Swarm Intelligence

Swarm intelligence (群體智能)

- Origins in Artificial Life (Alife) Research
  1. ALife studies how computational techniques can help when studying biological phenomena
  2. ALife studies how biological techniques can help out with computational problems

- Two main Swarm Intelligence based methods
  - Particle Swarm Optimization (PSO; 粒子群優化)
  - Ant Colony Optimization (ACO; 蟻群優化)

Swarm Intelligence

- **Swarm Intelligence (SI) 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.

- **SI provides a basis with which it is possible to explore collective (or distributed) problem solving without centralized control or the provision of a global model.**

- **Leverage the power of complex adaptive systems to solve difficult non-linear stochastic problems**

Swarm Intelligence

- **Characteristics of a swarm:**
  - **Distributed, no central control or data source;**
  - **Limited communication**
  - **No (explicit) model of the environment;**
  - **Perception of environment (sensing)**
  - **Ability to react to environment changes.**

Swarm Intelligence

- Social interactions (locally shared knowledge) provides the basis for unguided problem solving

- The efficiency of the effort is related to but not dependent upon
  the degree or connectedness of the network
  and the number of interacting agents

Swarm Intelligence

- Robust exemplars of problem-solving in Nature
  - Survival in stochastic hostile environment
  - Social interaction creates complex behaviors
  - Behaviors modified by dynamic environment.

- Emergent behavior observed in:
  - Bacteria, immune system, ants, birds
  - And other social animals

Ant Colony Optimization

Ant Colony Optimization

Ant colonies for the traveling salesman problem

Dorigo M, Gambardella LM. TR/IRIDIA, 1996-3, 1996.

Christian Blum. Ant colony optimization : Introduction and recent trends. Physics of Life Reviews, v.2, p.353-373. 2005.

Outline

- **Introduction**
  - **Background**
  - **Natural behavior of ants**
  - **Concept/System**
  - **Applications**
- **First Algorithm: Ant System**
  - **Ant Colony Optimization**
  - **Applications: A simple TSP example**
- **Result**

Introduction

Background

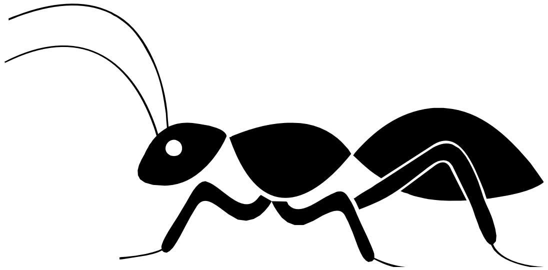
- Discrete optimization problems difficult to solve

- Popular "Soft computing techniques" developed in several decades:
  - Genetic algorithms (GAs)
    - based on natural selection and genetics
  - Ant Colony Optimization (ACO)
    - modeling ant colony behavior

Background, cont.

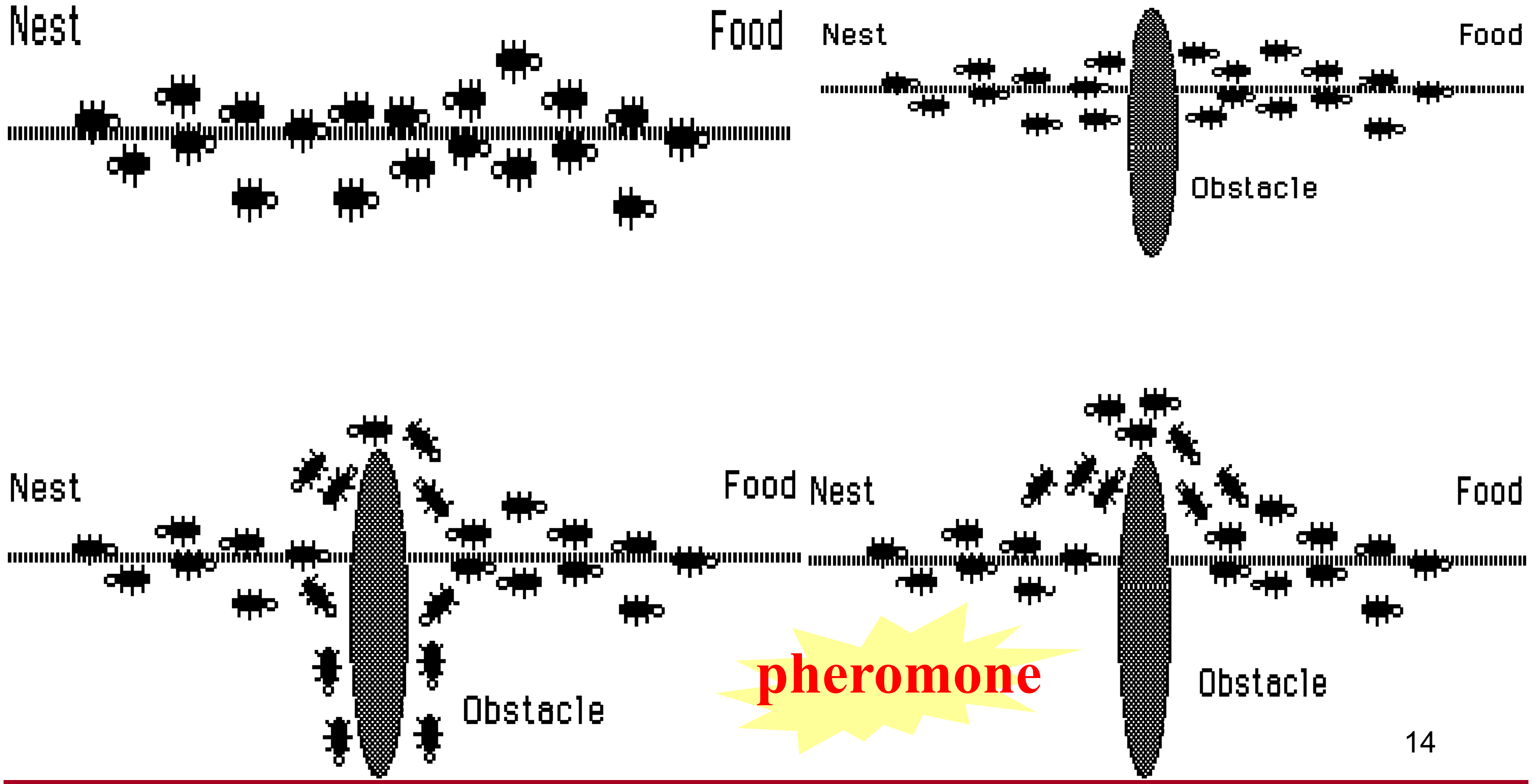
- Optimization Technique
  Proposed by Marco Dorigo in the early ’90

- Often applied to
  TSP (Travelling Salesman Problem):
  shortest path between n nodes

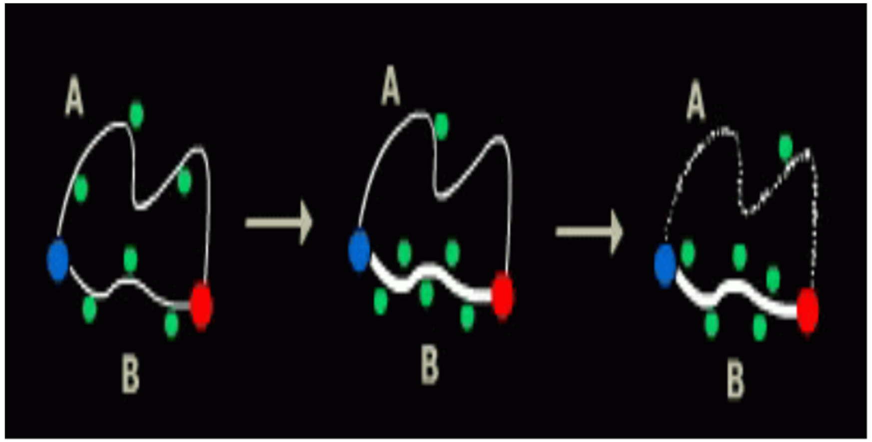


Natural Behavior of Ants

Natural Ants



Natural ants: How do they do it?

