Ant Colony Optimization

Part 1: Introduction

Spring 2009

Instructor: Dr. Masoud Yaghini

Outline

- Introduction
- Real Ants
- Artificial Ants
- Simple Ant Colony Optimization (S-ACO)
- Experiments with S-ACO
- References

Introduction

Swarm Intelligence

- **Swarm intelligence** (**SI**) is artificial intelligence based on the collective behavior of decentralized, self-organized systems.
- The expression was introduced by **Gerardo Beni** and **Jing Wang** in 1989.
- The natural examples of SI includes the behaviors of certain **ants**, **honeybees**, **wasps**, **beetles**, **caterpillars**, and **termites**

Swarm Intelligence

- Example of swarm intelligence algorithms:
  - **Ant colony optimization**
  - **Particle swarm optimization**
  - **Stochastic diffusion search**
  - **Swarm robotics**

Ant Colony Optimization

- **Ant Colony Optimization (ACO)** is inspired by the foraging behavior of ant colonies
- ACO algorithms are used for solving Discrete optimization problems.
- ACO is one of the most successful examples of metaheuristic algorithms.

Ant Colony Optimization

- Examples of ACO algorithms
  - Ant System (AS)
  - Elitist Ant System (EAS)
  - Rank-Based Ant System (ASrank)
  - Min-Max Ant System (MMAS)
  - Ant Colony System (ACS)
  - Approximate Nondeterministic Tree Search (ANTS)
  - Hyper-Cube Framework

ACO Brief History

- **1989 & 1990**:
  - By **Goss et al.** & **Deneuborg et al.**
  - Experiments with Argentine ants
  - The ants prefer the shortest path from the nest to the food source
- **1991**:
  - By **Dorigo et. al.**
  - Ant System (AS) was the first ACO algorithm presented for shortest paths
- **1998**:
  - **Ant Colony Optimization** is the name given by **Dorigo** (Milan, Italy),
  - A class of algorithms whose first member was AS.

ACO Applications

Problem type	Problem name	Main references
Routing	Traveling salesman	Dorigo, Maniezzo, & Colorni (1991a,b, 1996) Dorigo (1992) Gambardella & Dorigo (1995) Dorigo & Gambardella (1997a,b) Stützle & Hoos (1997, 2000) Bullnheimer, Hartl, & Strauss (1999c) Cordón, de Viana, Herrera, & Morena (2000)
	Vehicle routing	Bullnheimer, Hartl, & Strauss (1999a,b) Gambardella, Taillard, & Agazzi (1999) Reimann, Stummer, & Doerner (2002)
	Sequential ordering	Gambardella & Dorigo (1997, 2000)

ACO Applications

Problem type	Problem name	Main references
Assignment	Quadratic assignment	Maniezzo, Colorni, & Dorigo (1994) Stützle (1997b) Maniezzo & Colorni (1999) Maniezzo (1999) Stützle & Hoos (2000)
	Graph coloring	Costa & Hertz (1997)
	Generalized assignment	Lourenço & Serra (1998, 2002)
	Frequency assignment	Maniezzo & Carbonaro (2000)
	University course timetabling	Socha, Knowles, & Sampels (2002) Socha, Sampels, & Manfrin (2003)

ACO Applications

Problem type	Problem name	Main references
Scheduling	Job shop	Colorni, Dorigo, Maniezzo, & Trubian (1994)
	Open shop	Pfahringer (1996)
	Flow shop	Stützle (1998a)
	Total tardiness	Bauer, Bullnheimer, Hartl, & Strauss (2000)
	Total weighted tardiness	den Besten, Stützle, & Dorigo (2000) Merkle & Middendorf (2000, 2003a) Gagné, Price, & Gravel (2002)
	Project scheduling	Merkle, Middendorf, & Schmeck (2000a, 2002)
	Group shop	Blum (2002a, 2003a)

ACO Applications

Problem type	Problem name	Main references
Subset	Multiple knapsack	Leguizamón & Michalewicz (1999)
	Max independent set	Leguizamón & Michalewicz (2000)
	Redundancy allocation	Liang & Smith (1999)
	Set covering	Leguizamón & Michalewicz (2000) Hadji, Rahoual, Talbi, & Bachelet (2000)
	Weight constrained graph tree partition	Cordone & Maffioli (2001)
	Arc-weighted *l*-cardinality tree	Blum & Blesa (2003)
	Maximum clique	Fenet & Solnon (2003)

ACO Applications

Problem type	Problem name	Main references
Machine learning	Classification rules	Parpinelli, Lopes, & Freitas (2002b)
	Bayesian networks	de Campos, Gámez, & Puerta (2002b)
	Fuzzy systems	Casillas, Cordón, & Herrera (2000)
Network routing	Connection-oriented network routing	Schoonderwoerd, Holland, Bruten, & Rothkrantz (1996) Schoonderwoerd, Holland, & Bruten (1997) White, Pagurek, & Oppacher (1998) Di Caro & Dorigo (1998d) Bonabeau, Henavy, Guérin, Snyers, Kuntz, & Theraulaz (1998)
	Connectionless network routing	Di Caro & Dorigo (1997, 1998c,f) Subramanian, Druschel, & Chen (1997) Heusse, Snyers, Guérin, & Kuntz (1998) van der Put (1998)
	Optical network routing	Navarro Varela, & Sinclair (1999)

Real Ants

Stigmergy

- Ant colonies, in spite of the simplicity of their individuals, present a highly structured social organization.
- As a result of this organization, ant colonies can accomplish complex.
- Ants coordinate their activities via **stigmergy**

Stigmergy

- Stigmergy is a form of **indirect communication** mediated by modifications of the environment.
  - an individual modifies the environment
  - other individuals respond to that change at a later time
- The environment mediates the communication among individuals
- A foraging ant deposits a chemical on the ground which increases the probability that other ants will follow the same path.

Pheromones

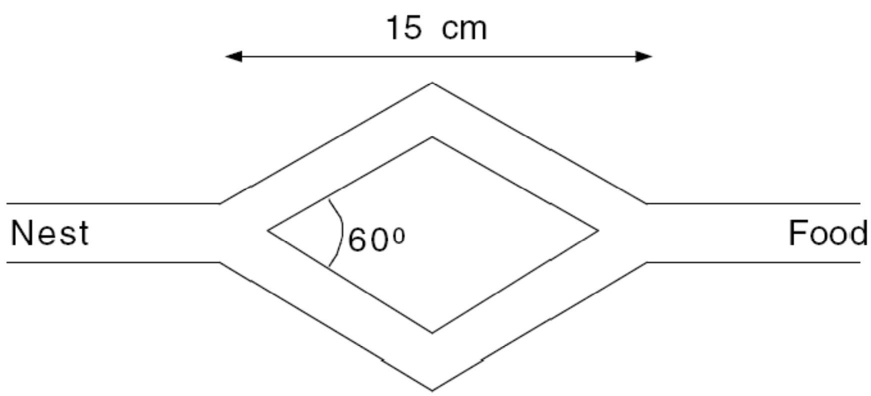
- The communication among individuals, or between individuals and the environment, is based on the use of chemicals produced by the ants.
- These chemicals are called **pheromones**.
- **Trail pheromone** is a specific type of pheromone that some ants use for marking paths on the ground, for example, paths from food sources to the nest.

