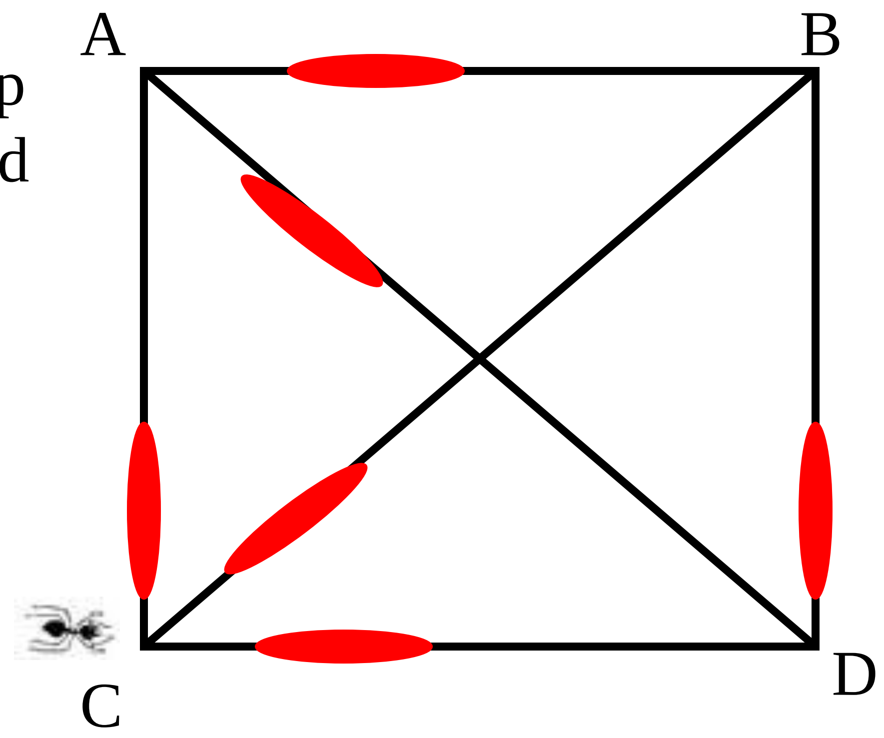
AB: 10, AC: 10, AD, 30, BC, 40, CD 20

E.g. A 4-city TSP

The ant is now at AC, 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

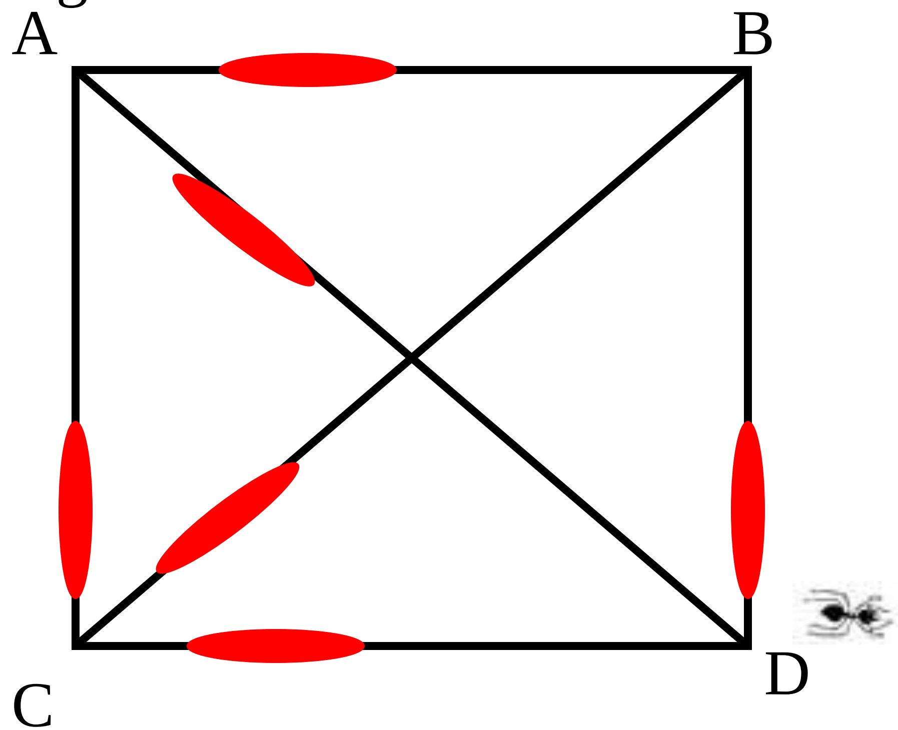
 Ant

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:



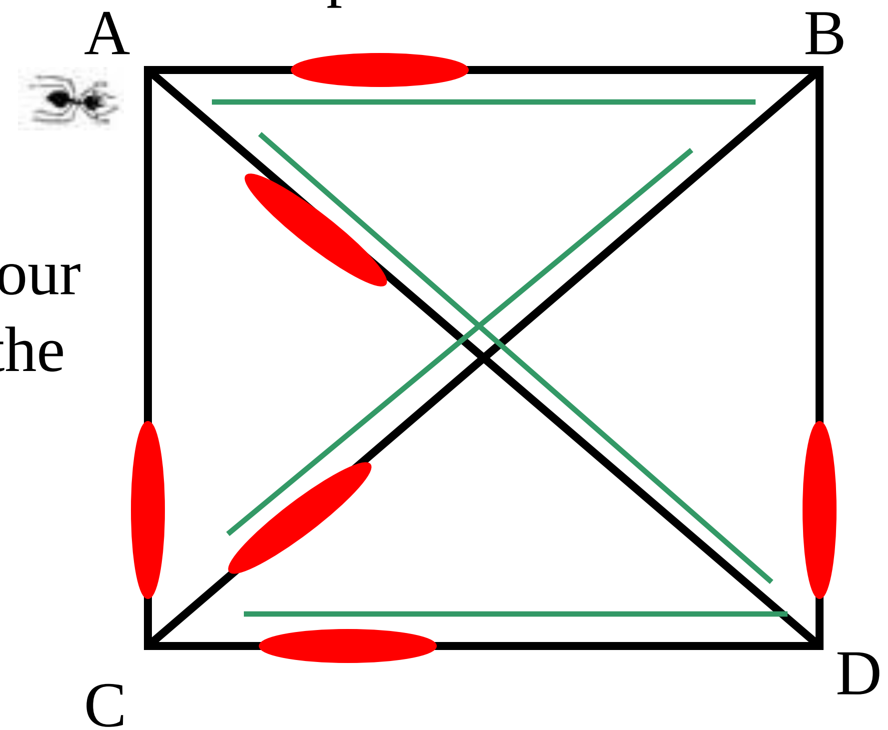
 Pheromone

 Ant

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

E.g. A 4-city TSP

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



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

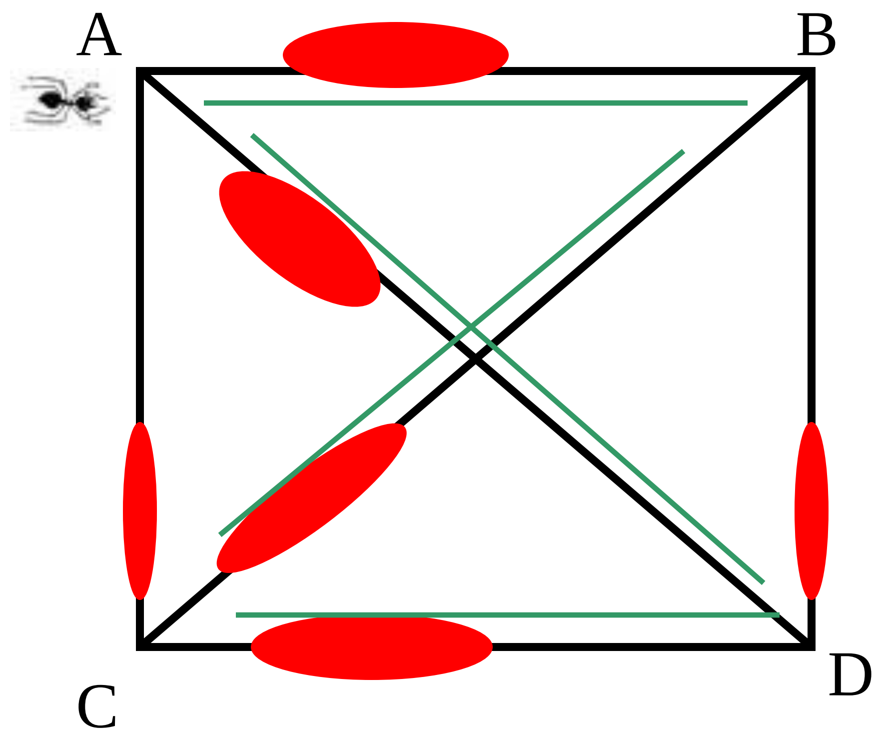
 Pheromone

 Ant

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



Pheromone

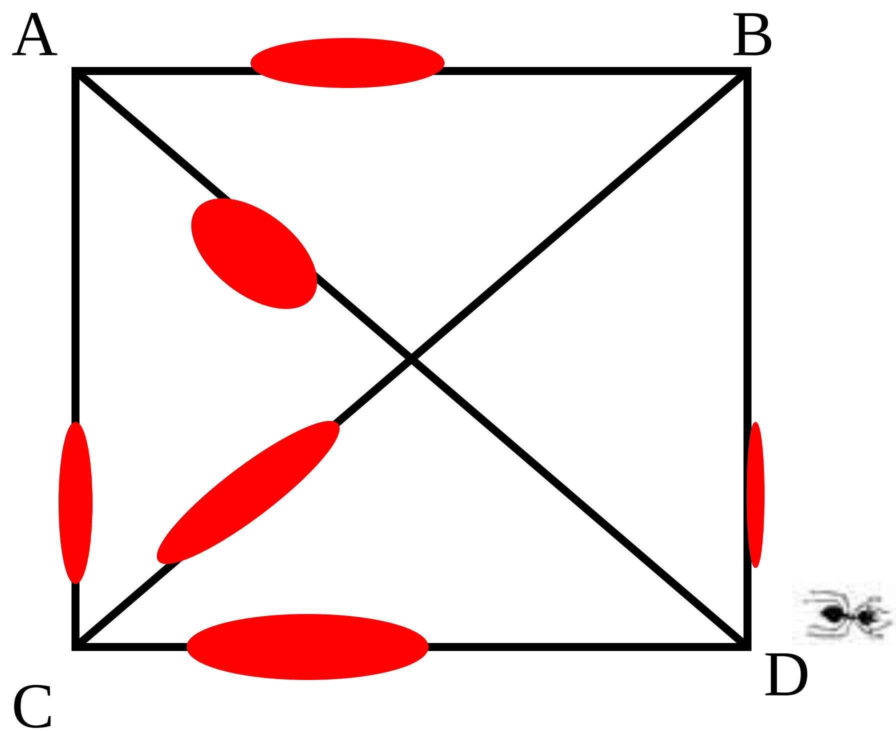
Ant

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?



 Pheromone

 Ant

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

Optimization problem for ACO

- more rigorous mathematical models.
- TSP has been a popular problem for the ACO models.
  - several reasons why TSP is chosen…..

Key concepts:

- **Positive feedback** – build a solution using local solutions, *by keeping good solutions in memory*
- **Negative feedback** – want to avoid premature convergence, *evaporate the pheromone*.
- **Time scale** – number of runs are also critical.

Design choices

- Ants are given a **memory** of visited nodes
- Ants **build solutions probabilistically** without updating pheromone trails
- Ants deterministically backward retrace the forward path to **update pheromone**
- Ants deposit a quantity of pheromone function of the **quality of the solution** they generated

Ant System

- Developed 1991 by Marco Dorigo
- Used to solve TSP
- Transition from city i to j depends on:
  - **Tabu list:** list of visited cities
  - **Visibility:** η_{ij} =1/lenght$_{ij}$; represents local information – heuristic desirability to visit city j when in city i.
  - **Pheromone trail, marks:** $\tau_{ij}(t)$ for each edge – represents the learned desirability to visit city j when in city i.