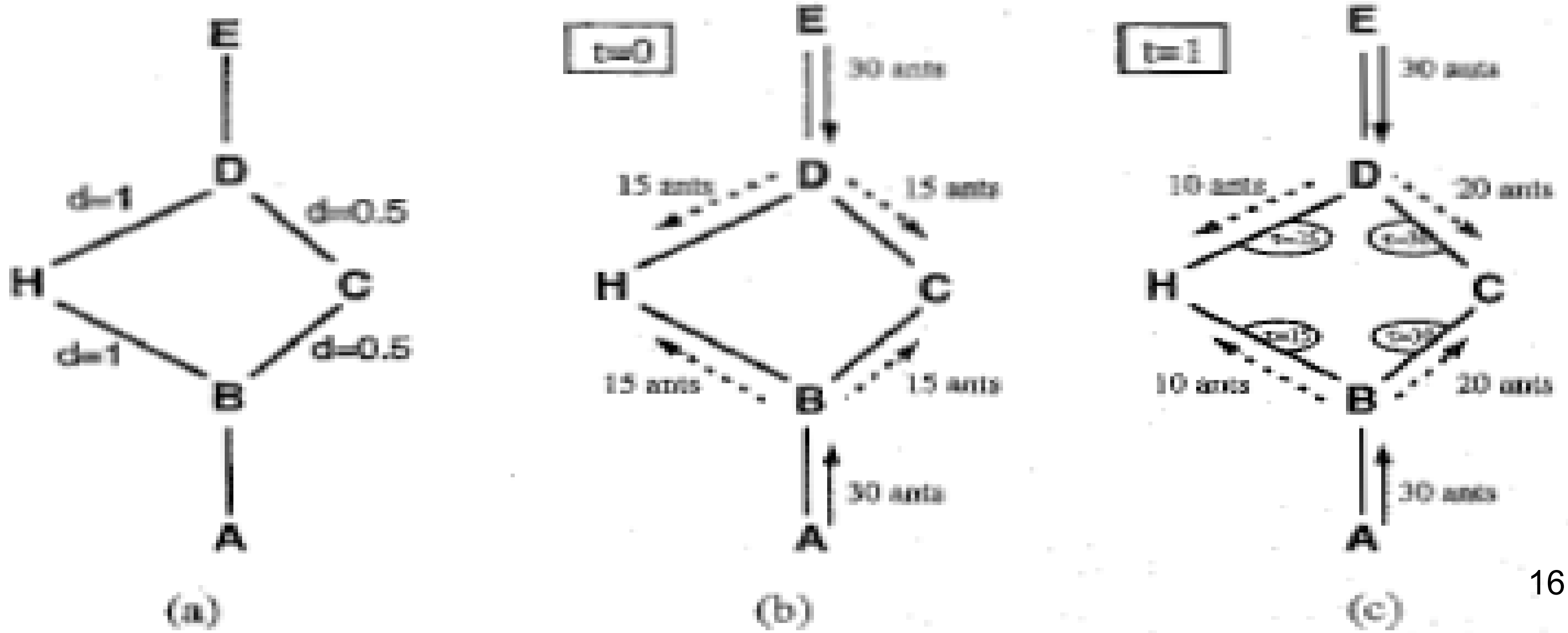


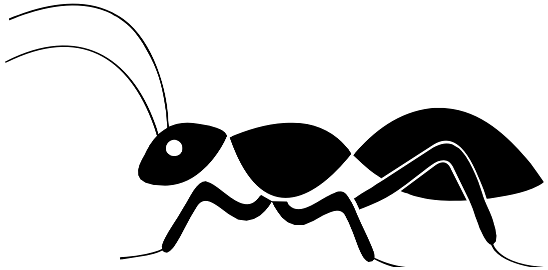
- **Since the route B is** shorter, **the ants on this path will complete the** travel more times **and thereby** lay more pheromone **over it.**

- **The** pheromone concentration **on trail B will** increase **at a higher rate than on A, and soon the ants on route A will** choose to follow route B
- **Since most ants will no longer travel on route A, and since the pheromone is volatile, trail A will start** evaporating
- **Only the** shortest route will remain**!**

An example
with artificial ants

- (a) The initial graph with distances.
- (b) At time $t = 0$ there is no trail on the graph edges.
- (c) At time $t = 1$ trail is stronger on shorter edges.





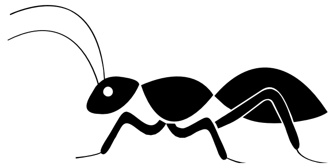
ACO Concept/System

ACO Concept

- Ants (blind) navigate from nest to food source
- Shortest path is discovered
  via pheromone trails (費洛蒙足跡)
  - each ant moves at random
  - pheromone is deposited on path
  - ants detect lead ant's path, inclined to follow
  - more pheromone on path increases probability of path being followed

ACO System

- Starting node selected at random
- Path selected at random
  - based on amount of "trail" present on possible paths from starting node
  - higher probability for paths with more "trail"
- Ant reaches next node, selects next path
- Continues until reaches starting node



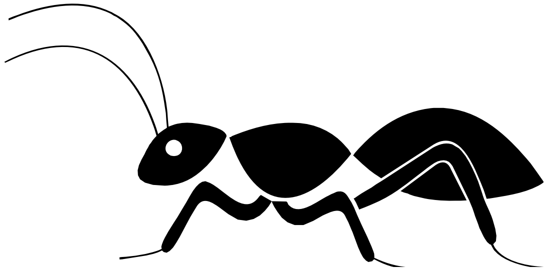
- Finished "tour" is a solution
- A completed tour is analyzed for optimality
- "Trail" amount adjusted to favor better solutions
  - better solutions receive more trail
  - worse solutions receive less trail
  - higher probability of ant selecting path that is part of a better-performing tou
- New cycle is performed
- Repeated until most ants select the same tour on every cycle (convergence to solution)

ACO System, cont.