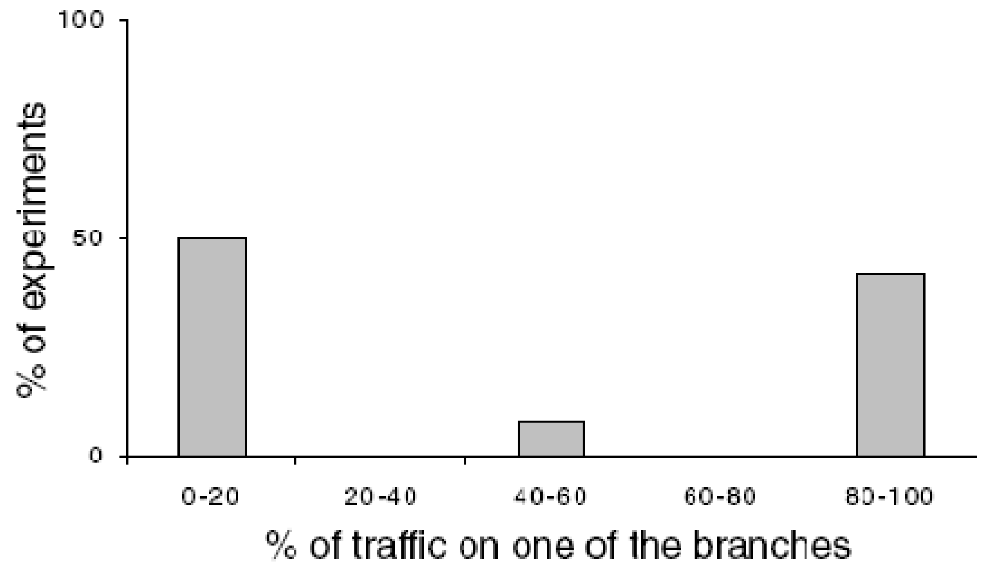
Double Bridge Experiments

- **Deneubourg** and colleagues have shown that foraging ants can find the shortest path between their nest and a food source
- They used a double bridge connecting a nest of ants and a food source.
- They ran experiments varying the length of the two branches of the double bridge.