- Generally, have several ants searching the solution space

  Nr. cities = Nr. ants

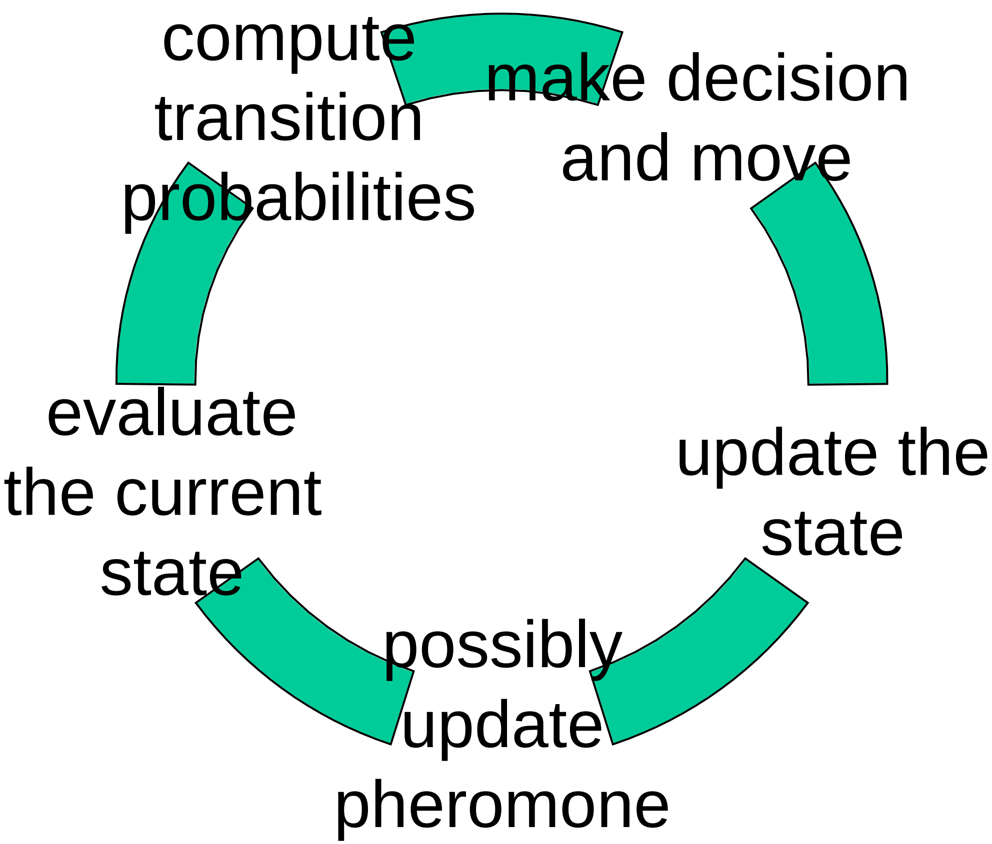
General Ant Colony Heuristic

- Ants generation and activity:
- while resources available: create ant
- for each ant:

  1. initialize

  2. let ant run until a solution is found

  3. possibly: update pheromone and routing table



Transition Rule

- Probability of ant k going from city i to city j:

Pheromone-matrix, trails, marks

Visibility-matrix, local info, 1/lenght

$$p_{ij}^{k}(t) = \frac{\left[\tau_{ij}(t)\right]^{\alpha} \cdot \left[\eta_{ij}\right]^{\beta}}{\sum_{\ell \in J_i^k} \left[\tau_{il}(t)\right]^{\alpha} \cdot \left[\eta_{il}\right]^{\beta}}$$

- Alpha and beta are adjustable parameters:

α = sensitivity of the algorithm to pheromone

β = sensitivity of the algorithm to distance

Transition Rule

$$p_{ij}^{k}(t) = \frac{\tau_{ij}(t)^{\alpha} . \eta_{ij}^{\ \beta}}{\sum_{\ell \in J_i^k} \tau_{il}(t)^{\alpha} . \eta_{il}^{\ \beta}}$$

- Alpha = 0 : represents a greedy approach
- Beta = 0 : represents rapid selection of tours that may not be optimal.
- Thus, a tradeoff is necessary.