- **Algorithm in Pseudocode:**
  - Initialize **Trail**
  - Do While **(Stopping Criteria Not Satisfied) – Cycle Loop**
    - Do Until **(Each Ant Completes a Tour) – Tour Loop**
      - **Local Trail Update**
    - End Do
    - **Analyze Tours**
    - **Global Trail Update**
  - End Do

- Can be used for both Static and Dynamic Combinatorial optimization problems

- Convergence is guaranteed, although the speed is unknown
  - Value
  - Solution



Algorithm

The Algorithm

- **Ant Colony Algorithms are typically used to solve minimum cost problems.**
- **We may usually have *N* nodes and *A* undirected arcs**
- **There are two working modes for the ants: either forwards or backwards.**
- **Pheromones are only deposited in backward mode. (so that we know how good the path was to update its trail)**

The Algorithm

- The ants memory allows them
  to retrace the path it has followed
  while searching for the destination node

- Before moving backward on their memorized path,
  they eliminate any loops from it.
  While moving backwards,
  the ants leave pheromones on the arcs they traversed.

The Algorithm

1. The ants evaluate the cost of the paths they have traversed.

2. The shorter paths will receive a greater deposit of pheromones.

3. An evaporation rule will be tied with the pheromones, which will reduce the chance for poor quality solutions.

1. Evaluate the cost of the paths they have traversed

- **At the beginning of the search process, a constant amount of pheromone is assigned to all arcs.**

- **When located at a node *i* an ant *k* uses the pheromone trail to compute the probability of choosing *j* as the next node:**

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

where N_i^k is the neighborhood of ant *k* when in node *i*.

2. The shorter paths will receive a greater deposit of pheromones

- When the arc (i,j) is traversed , the pheromone value changes as follows:

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

- By using this rule, the probability increases that forthcoming ants will use this arc.

3. Evaporation rule

- **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.**

- **An iteration is a complete cycle involving**
  - **ants' movement**
  - **pheromone evaporation**
  - **pheromone deposit**

End of First Run

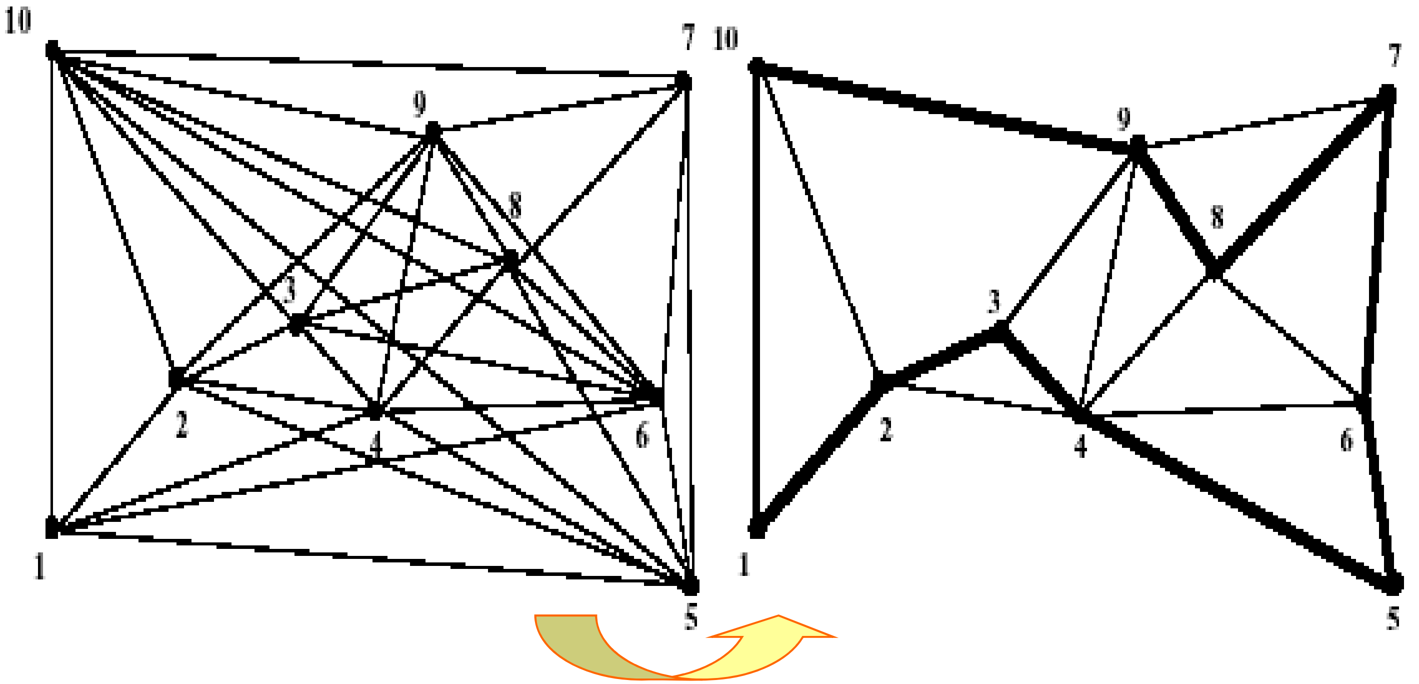
Save Best Tour (Sequence and length)

All ants die

New ants are born

Stopping Criteria

- **Stagnation**
- **Max Iterations**



Behavior for different combinations parameters

- **Bad solutions and stagnation-(by ∅)**
- **Bad solutions and no stagnation- (by ∞)**
- **Good solutions- (by ●)**

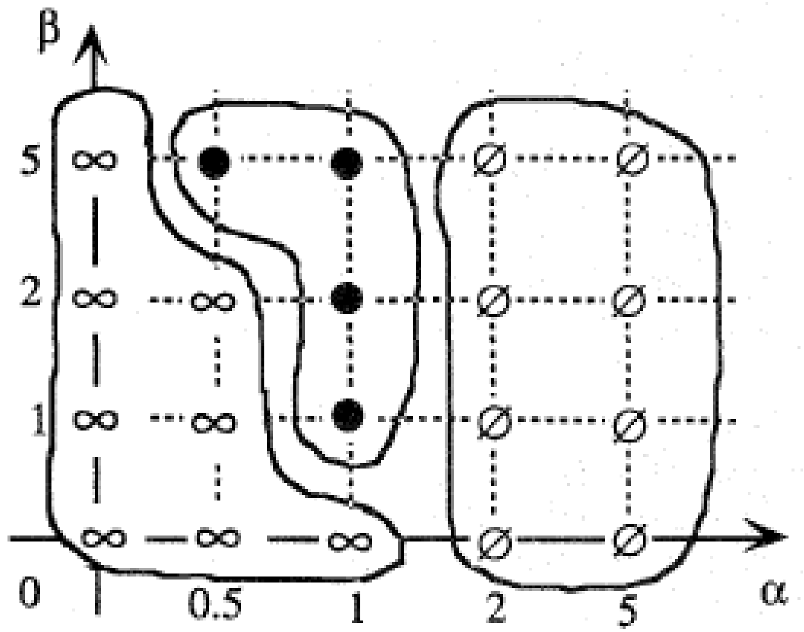


Fig. 8. *Ant-cycle* behavior for different combinations of α–β parameters.

Steps for Solving a Problem by ACO

1. **Represent the problem in the form of**
   sets of *components* and *transitions*,
   or by a set of weighted *graphs*,
   on which ants can build solutions

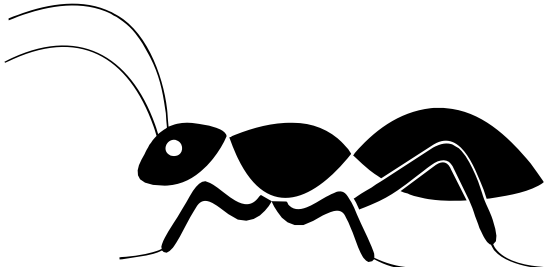
2. **Define the meaning of the *pheromone trails***

3. **Define the *heuristic preference* for the ant**
   while constructing a solution

4. **If possible, implement an efficient *local search* algorithm**
   for the problem to be solved.