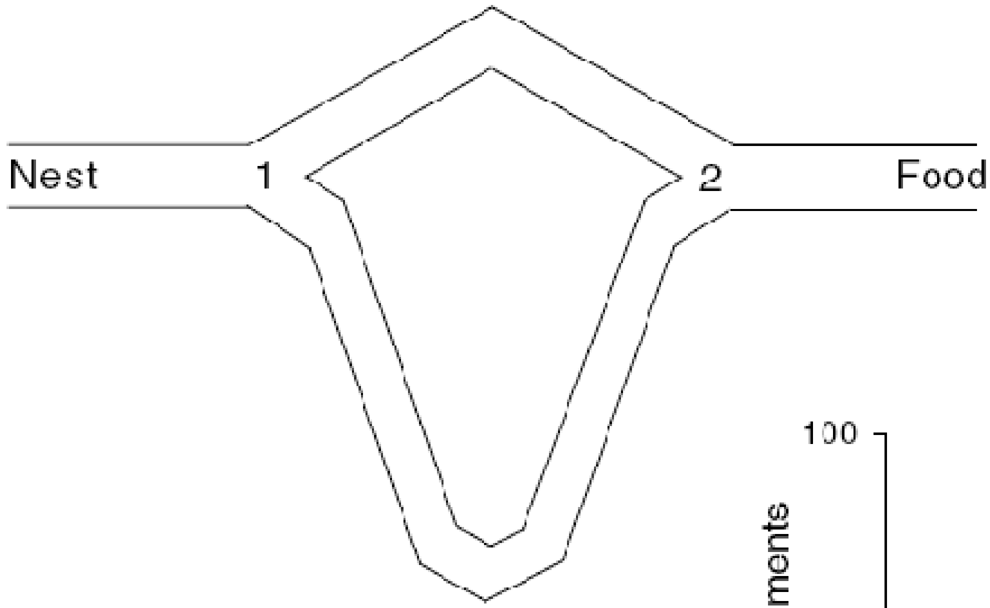
Double Bridge Experiments

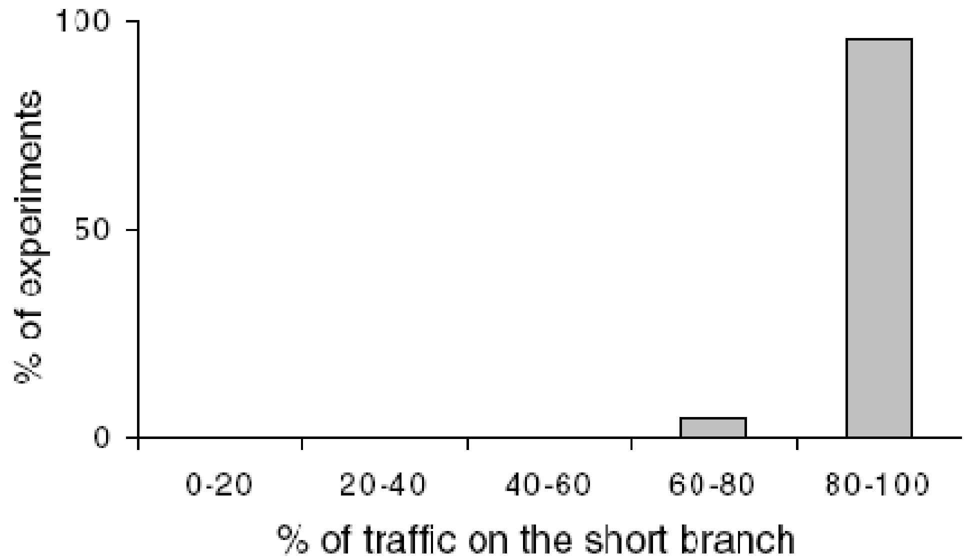
First Experiment



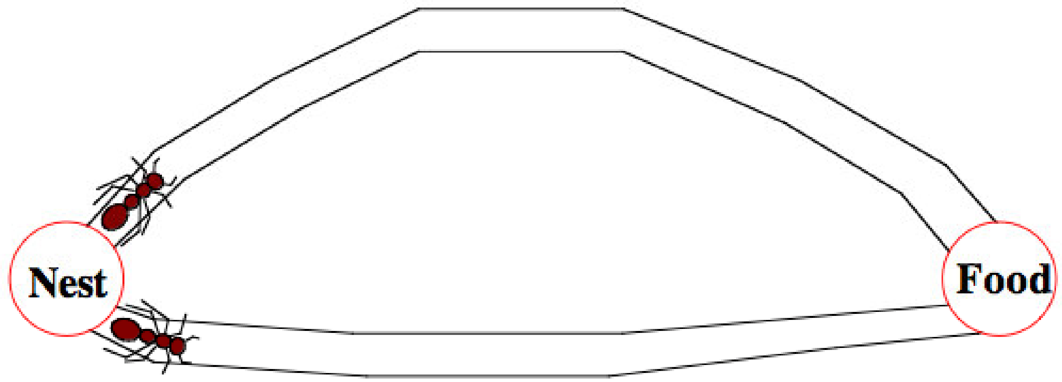


Second Experiment



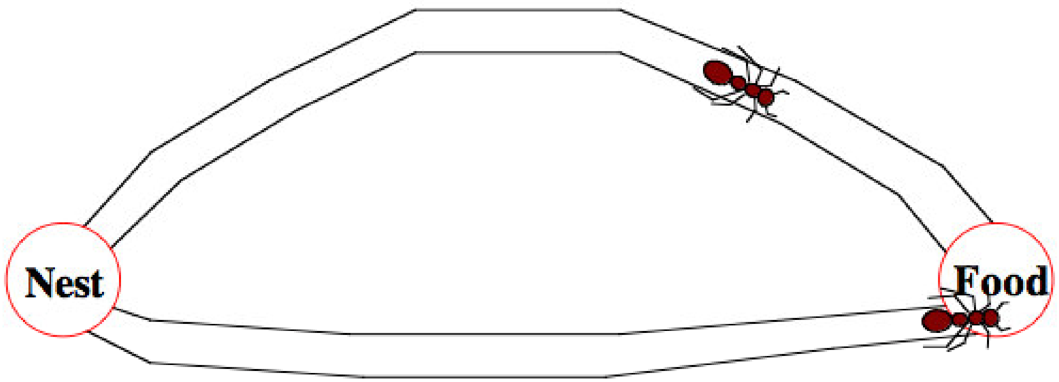


Foraging behavior of Ants



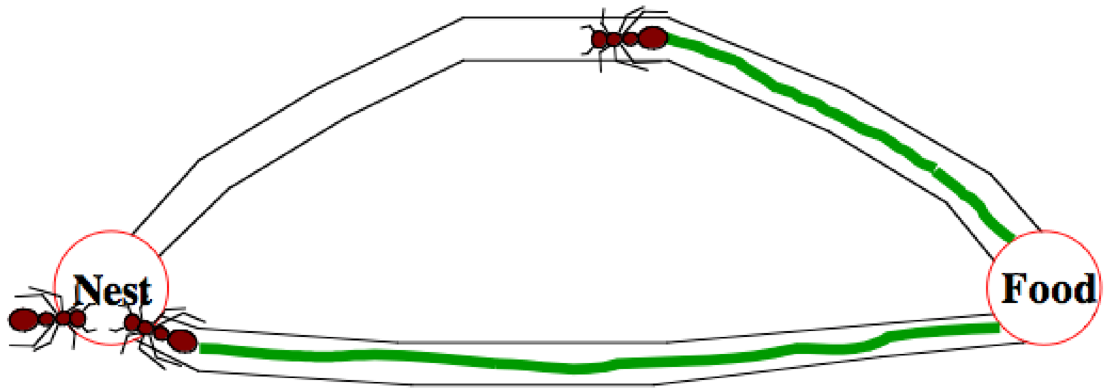
- 2 ants start with equal probability of going on either path.

Foraging behavior of Ants



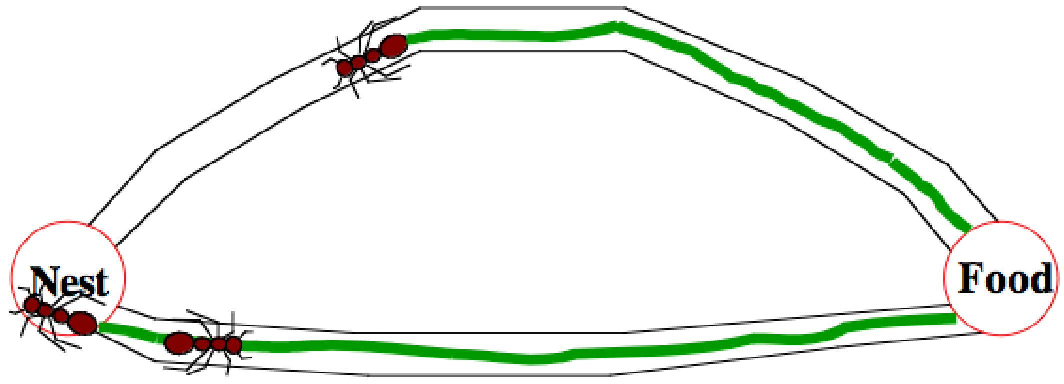
- The ant on shorter path has a shorter to-and-fro time from it's nest to the food.

Foraging behavior of Ants



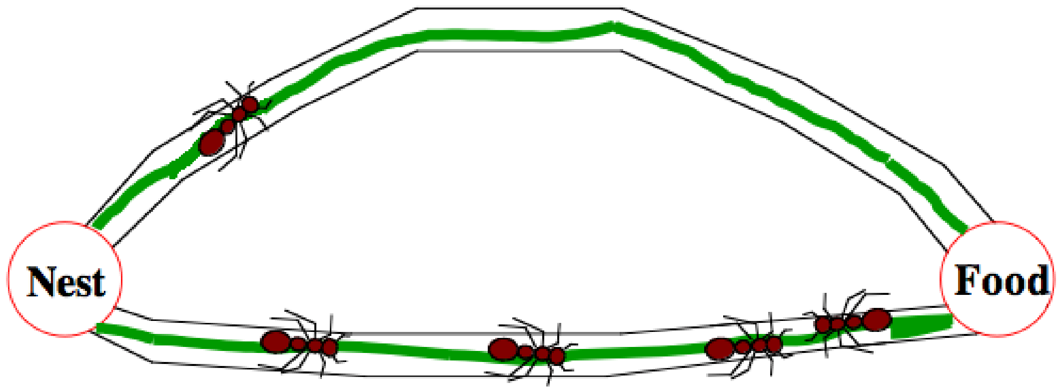
- The density of pheromone on the shorter path is higher because of 2 passes by the ant (as compared to 1 by the other).

Foraging behavior of Ants



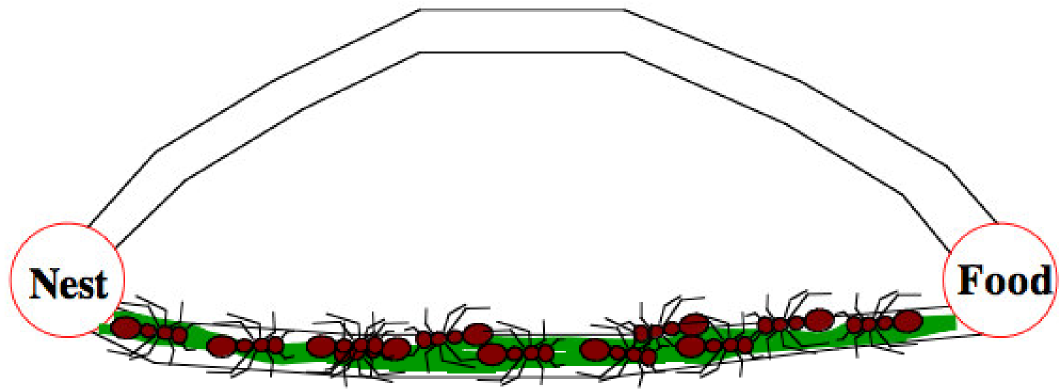
- The next ant takes the shorter route.