Pheromone update

- **Pheromone update** :

$$\Delta\tau_{ij}^{k} = Q / L^{k}(t) \quad if\ (i,j) \in T^{k}(t)\ else\ 0$$

- T is the tour done at time t by ant k, L is the total length of that tour, Q is a heuristic parameter.
- **Pheromone decay**:

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

„pheromone persistence“, $0 < \rho \leq 1$

ACO - Metaheuristic

init pheromone τ_i:=const for each component c_i;
while termination condition not met:
  for all ants i: **construct_solution**(i);
  for all ants i: **global_pheromone_update**(i);
  for all pheromones i: **evaporate**: $\tau_i := (1-\rho)\cdot\tau_i$;

ρ = „pheromone persistence"
$0 < \rho \leq 1$

Constraint „c(x)=True"?

construct_solution(i); init s:={ };

„trail intensities"

while s is not a solution:

choose c_j with probability p = $\begin{cases} 0 & j \text{ not allowed} \\ \dfrac{\tau_j^{\alpha} \cdot \eta_j^{\beta}}{\sum_{j' \text{ allowed}} \tau_{j'}^{\alpha} \cdot \eta_{j'}^{\beta}} & \text{otherwise} \end{cases}$

„visibility"

expand s by c_j;

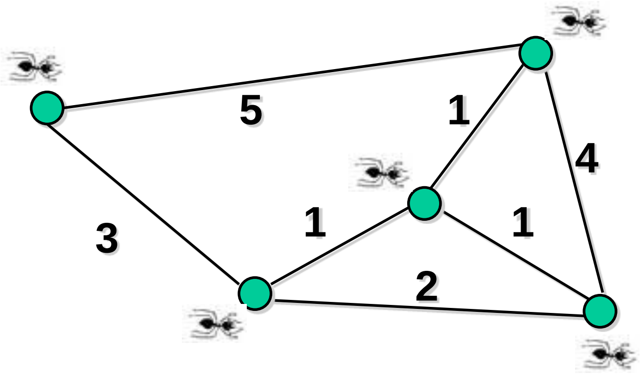
global_pheromone_update(i);

for all c_j in the solution s:
  increase pheromone: $\tau_j := \tau_j$ + const / f(s);

Cost function

Ant-Cycle for TSP

- Tabu List
- Random Walk
- Priorities
- Return and evaluate tour length



Tabu List

n : Nr. of cities = a tour
m : Nr. of ants
k : Ant index
s : pointer to Tabu list (current city)

For each action on each iteration from the Ant-Cycle:
Insert for each ant k the visited city

For k := 1 to m do
  insert town of ant k in $\text{Tabu}_k(s)$
od

Random Walk & Priorities

i, j : edge between nodes i, j
n_{ij} : visibility: 1/distance(i, j)
α : Weights for marking
β : Weights for close nodes

allowed_k : for k in i feasible, adjacent, not visited cities

From Ant-Routing-Table:

Transition probability from city i to city j for ant k

$$p_{ij}^{k}(t) = \frac{[\tau_{ij}(t)]^{\alpha} \cdot [n_{ij}]^{\beta}}{\sum_{l \in allowed_k} [\tau_{il}(t)]^{\alpha} \cdot [n_{il}]^{\beta}} \quad if\ j \in allowed_k,\ else\ 0$$

α, β are control parameters that determine the sensitivity of the algorithm to distance and pheromone

Return and Evaluate

i,j : Edges between nodes i, j
τ_{ij} (t): Marks at time t
ρ : evaporation each (t, t+n)

Q/L_k : Const/tour length ant k

With Tabu List, Ant-Routing-Table:

Mark_delta = Marks Sum for all ants k that have walked the edge (i, j)