5. **Choose a specific ACO algorithm**
   and apply to problem being solved

6. **Tune the parameter of the ACO algorithm.**

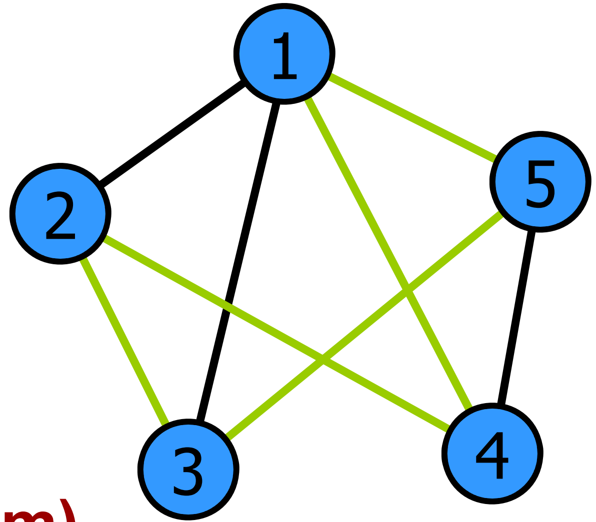


Applications

Applications

Efficiently Solves NP-hard Problems

- **Routing**
  - TSP (Traveling Salesman Problem)
  - Vehicle Routing
  - Sequential Ordering



- **Assignment**
  - QAP (Quadratic Assignment Problem)
  - Graph Coloring
  - Generalized Assignment
  - Frequency Assignment
  - University Course Time Scheduling

Applications

- **Scheduling**
  - Job Shop
  - Open Shop
  - Flow Shop
  - Total tardiness (weighted/non-weighted)
  - Project Scheduling
  - Group Shop
- **Subset**
  - Multi-Knapsack
  - Max Independent Set
  - Redundancy Allocation
  - Set Covering
  - Weight Constrained Graph Tree partition
  - Arc-weighted L cardinality tree
  - Maximum Clique

Applications

- Other
  - Shortest Common Sequence
  - Constraint Satisfaction
  - 2D-HP protein folding
  - Bin Packing
- Machine Learning
  - Classification Rules
  - Bayesian networks
  - Fuzzy systems
- Network Routing
  - Connection oriented network routing
  - Connection network routing
  - Optical network routing

Application to TSP

Traveling Salesperson Problem

- Famous NP-Hard Optimization Problem
- Given a fully connected, symmetric G(V,E) with known edge costs, find the minimum cost tour.
- Artificial ants move from vertex to vertex to order to find the minimum cost tour using only pheromone mediated trails.

Traveling Salesperson Problem

- The three main ideas that this ant colony algorithm has adopted from real ant colonies are:
  - The ants have a probabilistic preference for paths with high pheromone value
  - Shorter paths tend to have a higher rate of growth in pheromone value
  - It uses an indirect communication system through pheromone in edges

Traveling Salesperson Problem

- **Ants select the next vertex based on a weighted probability function based on two factors:**
  - **The number of edges and the associated cost**
  - **The trail (pheromone) left behind by other ant agents.**

- **Each agent modifies the environment in two different ways :**
  - Local trail updating:
    As the ant moves between cities it updates the amount of pheromone on the edge
  - Global trail updating:
    When all ants have completed a tour the ant that found the shortest route updates the edges in its path

Traveling Salesperson Problem

- Local Updating is used
  to avoid very strong pheromone edges
  and hence increase exploration
  (and hopefully avoid locally optimal solutions).

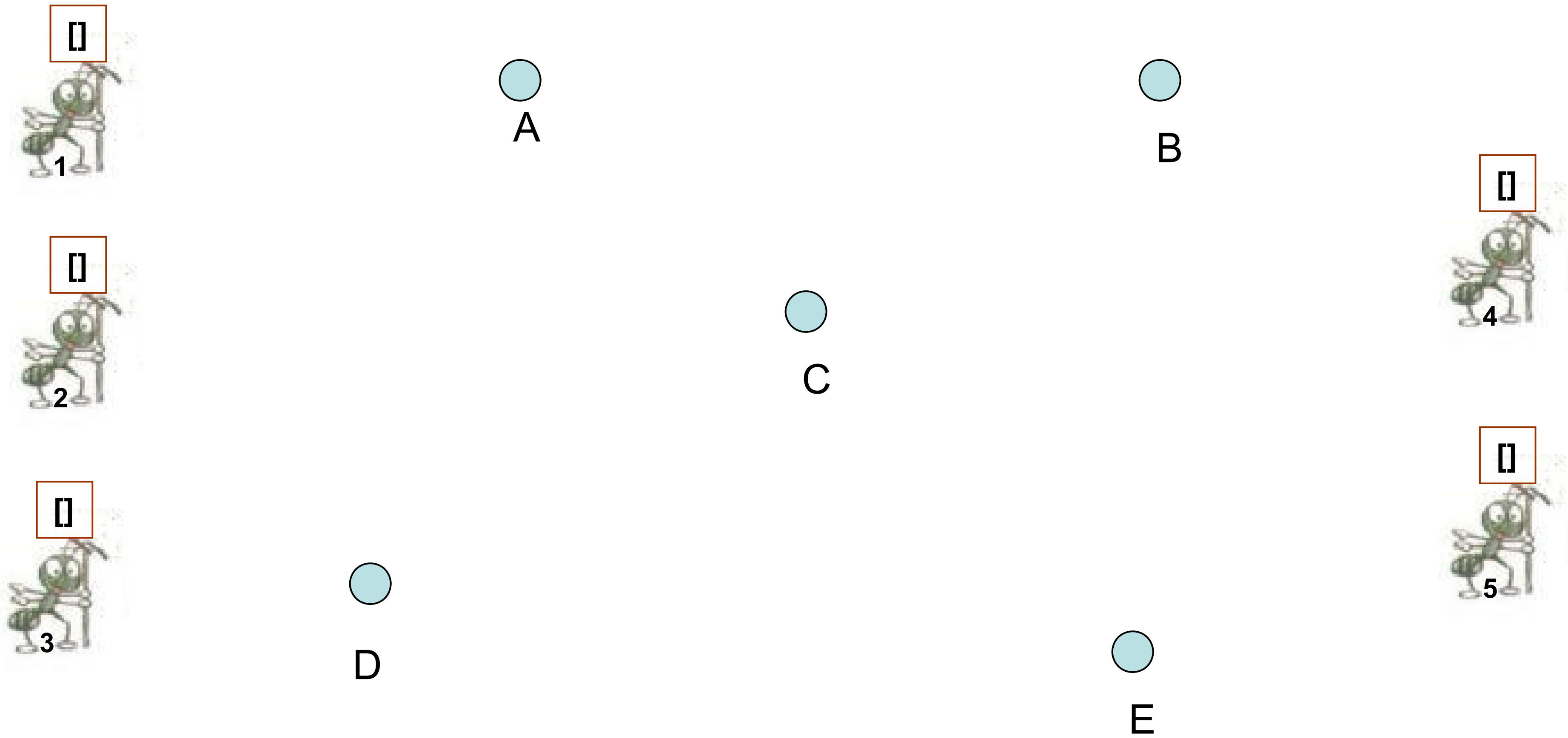
- The Global Updating function
  gives the shortest path higher reinforcement
  by increasing the amount of pheromone
  on the edges of the shortest path.