Foraging behavior of Ants



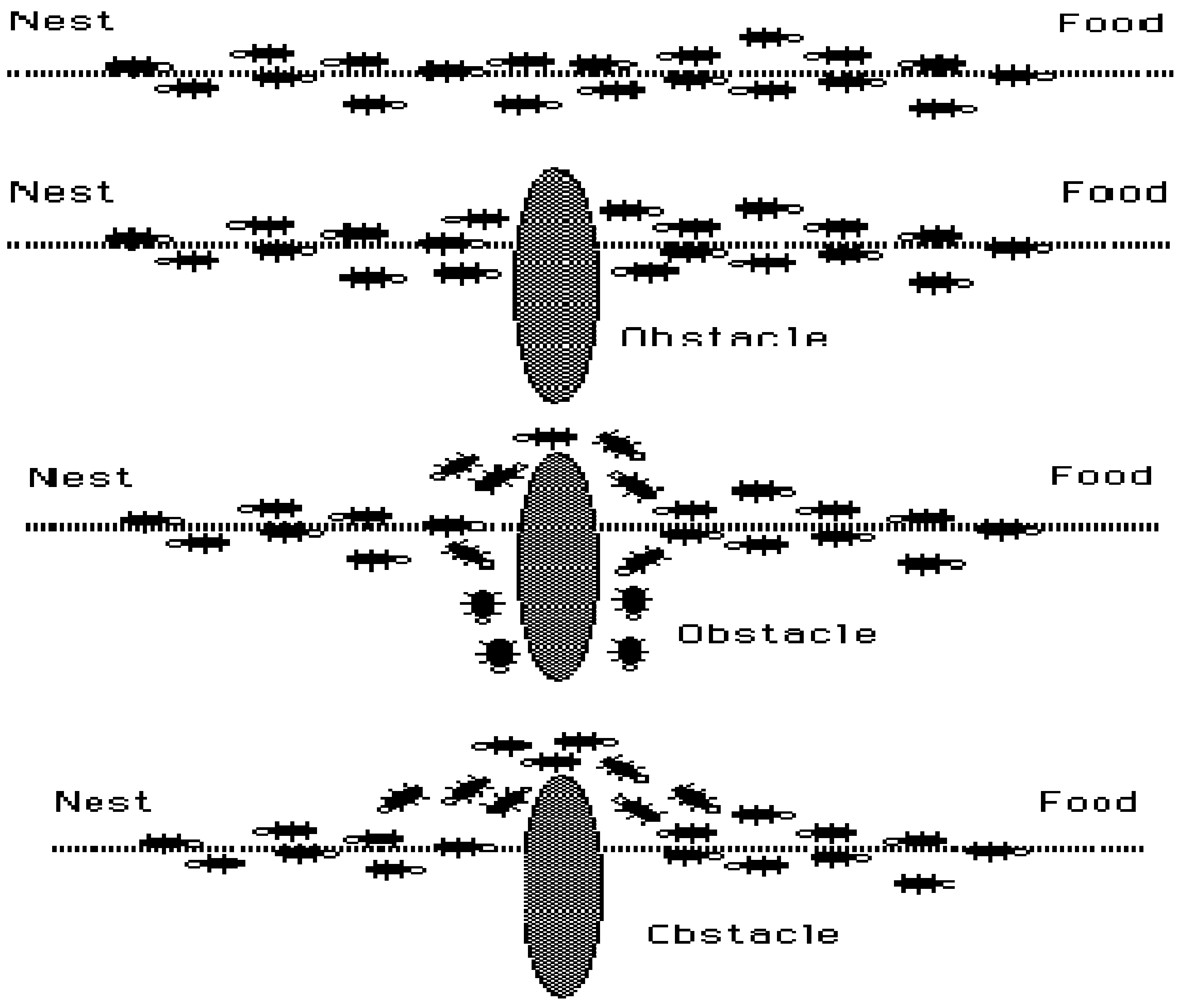
- Over many iterations, more ants begin using the path with higher pheromone, thereby further reinforcing it.

Foraging behavior of Ants



- After some time, the shorter path is almost exclusively used.

Foraging behavior of Ants



Inspiring Source of ACO

- This collective trail-laying and trail-following behavior whereby an ant is influenced by a chemical trail left by other ants is the inspiring source of ACO.

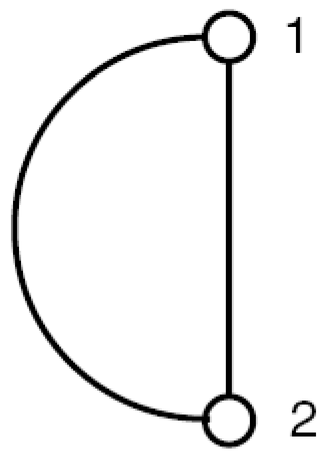
Artificial Ants

Artificial Ants

- The **double bridge experiments** show clearly that ant colonies have a built-in optimization capability
- By the use of probabilistic rules based on local information they can find the shortest path between two points in their environment.
- It is possible to design **artificial ants** that, by moving on a graph modeling the double bridge, find the shortest path between the two nodes corresponding to the nest and to the food source.

Artificial Ants

- As a first step toward the definition of artificial ants, consider this graph



- The graph consists of two nodes (1 and 2, representing the nest and the food respectively)

Artificial Ants

- The nodes are connected by a **short** and a **long** arc
- In the example the long arc is **r times** longer than the short arc, where r is an integer number.
- We assume the time to be discrete (t=1, 2, …) and that at each time step each ant moves toward a neighbor node at constant **speed of one unit of length per time unit**.

Artificial Ants

- Ants add **one unit of pheromone** to the arcs they use.
- Ants move on the graph by choosing the path probabilistically:
  - $P_{is}(t)$ is the probability for an ant located in node i at time t to choose the short path, and
  - $P_{il}(t)$ the probability to choose the long path.
- These probabilities are a function of the pheromone trails φ_{ia} that ants in node i

Artificial Ants

- The probabilities

$$p_{is}(t) = \frac{[\varphi_{is}(t)]^{\alpha}}{[\varphi_{is}(t)]^{\alpha} + [\varphi_{il}(t)]^{\alpha}}$$

$$p_{il}(t) = \frac{[\varphi_{il}(t)]^{\alpha}}{[\varphi_{is}(t)]^{\alpha} + [\varphi_{il}(t)]^{\alpha}}$$

Artificial Ants

- Trail update on the two branches is performed as follows:

$$\varphi_{is}(t) = \varphi_{is}(t-1) + p_{is}(t-1)m_i(t-1) + p_{js}(t-1)m_j(t-1),$$
$$(i = 1, j = 2; i = 2, j = 1),$$