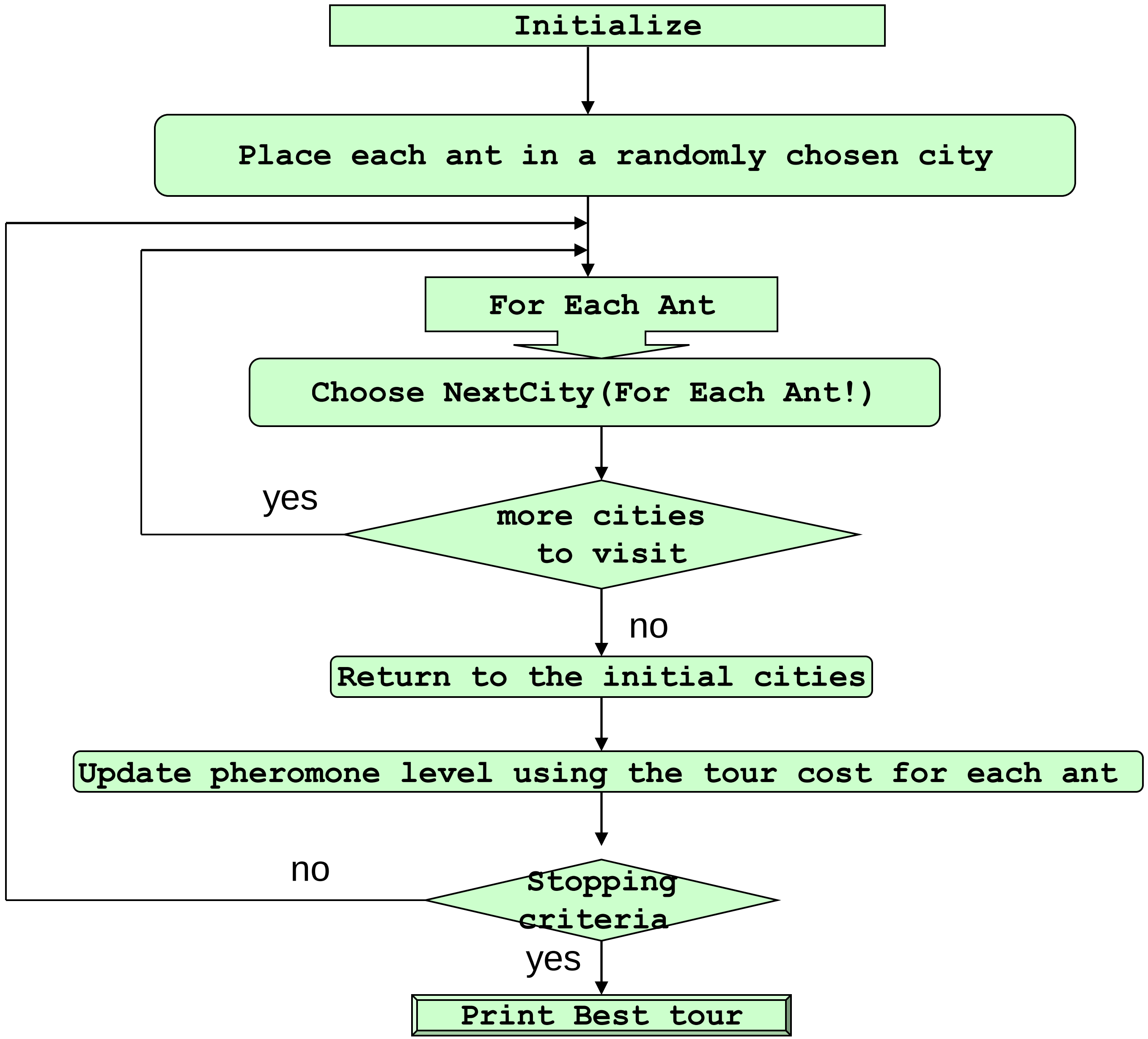
$$\Delta\tau_{ij} = \sum_{k=1}^{m} \Delta\tau_{ij}^{k}$$

$$\Delta\tau_{ij}^{k} = \frac{Q}{L_k} \; if\,(i,j) \in Tour^{k}, else\; 0$$

Marks tour = evaporation * Mark_alt + Mark_delta

$$\tau_{ij}(t+n) = \rho \cdot \tau_{ij}(t) + \Delta\tau_{ij}$$

Ant Systems Algorithm for TSP



TSP Models

- **Ant density model**
  - $\Delta\tau_{ij}^{k} = Q$ (Q: heuristic parameter)
  - Pheromone increase in trail is independent of length_{ij}
- **Ant quantity model**
  - $\Delta\tau_{ij}^{k} = Q / \text{length}_{ij}$
  - Shorter edges made more desirable by making trail inversely proportional to length_{ij}

Experimental studies

- 30 city problem, NC = 5000 cycles
- Q found to be (relatively) unimportant

	Best Parameter Set	Average Result	Best Result
Ant-density	$\alpha=1$, $\beta=5$, $\rho=0.99$	426.740	424.635
Ant-quantity	$\alpha=1$, $\beta=5$, $\rho=0.99$	427.315	426.635
Ant-cycle	$\alpha=1$, $\beta=5$, $\rho=0.5$	424.250	423.741