Reference

Ant colonies for the traveling salesman problem

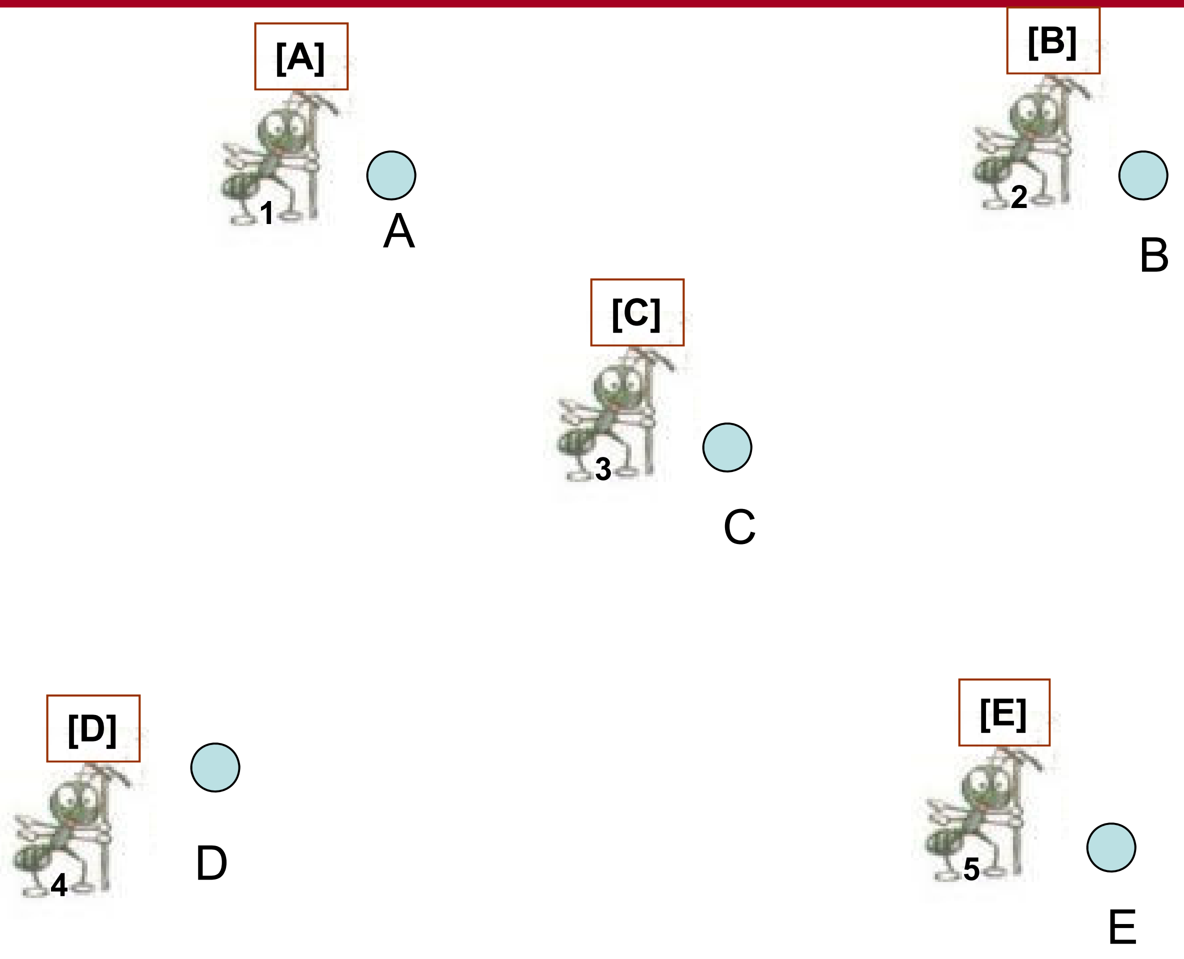
Dorigo M, Gambardella LM. TR/IRIDIA, 1996-3, 1996.

