CSE 460
Ant Colony Optimization

In this section we will look at swarm-based optimization

- Emergent Swarm Behaviour
- Ant System Meta-Heuristics
- Ant Colony Meta-Heuristics
- Applications of Ant Colony Optimization

References

- The inventors of Ant-based optimization is Marco Dorigo.
  The ACO home page of his group is at
  http://iridia.ulb.ac.be/~mdorigo/ACO/ACO.html

- Lots of Material at the Santa Fe Institute (www.santafe.edu)

- Book recommendations
  “Swarm Intelligence” by Eric Bonabeau, Marco Dorgio, Guy Theraulaz
  SFI / Oxford University Press, 1999

  “New Ideas in Optimization” (Part One)
  by David Corne, Marco Dorigo and Fred Glover (eds.)
  Mc Graww Hill, 1999

- Some Material is also found in the following conference series:
  International Conference on Evolutionary Computation
  published as Lecture Notes in Computer Science, Springer-Verlag

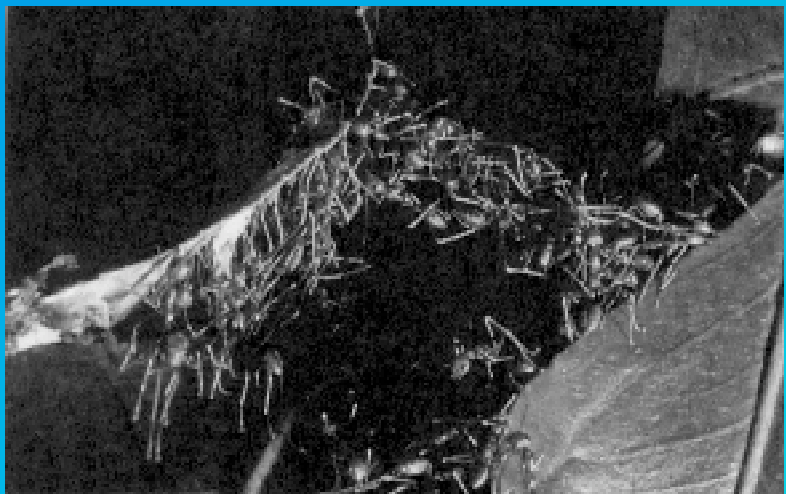
- Some information on collective robotics can be found at
  http://www.cs.ualberta.ca/~kube/crip.cgi

Social Insect Behaviour

- Large assemblies of simple units solve complex tasks
- no Central Control
- self-Organized Behaviour
- emergent Behaviour ("swarm intelligence")

  "Swarm Intelligence" by E. Bonabeau, M. Dorigo and G. Theraulaz
  SFI / Oxford University Press, 1999
- social tasks: foraging, defense & attack, nest building etc.
- Some of these tasks can be viewed are "natural optimization problems"