Parameter Sensitivity

- Bad solutions *and stagnation*
  - For high values of α the algorithm enters stagnation behavior very quickly without finding very good solutions

- Bad solutions *and no stagnation*
  - α too low, insufficient importance associated with trail

- <u>Good solutions</u>
  - α , β in the central area (1,1), (1,2), (1,5), (0.5, 5)

Exploiting ant synergy

- In original algorithm, all ants start from one town. Modify algorithm *to distribute ants amongst nodes*
  - Better than “one town” algorithm.
  - Approximately $n = m$ proved optimal.
  - Allow communication between ants, i.e. pheromone sensing ($0 < \gamma < 1$)

Exploiting ant synergy

- Initialization
  - Placing ants **uniformily** (rather than aggregated on individual nodes) resulted in superior performance.
- Employ ‘elitest’ (GAs) strategy
  - best-so-far trail is **reinforced** more than in the standard algorithm;
  - found optimal number of elitest ants.

Modifications

- New transition rules
- New pheromone update rules
- **Candidate lists** of closest cities
- Local search methods in conjunction with ACO (**Hybrid ACO**)
- **Elitism**, worst tours (pheromone removed), local search enhancement
- **Diversification**: All pheromone trail values are reinitialized if no improvement is made in S generations
- **Intensification** – keeping new best solutions in memory and replacing the current ones with them; again similar to elitism

Artificial vs. real ants

<u>Main similarities:</u>

- **Colony of individuals**
- **Exploitation of stigmergy & pheromone trail**
  - Stigmergic, indirect communication
  - Pheromone evaporation
  - Local access to information
- **Shortest path & local moves (no jumps)**
- **Stochastic state transition**

Artificial vs. real ants

Main differences:

- **Artificial ants:**
  - Live in a discrete world
  - Deposit pheromone in a problem dependent way
  - Can have extra capabilities (local search, lookahead, etc.)
  - Exploit an internal state (memory)
  - Deposit an amount of pheromone function of the solution quality
  - Can use local heuristic information

Applications of ACO

ACO algorithms have been applied to several optimization problems now.

Some of them are:

- Job-scheduling problem
- TSP
- Graph-coloring
- Vehicle Routing
- Routing in telecommunication networks
- Sequential ordering
- Multiple knapsack problem

Some references

- Dorigo M. and G. Di Caro (1999). **The Ant Colony Optimization Meta-Heuristic.** In D. Corne, M. Dorigo and F. Glover, editors, *New Ideas in Optimization*, McGraw-Hill, 11-32.
- M. Dorigo and L. M. Gambardella. Ant colonies for the traveling salesman problem. *BioSystems*, 43:73–81, 1997.
- M. Dorigo and L. M. Gambardella. Ant Colony System: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1):53–66, 1997.
- G. Di Caro and M. Dorigo. Mobile agents for adaptive routing. In H. El-Rewini, editor, *Proceedings of the 31st International Conference on System Sciences (HICSS-31)*, pages 74–83. IEEE Computer Society Press, Los Alamitos, CA, 1998.
- M. Dorigo, V. Maniezzo, and A. Colorni. The Ant System: An autocatalytic optimizing process. Technical Report 91-016 Revised, Dipartimento di Elettronica,Politecnico di Milano, Italy, 1991.
- L. M. Gambardella, ` E. D. Taillard, and G. Agazzi. MACS-VRPTW: A multiple ant colony system for vehicle routing problems with time windows. In D. Corne, M. Dorigo, and F. Glover, editors, *New Ideas in Optimization*, pages 63–76. McGraw Hill, London, UK, 1999.
- L. M. Gambardella, ` E. D. Taillard, and M. Dorigo. Ant colonies for the quadratic assignment problem. *Journal of the Operational Research Society*,50(2):167–176, 1999.
- V. Maniezzo and A. Colorni. The Ant System applied to the quadratic assignment problem. *IEEE Transactions on Data and Knowledge Engineering*, 11(5):769–778, 1999.
- Gambardella L. M., E. Taillard and M. Dorigo (1999). **Ant Colonies for the Quadratic Assignment Problem.** *Journal of the Operational Research Society*, 50:167-176.

The end