A simple TSP example

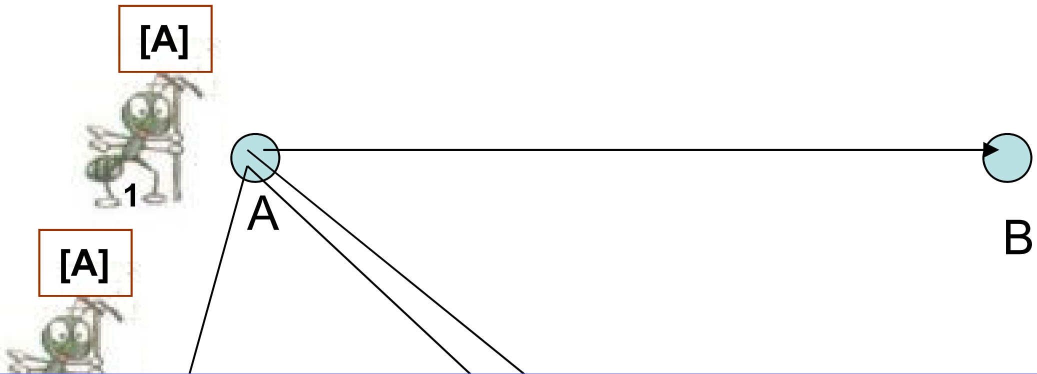


$d_{AB}=100; d_{BC}=60\ldots; d_{DE}=150$

Iteration 1



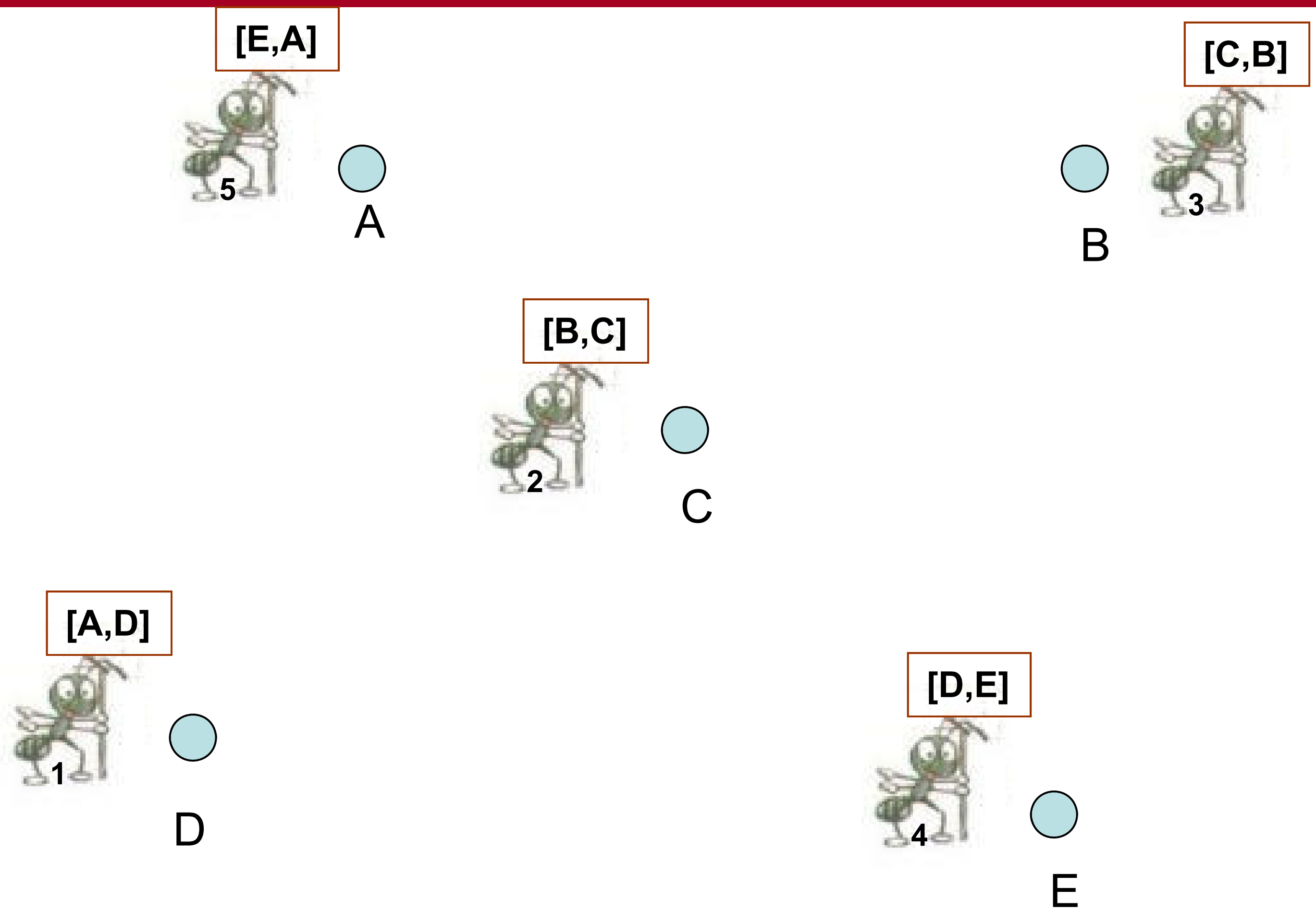
How to build next sub-solution?



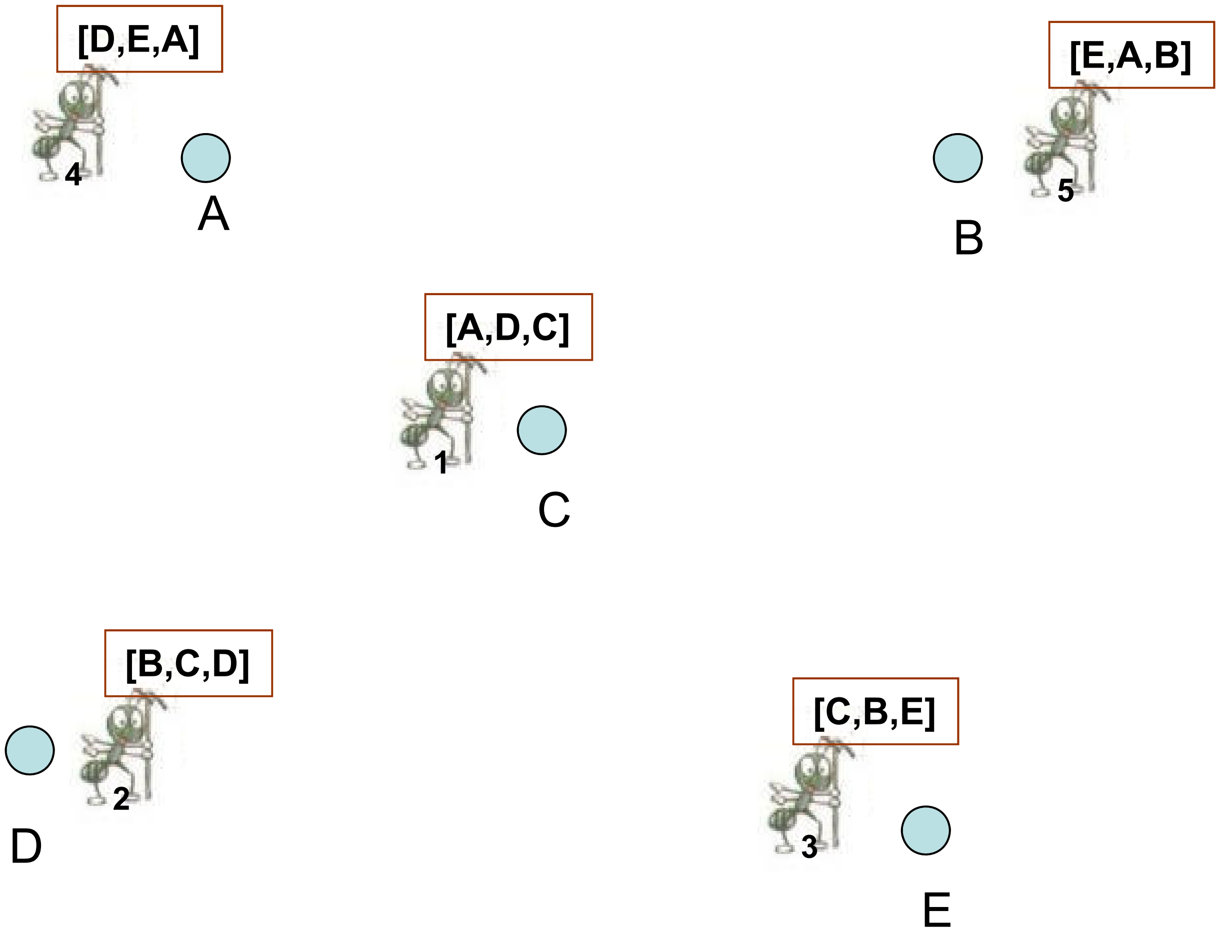
$$p_{ij}^{k}(t)=\begin{cases}\dfrac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}]^{\beta}}{\sum\limits_{k\in allowed_k}[\tau_{ik}(t)]^{\alpha}[\eta_{ik}]^{\beta}} & \text{if } j\in \text{allowed}_k\\ 0 & \text{otherwise}\end{cases}$$