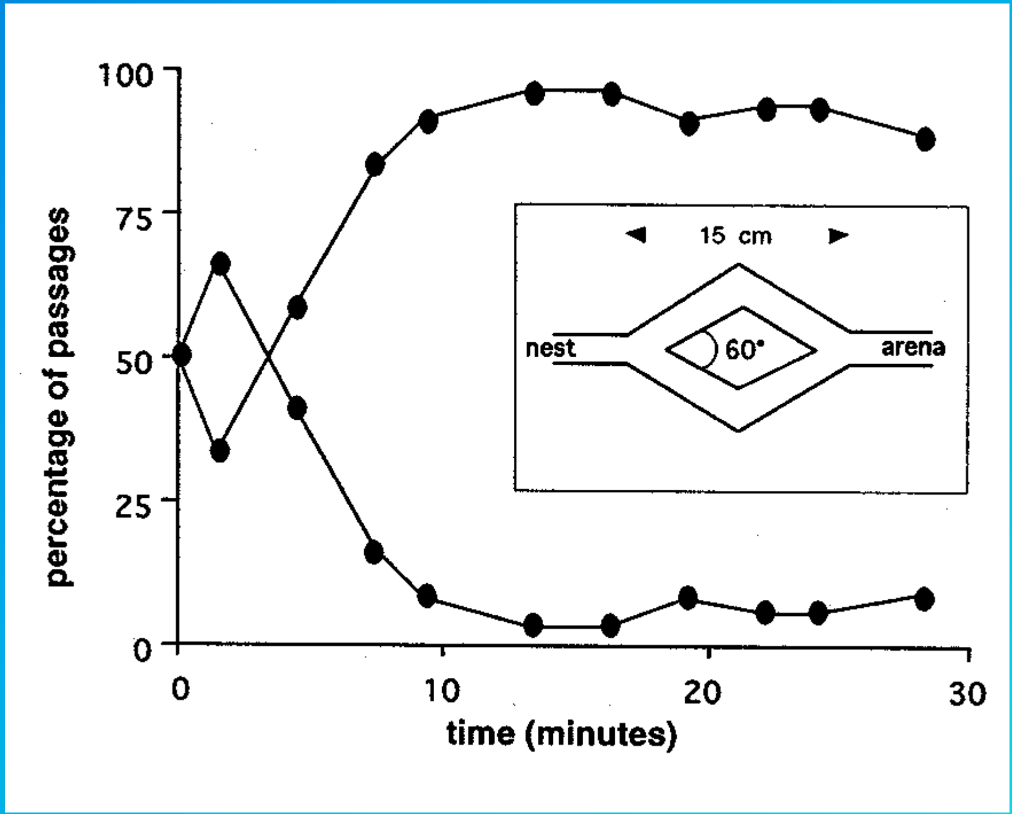
  e.g. foraging as optimal resource usage (food vs. energy waste)



Weaver Ants (Oecophylla)
bridging leaves to pull them together

Collaborative Robotics

Ant Paths: Binary Bridge Experiment



both paths have equal length

- Chemically mediates re-inforcement
- Initial random fluctuations are amplified until a steady state is reached

from J.-L. Deneubourg et al. "The Self-organizing Exploratory Pattern of Argentine Ant" in Journal of Insect Behavior, Vol. 3, 1990, pp. 159-168

Pheromone-based Path Selection

- amount of pheromone encountered influences path selection of ants
- path selection is probabilistic

$$P_A = 1 - P_B = \frac{(k + A_i)^n}{(k + A_i)^n + (k + B_i)^n}$$

A_i / B_i are the numbers of ants that have already used branch A/B after a total of i ants has used A and B.

$P_{A/B}$ probability of $(i+1)$st ant to choose A/B.