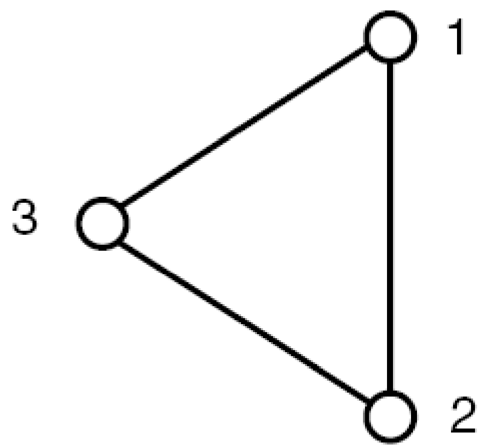
$$\varphi_{il}(t) = \varphi_{il}(t-1) + p_{il}(t-1)m_i(t-1) + p_{jl}(t-r)m_j(t-r),$$
$$(i = 1, j = 2; i = 2, j = 1),$$

- Where $m_i(t)$ the number of ants on node i at time t, is given by

$$m_i(t) = p_{js}(t-1)m_j(t-1) + p_{jl}(t-r)m_j(t-r),$$
$$(i = 1, j = 2; i = 2, j = 1).$$

Artificial Ants

- Another way of modeling:

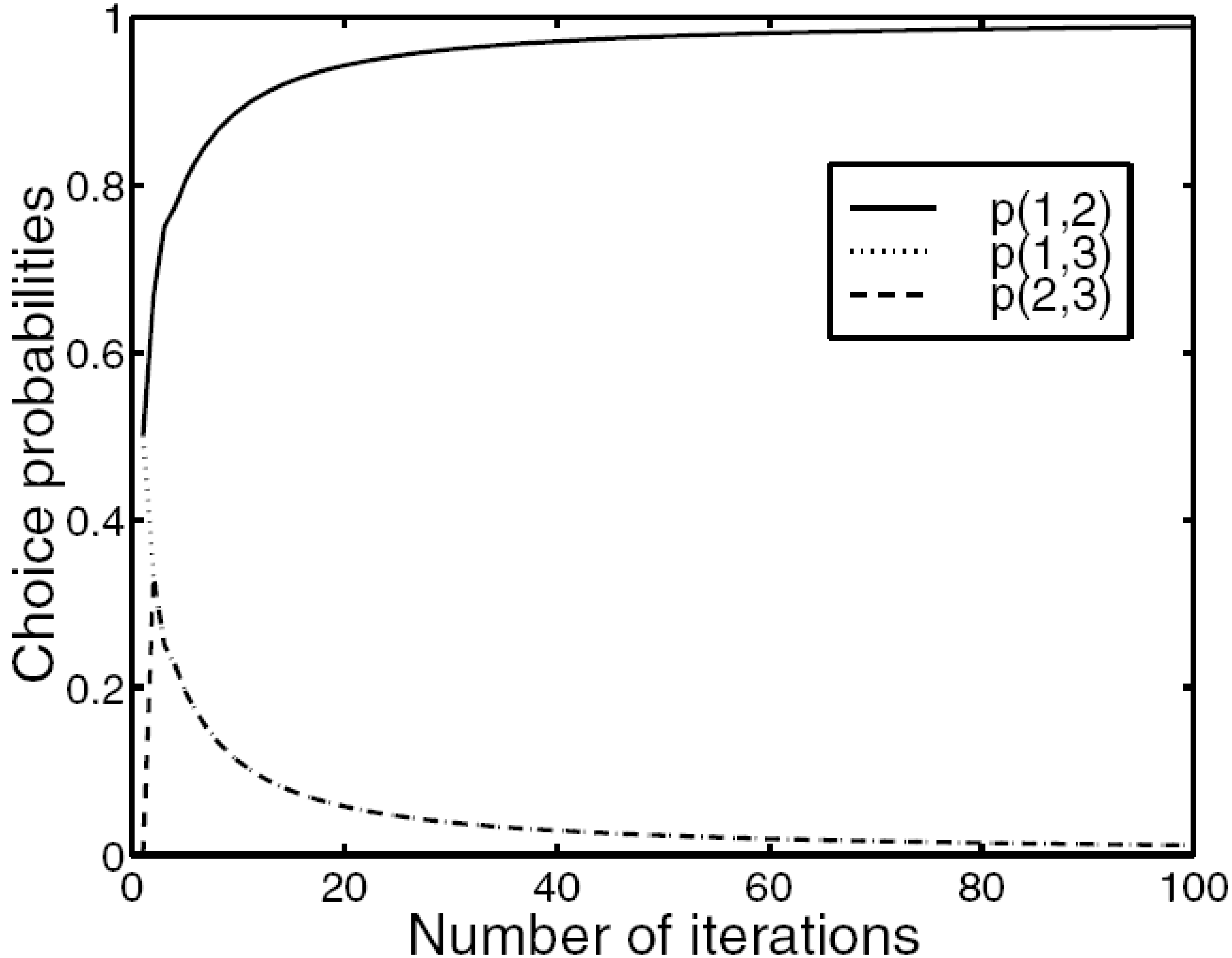


- In this model each arc of the graph has the same length, and a longer branch is represented by a sequence of arcs.
- In the figure, for example, the long branch is twice as long as the short branch.

Artificial Ants

- Pheromone updates are done with one time unit delay on each arc.
- The two models are equivalent from a computational point of view, yet the second model permits an easier algorithmic implementation when considering graphs with many nodes.
- By setting the **number of ants to 20**, the **branch length ratio** to r=2, and the **parameter α** to 2, and t=100, the system converges rapidly toward the use of the short branch.

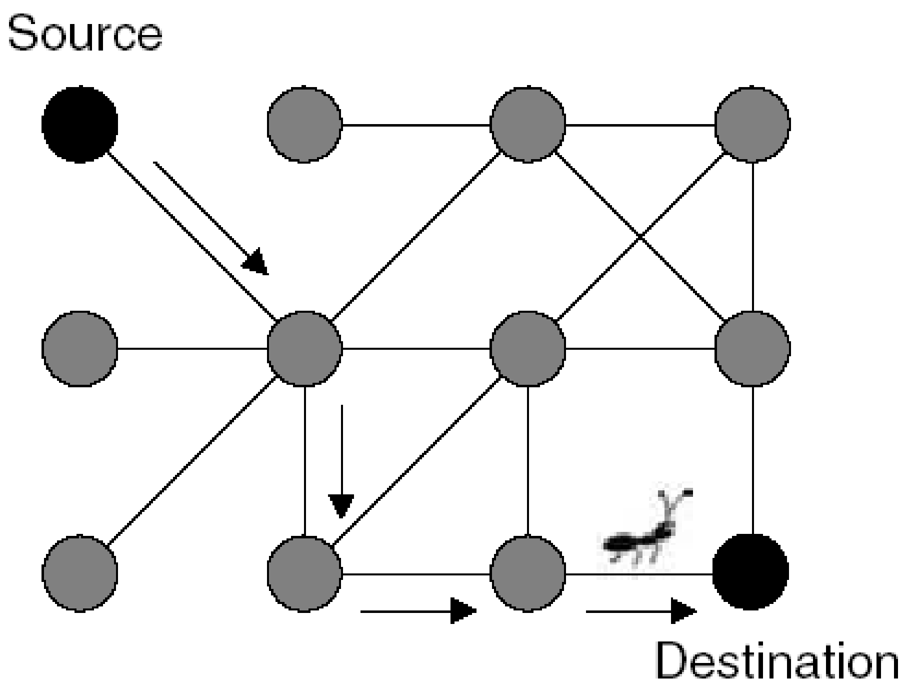
Artificial Ants



Minimum Cost Paths

Artificial Ants

- Let us consider a static, connected graph $G = (N, A)$, where N is the set of nodes and A is the set of undirected arcs connecting them.