E

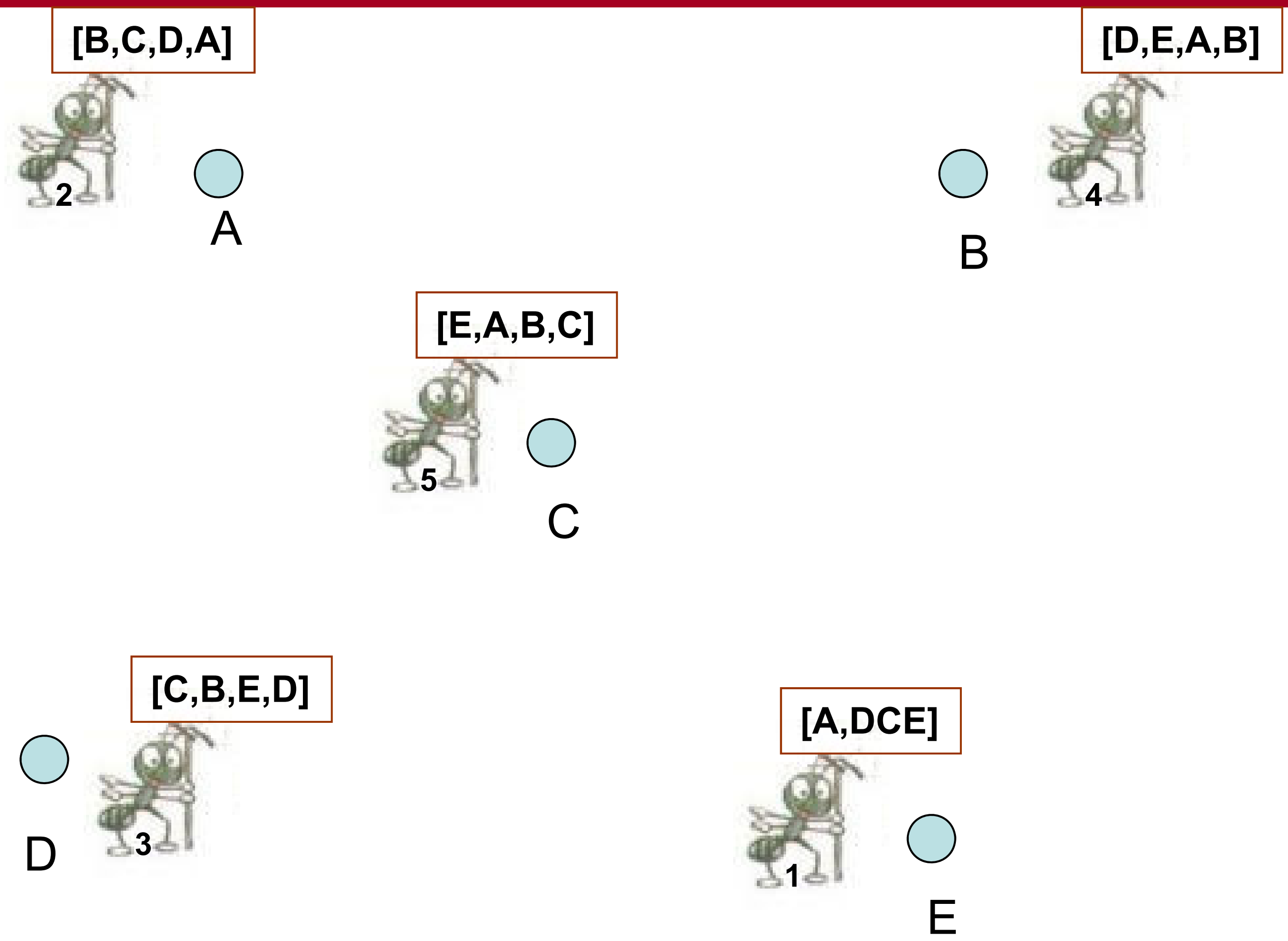
Iteration 2



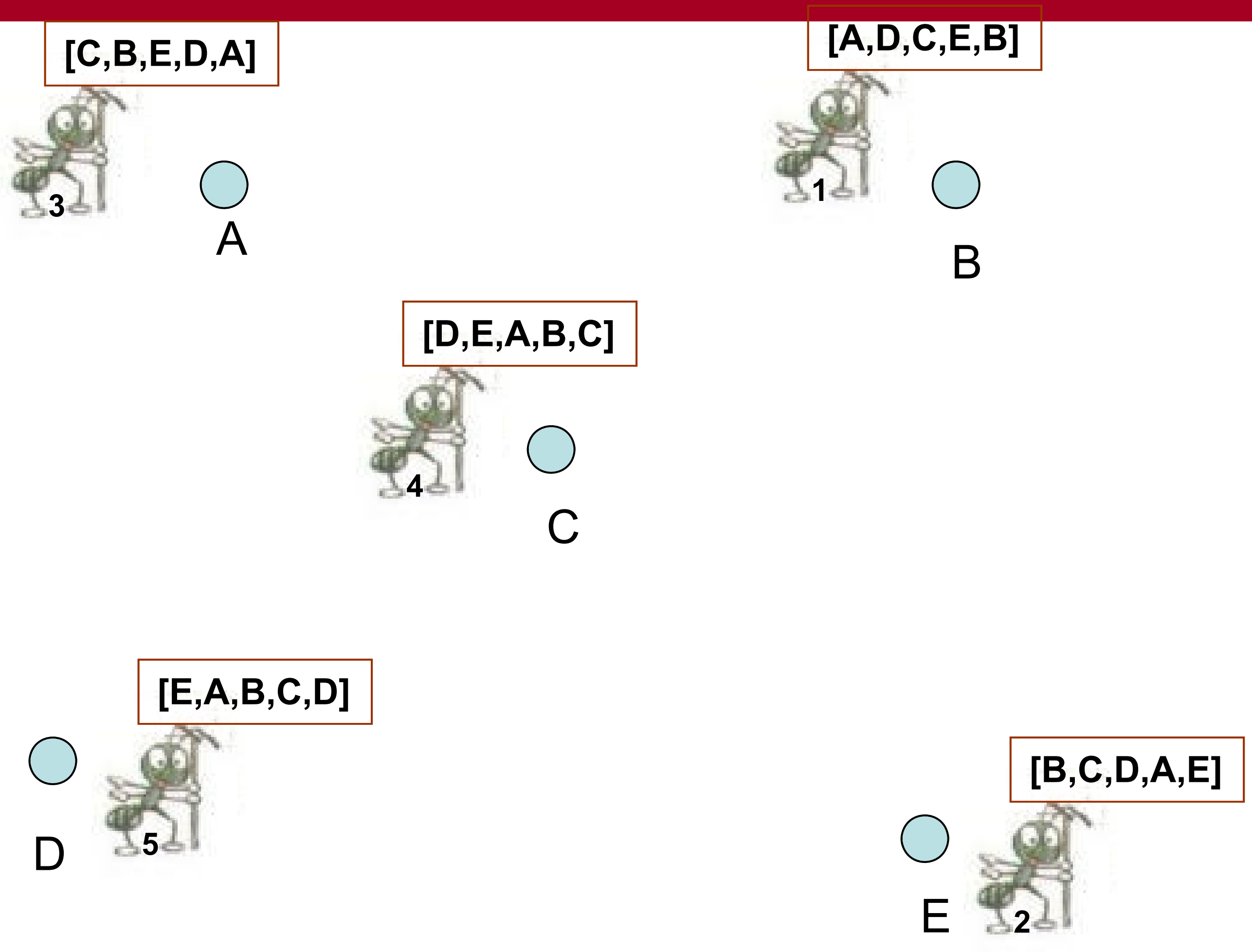
Iteration 3



Iteration 4



Iteration 5

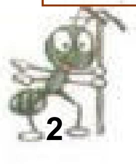


Path and Pheromone Evaluation

[A,D,C,E,B]

$L_1 = 300$

[B,C,D,A,E]