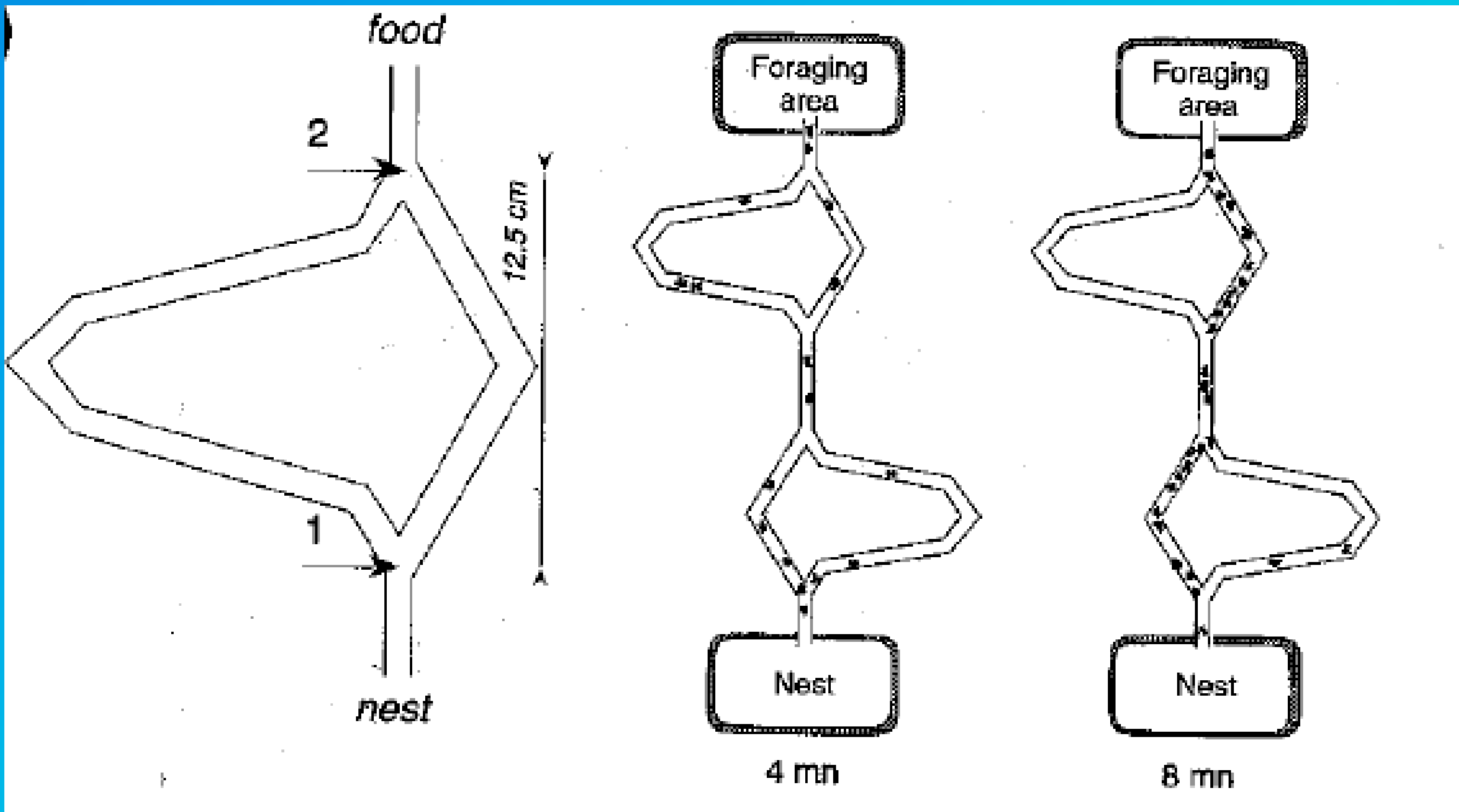
Assumptions

- each ant deposits same amount of pheromone, total amount proportional ant number
- no pheromone evaporation (experiment duration much shorter than evaporation time)
- experimental verification for n=2, k=80