Artificial Ants

- **Artificial ants** whose behavior is a straightforward extension of the behavior of the real ants, while building a solution, may **generate loops**.
- As a consequence of the forward pheromone trail updating mechanism, loops tend to become more and more attractive and ants can get **trapped** in them.

Artificial Ants

- Artificial ants are given a **limited form of memory** in which they can store:
  - The **paths** they have followed so far, and
  - The **cost** of the links they have traversed.
- Via the use of memory, the ants can implement a number of useful behaviors

Artificial Ants

- The artificial ants have these behaviors:
  1. Probabilistic solution construction biased by **pheromone trails**, without forward pheromone updating
  2. Deterministic backward path with **loop elimination** and with pheromone updating
  3. **Evaluation of the quality** of the solutions generated and use of the solution quality in determining the quantity of pheromone to deposit

Simple Ant Colony Optimization (S-ACO)

S-ACO

- The **simple ACO algorithm** (**S-ACO**) can be used to find a solution to the shortest path problem defined on the graph.
- A complete cycle of S-ACO:
  - **Forward ants and solution construction**
  - **Backward ants and loop elimination**
  - **Pheromone updates**
  - **Pheromone evaporation**

Forward ants and solution construction

- There are two working modes for the ants: either **forwards** or **backwards**.
- Each ant builds, starting from the source node, a solution to the problem by applying a step-by-step decision policy.
- The ants memory allows them to retrace the **path it has followed** while searching for the destination node
- Pheromones are **only deposited** in backward mode.

Forward ants and solution construction

- Assume a connected graph $G = (N, A)$.
- Associated with each edge (i, j) of the graph there is a variable τ_{ij} termed **artificial pheromone trail**.
- Every **artificial ant** is capable of "marking" an edge with pheromone and "smelling" (reading) the pheromone on the trail.
- At the beginning of the search process, a constant amount of pheromone (e.g., $\tau_{ij}=1$) is assigned to all the arcs.

Forward ants and solution construction

- An ant k located at node i uses the pheromone trail $\tau_{ij}(t)$ to compute the probability of choosing j as next node:

$$p_{ij}^{k} = \begin{cases} \dfrac{\tau_{ij}{}^{\alpha}}{\sum_{j \in N_i^k} \tau_{ij}{}^{\alpha}}, & if \;\; j \in N_i^k \\ 0, & if \;\; j \notin N_i^k \end{cases}$$

- Where
  - N_i^k is the neighborhood of ant k in node i.
  - α is a parameter that controls the relative weight of pheromone trail

The neighborhood of ant *k* in node *i*

- The neighborhood of a node *i* contains all the nodes directly connected to node *i* in the graph $G = (N, A)$, except for the predecessor of node i (i.e., the last node the ant visited before moving to *i*).
- In this way the ants avoid returning to the same node they visited immediately before node *i*.
- Only in case N_i^k is empty, which corresponds to a dead end in the graph, node *i*'s predecessor is included into N_i^k.

Forward ants and solution construction

- Ants use differences paths.
- Therefore the time step at which ants reach the destination node may differ from ant to ant.
- Ants traveling on shorter paths will reach their destinations faster.

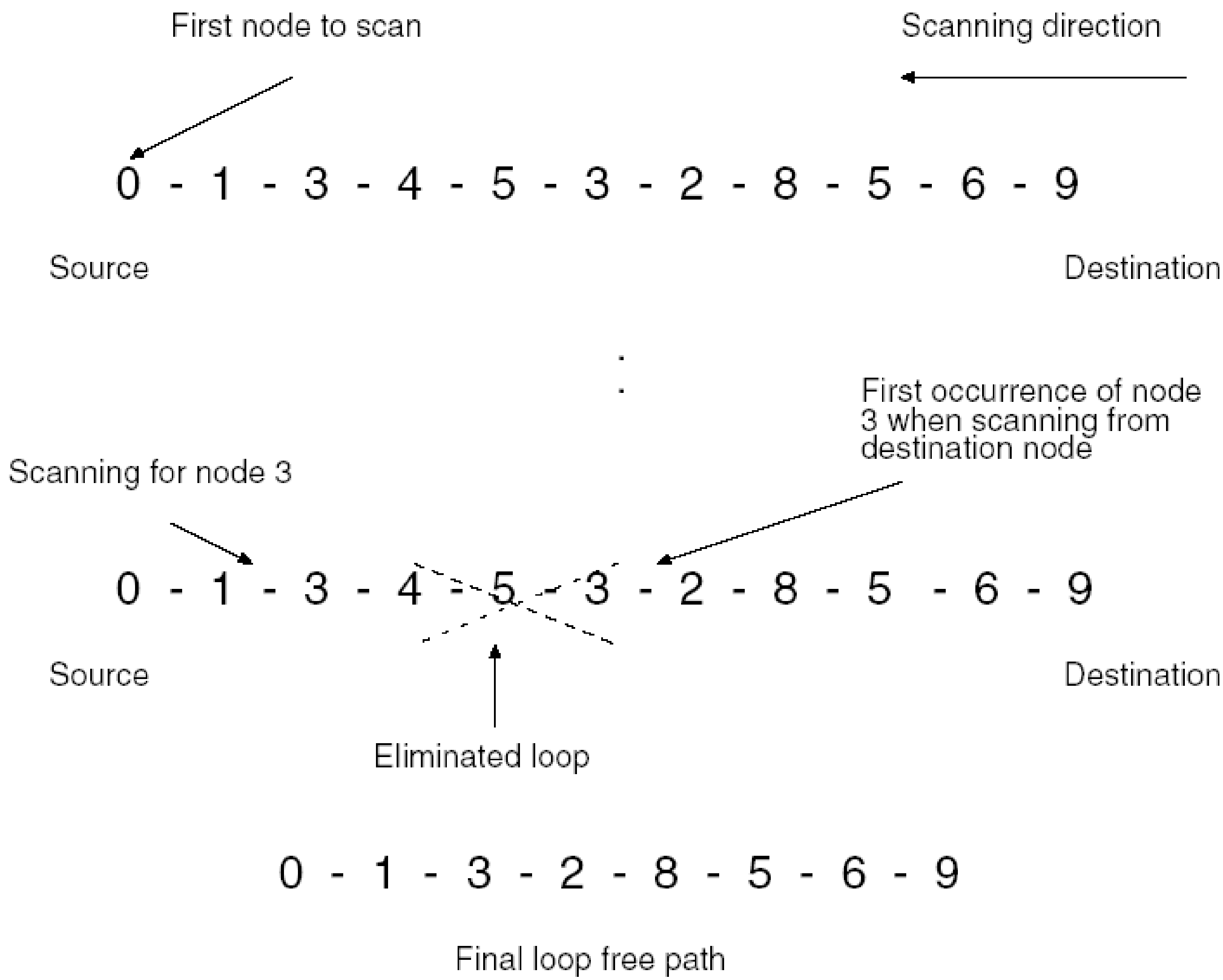
Backward ants and loop elimination

- When reaching the destination node, the ant switches from the forward mode to the backward mode
- Before moving backward on their memorized path, they eliminate any loops from it has built while searching for its destination node.
- While moving backwards, the ants leave pheromones on the arcs they traversed.