$L_2 = 450$

$$\Delta \tau_{i,j}^{k} = \begin{cases} \frac{Q}{L_k} & if\ (i,j) \in \text{tour} \\ 0 & \text{otherwise} \end{cases}$$

[C,B,E,D,A]

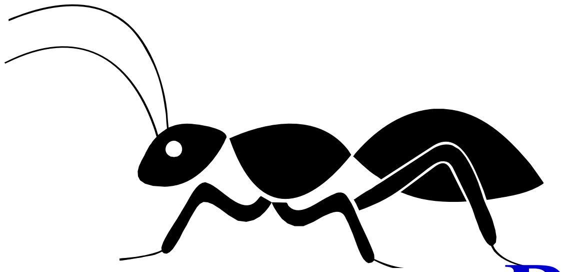
$$\Delta\tau_{A,B}^{total} = \Delta\tau_{A,B}^{1} + \Delta\tau_{A,B}^{2} + \Delta\tau_{A,B}^{3} + \Delta\tau_{A,B}^{4} + \Delta\tau_{A,B}^{5}$$

[D,E,A,B,C]

$L_4 = 280$

[E,A,B,C,D]

$L_5 = 420$



Result

Empirical Results

- Compared Ant Colony Algorithm
  to standard algorithms and meta-heuristic algorithms
  on Oliver 30 – a 30 city TSP
  - Standard: 2-Opt, Lin-Kernighan,
  - Meta-Heuristics: Tabu Search and Simulated Annealing

- Conducted 10 replications of each algorithm
  and provided averaged results

Comparison to Standard Algorithms

- Examined Solution Quality – not speed; in general, standard algorithms were significantly faster.

- Best ACO solution - 420

	2-Opt	L-K
Near Neighbor	437	421
Far Insert	421	420
Near Insert	492	420
Space Fill	431	421
Sweep	426	421
Random	663	421

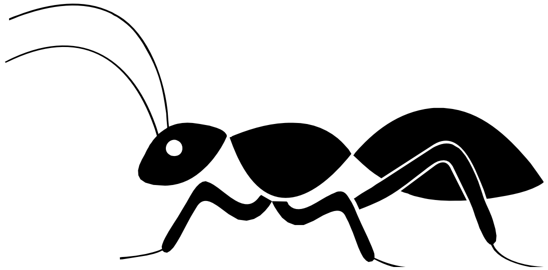
Comparison to Meta-Heuristic Algorithms

- Meta-Heuristics are algorithms that can be applied to a variety of problems with a minimum of customization.
- Comparing ACO to other Meta-heuristics provides a “fair market” comparison (vice TSP specific algorithms).

	Best	Mean	Std Dev
ACO	420	420.4	1.3
Tabu	420	420.6	1.5
SA	422	459.8	25.1

Table 1. Comparison of ACS with other nature-inspired algorithms on random instances of the symmetric TSP. **Comparisons on average tour length** obtained on five 50-city problems.

Problem name	ACS	SA	EN	SOM	FI
City set 1	5.86	5.88	5.98	6.06	6.03
City set 2	6.05	6.01	6.03	6.25	6.28
City set 3	5.57	5.65	5.70	5.83	5.85
City set 4	5.70	5.81	5.86	5.87	5.96
City set 5	6.17	6.33	6.49	6.70	6.71



Advantages and Disadvantages

Advantages and Disadvantages

- For TSPs (Traveling Salesman Problem), relatively efficient
  - for a small number of nodes,
    TSPs can be solved by exhaustive search
  - for a large number of nodes,
    TSPs are very computationally difficult to solve (NP-hard)
    – exponential time to convergence

- Performs better against other global optimization techniques for TSP (neural net, genetic algorithms, simulated annealing)

- Compared to GAs (Genetic Algorithms):
  - retains memory of entire colony
    instead of previous generation only
  - less affected by poor initial solutions
    (due to combination of random path selection and colony memory)

Advantages and Disadvantages, cont.

- Can be used in dynamic applications (adapts to changes such as new distances, etc.)
- Has been applied to a wide variety of applications
- As with GAs, good choice for constrained discrete problems (not a gradient-based algorithm)

Advantages and Disadvantages, cont.

- Theoretical analysis is difficult:
  - Due to sequences of random decisions (not independent)
  - Probability distribution changes by iteration
  - Research is experimental rather than theoretical

- Convergence is guaranteed,
  but time to convergence is uncertain

Advantages and Disadvantages, cont.

- **Tradeoffs in evaluating convergence:**
  - **In NP-hard problems, need high-quality solutions quickly – focus is on quality of solutions**
  - **In dynamic network routing problems, need solutions for changing conditions – focus is on effective evaluation of alternative paths**
- **Coding is somewhat complicated, not straightforward**
  - **Pheromone "trail" additions/deletions, global updates and local updates**
  - **Large number of different ACO algorithms to exploit different problem characteristics**

Problem name	Authors	Algorithm name	Year
Traveling salesman	Dorigo, Maniezzo & Colorni	AS	1991
	Gamberdella & Dorigo	Ant-Q	1995
	Dorigo & Gamberdella	ACS &ACS 3 opt	1996
	Stutzle & Hoos	MMAS	1997
	Bullnheimer, Hartl & Strauss	AS_{rank}	1997
	Cordon, et al.	BWAS	2000
Quadratic assignment	Maniezzo, Colorni & Dorigo	AS-QAP	1994
	Gamberdella, Taillard & Dorigo	HAS-QAP	1997
	Stutzle & Hoos	MMAS-QAP	1998
	Maniezzo	ANTS-QAP	1999
	Maniezzo & Colorni	AS-QAP	1994
Scheduling problems	Colorni, Dorigo & Maniezzo	AS-JSP	1997
	Stutzle	AS-SMTTP	1999
	Barker et al	ACS-SMTTP	1999
	den Besten, Stutzle & Dorigo	ACS-SMTWTP	2000
	Merkle, Middenderf & Schmeck	ACO-RCPS	1997
Vehicle routing	Bullnheimer, Hartl & Strauss	AS-VRP	1999
	Gamberdella, Taillard & Agazzi	HAS-VRP	1999

Problem name	Authors	Algorithm name	Year
Connection-oriented network routing	Schoonderwood et al.	ABC	1996
	White, Pagurek & Oppacher	ASGA	1998
	Di Caro & Dorigo	AntNet-FS	1998
	Bonabeau et al.	ABC-smart ants	1998
Connection-less network routing	Di Caro & Dorigo	AntNet & AntNet-FA	1997
	Subramanian, Druschel & Chen	Regular ants	1997
	Heusse et al.	CAF	1998
	van der Put & Rethkrantz	ABC-backward	1998
Sequential ordering	Gamberdella& Dorigo	HAS-SOP	1997
Graph coloring	Costa & Hertz	ANTCOL	1997
Shortest common supersequence	Michel & Middendorf	AS_SCS	1998
Frequency assignment	Maniezzo & Carbonaro	ANTS-FAP	1998
Generalized assignment	Ramalhinho Lourenco & Serra	MMAS-GAP	1998
Multiple knapsack	Leguizamon & Michalewicz	AS-MKP	1999
Optical networks routing	Navarro Varela & Sinclair	ACO-VWP	1999
Redundancy allocation	Liang & Smith	ACO-RAP	1999