$$A_i >> B_i \wedge A_i >> 20 \quad \Rightarrow \quad P_A \approx 1$$

Foraging

finding a good path to a food source is essentially shortest path selection

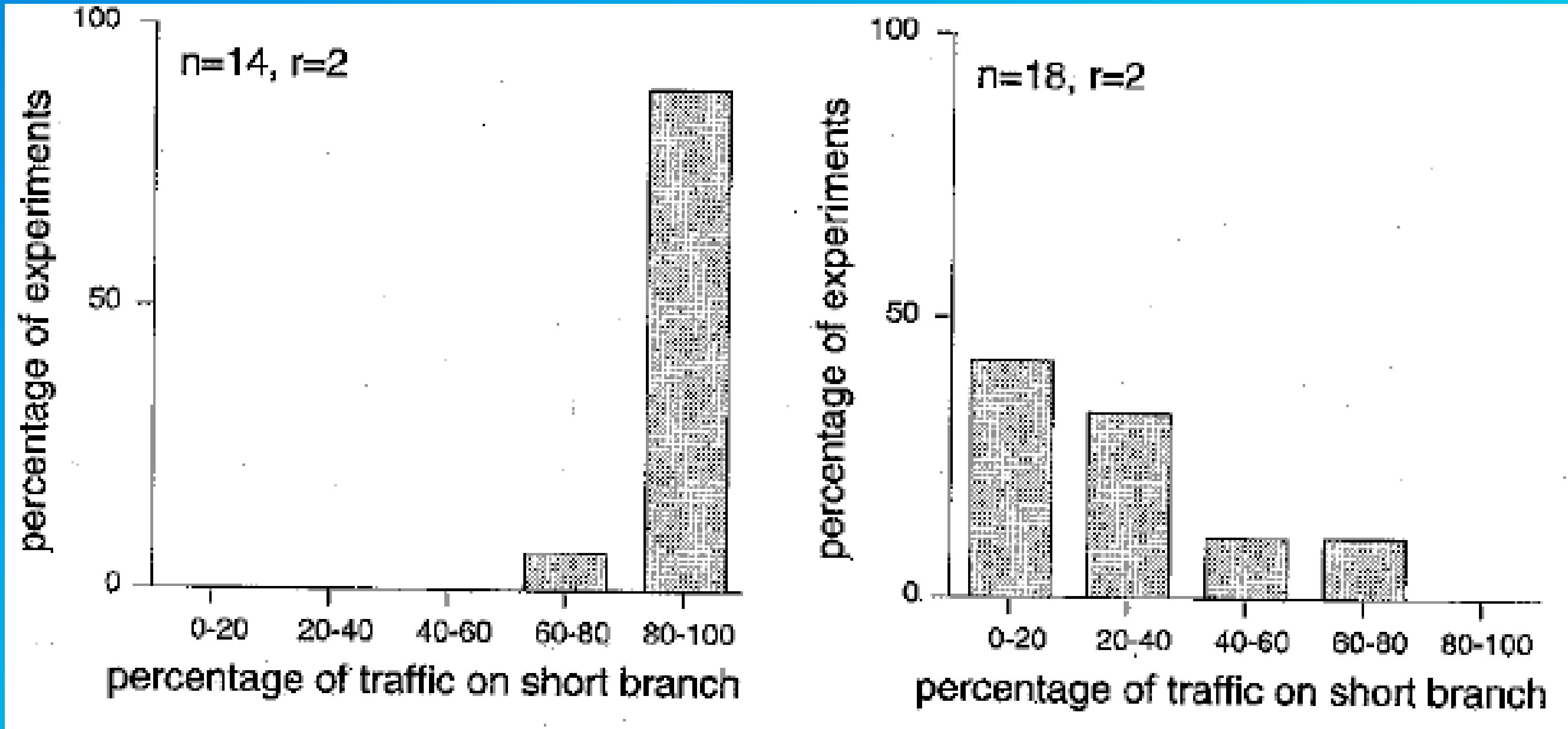


Adaptivity 1

(Linepithema humile)