Loop elimination

- Loop elimination can be done by iteratively scanning the node identifiers position by position starting from the source node
- For the node at the *i-th* position, the path is scanned starting from the destination node until the first occurrence of the node is encountered
- If we have $j > i$, the subpath from position $i + 1$ to position j corresponds to a loop and can be eliminated.

The scanning process for loop elimination



Pheromone Update

- During its return travel to the source, the k-th ant deposits an amount $\Delta\tau^k$ of pheromone on arcs it has visited.

$$\tau_{ij} \leftarrow \tau_{ij} + \Delta\tau^k$$

- By using this rule, the probability increases that forthcoming ants will use this arc.
- An important aspect is the choice of $\Delta\tau k$.

Pheromone Update

Type of pheromone update:

- **The same constant value:**
  - The same constant value for all the ants.
  - Ants which have detected a shorter path can deposit pheromone earlier than ants traveling on a longer path.
- **Function of the solution quality**:
  - The ants evaluate the cost of the paths they have traversed.
  - The shorter paths will receive a greater deposit of pheromones.

Pheromone evaporation

- To avoid premature convergence pheromone evaporation is done
  - **Convergence**: when the probability of selecting the arcs of particular path becomes close to 1
- An evaporation rule will be tied with the pheromones, which will reduce the chance for poor quality solutions.

Pheromone evaporation

- After each ant k has moved to the next node, the pheromones evaporate by the following equation to all the arcs:

$$\tau_{ij} \leftarrow (1-p)\tau_{ij}, \quad \forall (i,j) \in A$$

- where $p \in (0,1]$ is a parameter.

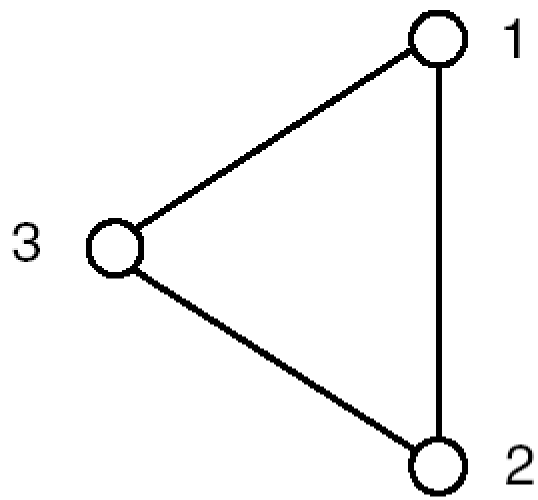
S-ACO importance aspects

- S-ACO importance aspects:
  - Number of ants
  - The Value of α
  - Pheromone evaporation rate (p)
  - Type of pheromone update

Experiments with S-ACO

First Experiments with S-ACO

- The experiments were run using the double bridge
- In this model, each arc of the graph has the same length, and a longer branch is represented by a sequence of arcs.



First Experiments

1. Run S-ACO with:
   - Different values for the number *m* of ants
   - Ants depositing a constant amount of pheromone on the visited arcs ($\Delta\tau^k$ =constant)
2. Run S-ACO With:
   - Different values for the number *m* of ants
   - Ants depositing an amount of pheromone is $\Delta\tau^k$ $=1/L^k$, where L^k is the length of ant k's path

First Experiments

- For each experiment we ran **100 trials** and each trial was stopped after each ant had moved **1000 steps** (moving from one node to the next).
- **Evaporation** was set to $p = 0$
- The parameter α was set to 2
- At the end of the trial we checked whether the pheromone trail was higher on the short or on the long path.

Results of First Experiments

- Percentage of trials in which S-ACO converged to the long path

m	1	2	4	8	16	32	64	128	256	512
without path length	50	42	26	29	24	18	3	2	1	0
with path length	18	14	8	0	0	0	0	0	0	0

- The results obtained in experiment 2 with pheromone updates based on solution quality are much better.

Influence of the parameter α

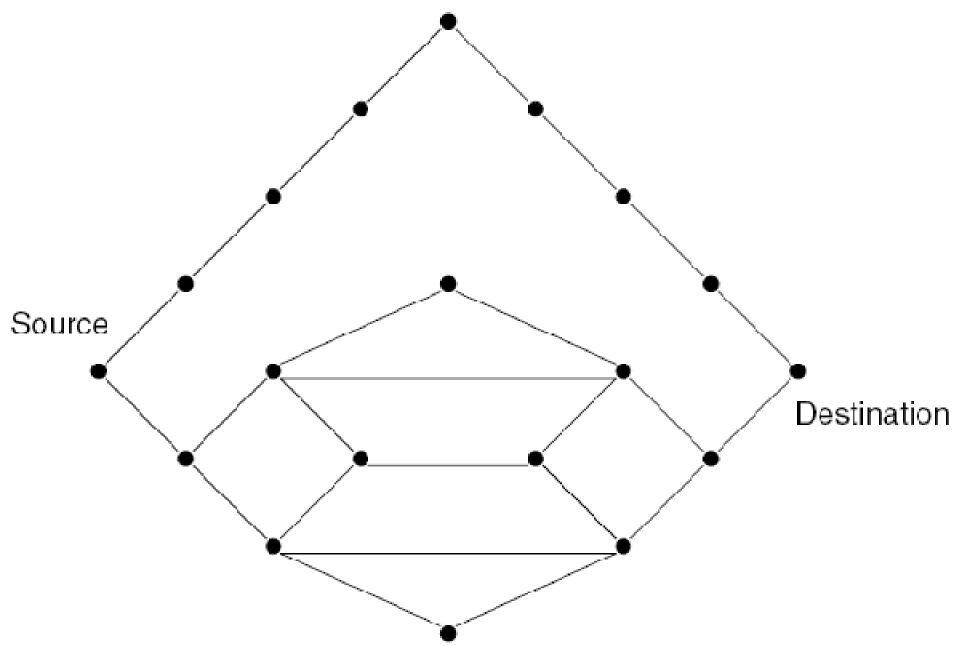
- In additional experiments, we examined the influence of the parameter α on the convergence behavior of S-ACO:
- Investigating the cases where a was changed in step sizes of **0.25 from 1 to 2**.
  - In the **first case** we found that **increasing** $\boldsymbol{\alpha}$ had a **negative effect** on the convergence behavior
  - In the **second case** the results were rather **independent** of the particular value of α.