short branch presented from beginning

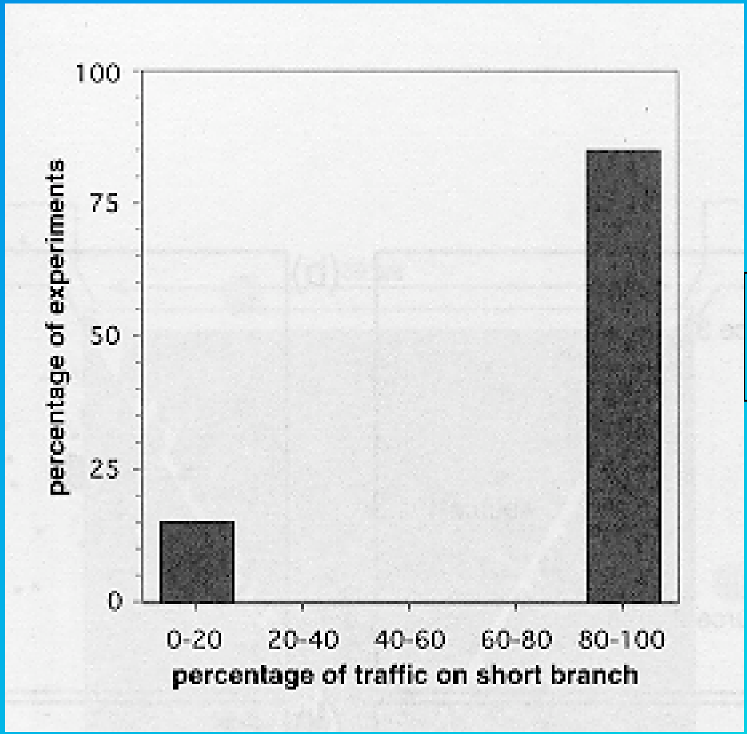
short branch presented after 30 minutes



path selection based on pheromone level only
with long pheromone evaporation time is
not sufficiently adaptive (to changing environments)

Adaptivity 2

(Lasius niger)



short branch
presented after 30 minutes

Lasius niger appears to have some form of direct direction perception.
When it finds itself heading too far out of the intended direction back to the nest it sometimes performs a U-turn.

Fast evaporating pheromone is another mechanism to escape from sub-optimal situations .

- possible in simulations and optimization
- mostly unrealistic in nature

Applications to Optimization

Ant Colony Meta-Heuristics

stochastic optimization
adaptive, dynamic on-line method

Many important optimization problems can be restated as network search or path-finding!

Travelling Salesman
Routing
Job Scheduling
Quadratic Assignment

If the problem is not defined on a graph / network, formalize the solution space as a graph in which the nodes are individual (partial) solutions and the edges represent transitions from a solution to another better or more complete solution.

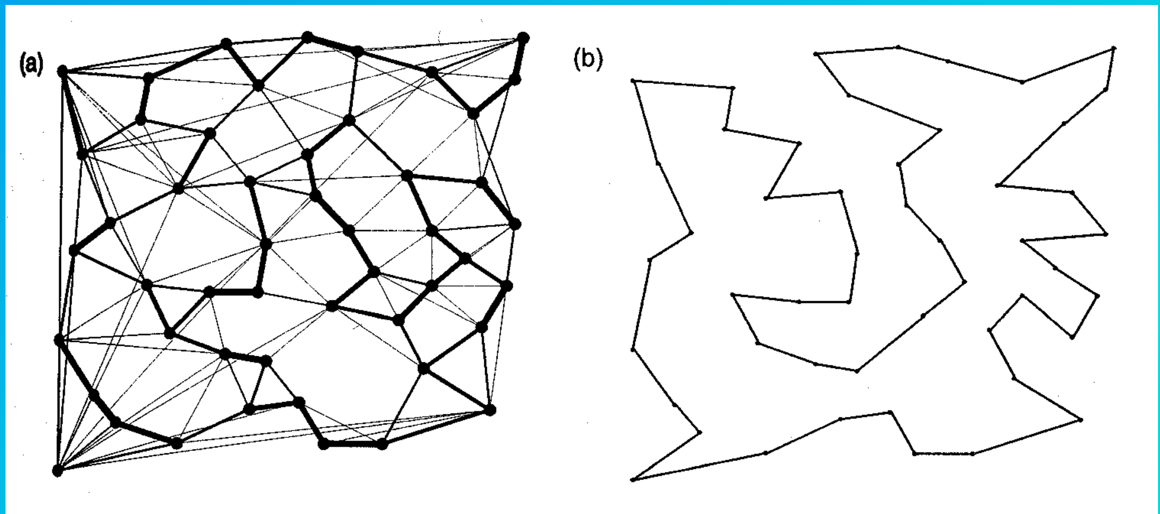
TSP, again

Ant system

- simplest form of ant-based TSP solver
- restricted problem size (<100)
- performance comparable to SA/GA approaches

Ant Colony System

- better performance
- significantly better with local search



Relative pheromone concentration (= line thickness) provides alternative paths as the basis of adaptive optimization to routing problems.

Ant System Transition Rule

Each ant maintains a “tabu list” of cities that it has already visited.
It does not revisit any city.

While an ant builds a tour, it decides on the next city in a probabilistic manner.

The probability to go from city *i* to city *j* is given by:

$$p_{i,j}^{k}(t) = \frac{\left[\tau_{i,j}(t)\right]^{\alpha} \cdot \left[\eta_{i,j}\right]^{\beta}}{\sum_{l \in J_i^k} \left[\tau_{i,l}(t)\right]^{\alpha} \cdot \left[\eta_{i,j}\right]^{\beta}}$$

where

J_i^k is the set of cities that ant k still has to visit when it is on city i

$\eta_{i,j} = 1/d_{i,j}$ is the inverse distance (visibility) between cities i and j

$\tau_{i,j}(t)$ is the amount of peheromone on the link

between cities$i$ and j at time t

Pheromone Deposit & Decay

Only after the completion of a tour each ant deposits pheromone along its path. The deposited amount $\Delta\tau$ is:

$$\Delta\tau_{i,j}^{k}(t) = \begin{cases} Q / L^{k}(t) & \text{for each edge } (i, j) \text{ visited by ant } k \text{ in iteration } t \\ 0 & \text{otherwise} \end{cases}$$

To escape local minima, pheromone evaporation is used.

Evaporation is applied uniformly to all edges with a simple decay coefficient ρ.

The resulting total update pheromone update function therefore is:

$$\tau_{i,j}(t+1) = (1-\rho)\cdot\tau_{i,j}(t) + \sum_{k=1}^{m}\left[\Delta\tau_{i,j}^{k}(t)\right]$$

where m is the number of ants in the system.

AS Algorithm

```
Algorithm AS-TSP

initialize all edges to (small) initial pheromone level τ0;
place each ant on a randomly chosen city;
for t := 1 to t_max do

      for k := 1 to m do
            build a tour T(k,t) for ant k by applying the probabilistic transition rule;
      end;

      if (best T(k,t) better than current solution) update current solution to T(k,t);

      for every edge (i,j) do
            apply pheromone update;
      end;

      // the next block simulates a special "elitist ant" and is optional

      Let b be the ant with the best tour
      for all edges egdes in best tour T(b,t) do
            apply additional pheromone increment with Q/length(T(b,t));
      end

end.
```

Ant Colony Modifications

Improvements over basic Ant System by the following changes

- transition rule
  balance between pheromone and exploration is djustable

- pheromone trail update
  more directed towards re-inforcing the best tour

- local trail updates
  ants “consume” pheromone on the edges they use
  thereby making them less attractive to other ants.
  This envorces diversity of the solutions explored.

- candiate lists (of cities)
  implements heuristics: closer cities are preferred choices.
  Initialized for each city i with the c cities closest to city i

ACS Transition Rule

On each step each ant tosses a coin whether to explore or to follow a trail.
The next city visited is *city*(*j*) with:

$$j = \begin{cases} \arg\max_{u \in J_i^k} \left\{ [\tau_{i,u}(t)]^{\alpha} \cdot [\eta_{i,J}]^{\beta} \right\} & \text{if } random[0,1] < q_0 \\ J & \text{if } random[0,1] \geq q_0 \end{cases}$$

where

J_i^k is the set of cities that ant k still has to visit when it is on city i

$\eta_{i,j} = 1/d_{i,j}$ is the inverse distance (visibility) between cities i and j

$\tau_{i,j}(t)$ is the amount of pheromone on the link between cities i and j at time t

J is chosen according to the same probabilistic transition rule used by the basic AS-TSP

ACS Peromone Update & Decay

- Global Trail Update (intensifies search in the neighborhood of good solutions)

  Only the best ant updates the edges belonging to the best tour in this trial.

$$\Delta\tau_{i,j}(t) = \begin{cases} 1/L^{+} & \text{for all edges visited by the best ant in iteration } t \\ 0 & \text{otherwise} \end{cases}$$

$$\tau_{i,j}(t+1) = (1-\rho)\cdot\tau_{i,j}(t) + \rho\cdot\Delta\tau_{i,j}(t)$$

- Local trail update (diversifies search)