First Experiments

- The results with S-ACO indicate that differential path length alone can be enough to let S-ACO converge to the optimal solution on small graphs
  - at the price of having to use large colony sizes, which results in long simulation times.

Second Experiments with S-ACO

- In a second set of experiments, we studied the influence that pheromone trail evaporation.
- Experiments were run using **the extended double bridge graph**



Second Experiments

- The ants deposit an amount of pheromone that is the inverse of their path length (i.e., $\Delta\tau^k = 1/L^k$)
- Before depositing pheromone, ants eliminate loops

Second Experiments

- We ran experiments with S-ACO and different settings for the evaporation rate:

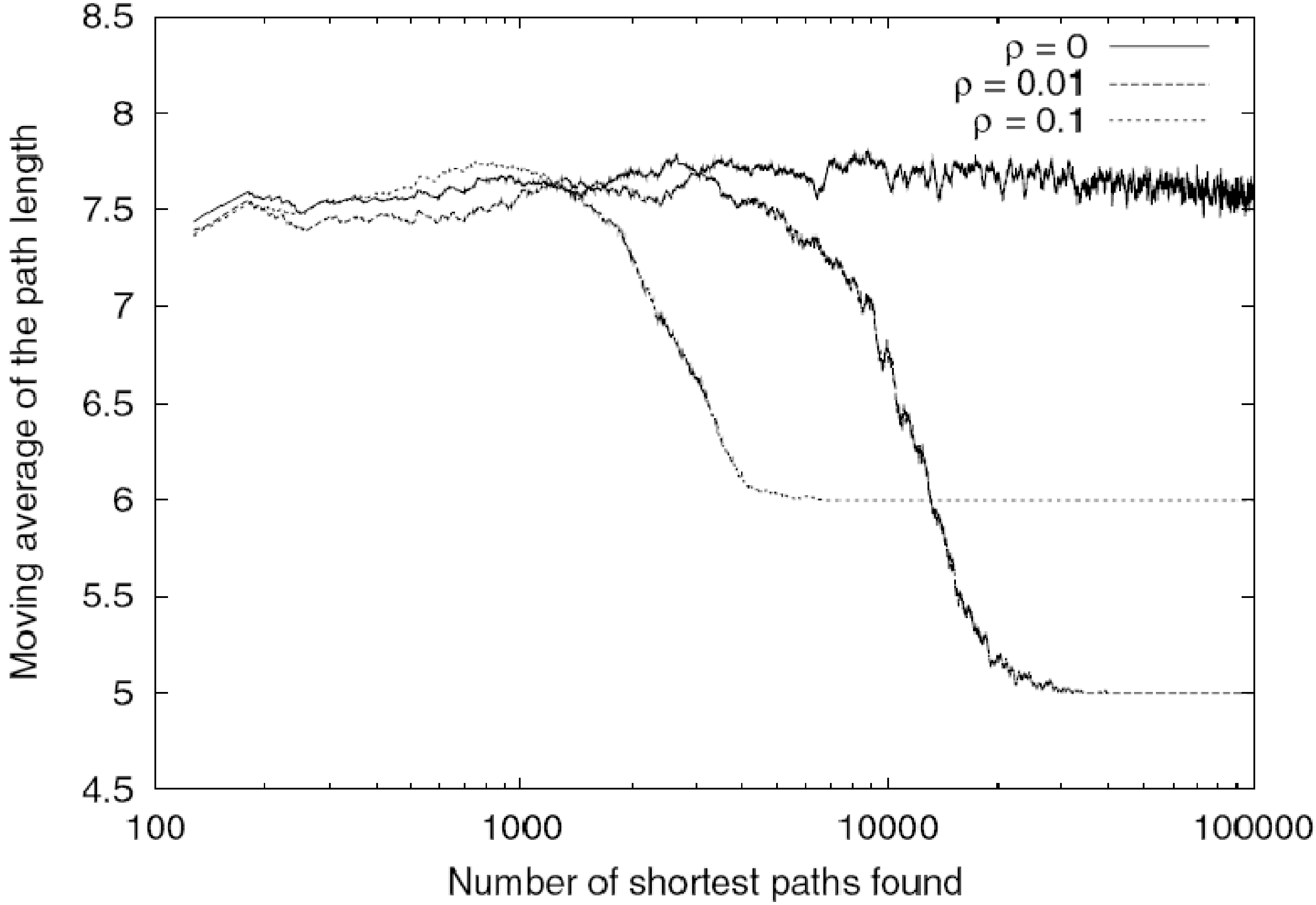
$$\rho \in \{0, 0.01, 0.1\}$$

- $\alpha = 1$ and m = 128 in all experiments.

Plot of Second Experiments

- To evaluate the behavior of the algorithm we observe the development of the path lengths found by the ants.
- We plot the moving averages of the path lengths after loop elimination (moving averages are calculated using the 4 most recent paths found by the ants).
- In the graph of figure a point is plotted each time an ant has completed a journey from the source to the destination and back

Number of shortest paths found



Pheromone Evaporation

- If p = 0, no pheromone evaporation takes place.
- An evaporation rate of p =0.1 is rather large,
  - Because evaporation takes place at each iteration of the S-ACO algorithm
  - After ten iterations, which corresponds to the smallest number of steps that an ant needs to build the shortest path and to come back to the source, roughly 65% of the pheromone on each arc evaporates,
  - While with p = 0.01 this evaporation is reduced to around 10%.

Results: No evaporation

- If **no evaporation** is used, the algorithm does **not converge**
- It can be seen by the fact that the moving average has approximately the value 7.5, which does not correspond to the length of any path
- With these parameter settings, this result typically **does not change** if the run lasts a much higher number of iterations.

Results: With Evaporation

- With pheromone evaporation, the behavior of S-ACO is significantly different.
- After a short transitory phase, S-ACO converges to a single path
- For $\rho = 0.01$ the value of shortest path is 5
- For $\rho = 0.1$ the path of length is 6

Results: Pheromone Updates

- Without pheromone updates based on solution quality, S-ACO performance is much worse.
- The algorithm converges very often to the suboptimal solution of length 8
- The larger the parameters α or ρ, the faster S-ACO converges to this suboptimal solution.

Results: Pheromone Evaporation Rate

- The pheromone evaporation rate p can be critical.
- when evaporation was set to a value that was too high, S-ACO often converged to suboptimal paths.
- For example, in fifteen trials with p set to 0.2, S-ACO converged:
  - once to a path of length 8,
  - once to a path of length 7, and
  - twice to a path of length 6.
- Setting p to 0.01 S-ACO converged to the shortest path in all trials.

Results: Values of α

- Large values of α generally result in a worse behavior of S-ACO
- Because they excessively emphasize the initial random fluctuations.

References

References

- M. Dorigo and T. Stützle. **Ant Colony Optimization**, MIT Press, Cambridge, 2004.

The End