  When an ant passes an edge it consumes pheromone according to

$$\tau_{i,j}(t+1) = (1-\rho)\cdot\tau_{i,j}(t) + \rho\cdot\tau_{0}$$

where τ_0 is the pheromone level for trail initialization

ACS Algorithm

```
Algorithm AS-TSP

initialize all edges to (small) initial pheromone level τ0;
place each ant on a randomly chosen city;
for k := 1 to m do initialize candidate list k to the c closest cities of k end;

for t := 1 to t_max do
    for k := 1 to number of ants do
        until (tour T(k,t) for ant k is complete) do
            if there is at least one unvisited city in candidate list k
                then choose the next city among the candidates
                        by applying the probabilistic transition rule;
                else choose the next city as the next city still to be visited;
            perform local trail update;
        end;
        optional: locally improve tour T(k,t);
    end;

    if (best T(k,t) better than current solution) update current solution to T(k,t);

    for every edge on the current solution do
        apply global trail update;
    end;
end.
```

ACS Evaluation

ACS compared with basic AS on standard benchmark TSPs

- better quality results
- slightly faster convergence

	ACS-TSP best	ACS-TSP # iter.	GA best	GA # iter.	EP best	EP # iter.	SA best	SA # iter.
Eil50 (50-city problem)	**425** (427.96)	**1830**	428 (N/A)	25000	426 **(427.86)**	100000	443 (N/A)	68512
Eil75 (75-city problem)	**535** **(542.37)**	**3480**	545 (N/A)	80000	542 (549.18)	325000	580 (N/A)	173250
KroA100 (100-city problem)	**21282** **(21285.44)**	**4820**	21761 (N/A)	103000	N/A (N/A)	N/A	N/A (N/A)	N/A

Comparsion of ACS with other general purpose meta-heuristics

(*numbers in brackets are for real-valued problem instances*)

ACS + Local Optimization

- ACS still not competitive for larger problem
- hybridization with local search (3-opt) produces competitive results

	ACS-3-opt best	ACS-3-opt average	STSP best	STSP average
d198 (198-city problem)	**15780**	15781.7	**15780**	**15780**
lin318 (318-city problem)	**42029**	**42029**	**42029**	**42029**
att532 (532-city problem)	27693	27718.2	**27686**	**27693.7**
rat783 (783-city problem)	8818	8837.9	**8806**	**8807.3**

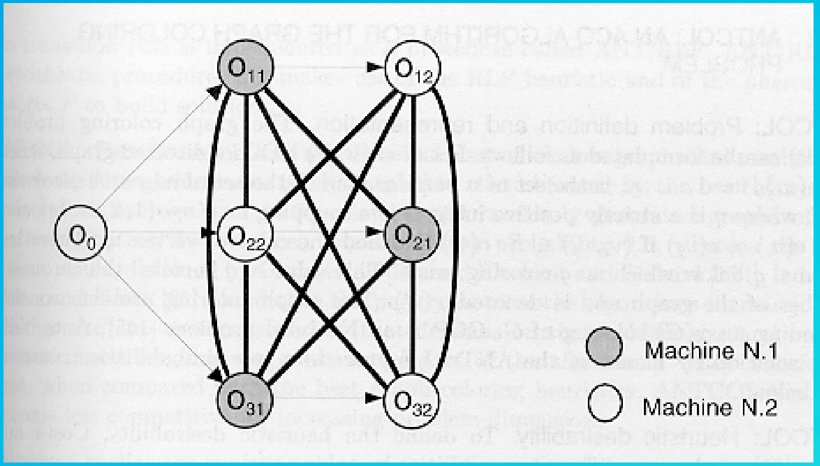
Comparsion of ACS-3opt with best known genetic method

Generalization of AS

Ant meta-heuristics can be generalized to non graph-like problems.

- Job Scheduling
- Graph Coloring
- Shortest Supersequence
- Vehicle Routing
- Call Routing

...



we need to find a problem representation that allows to view

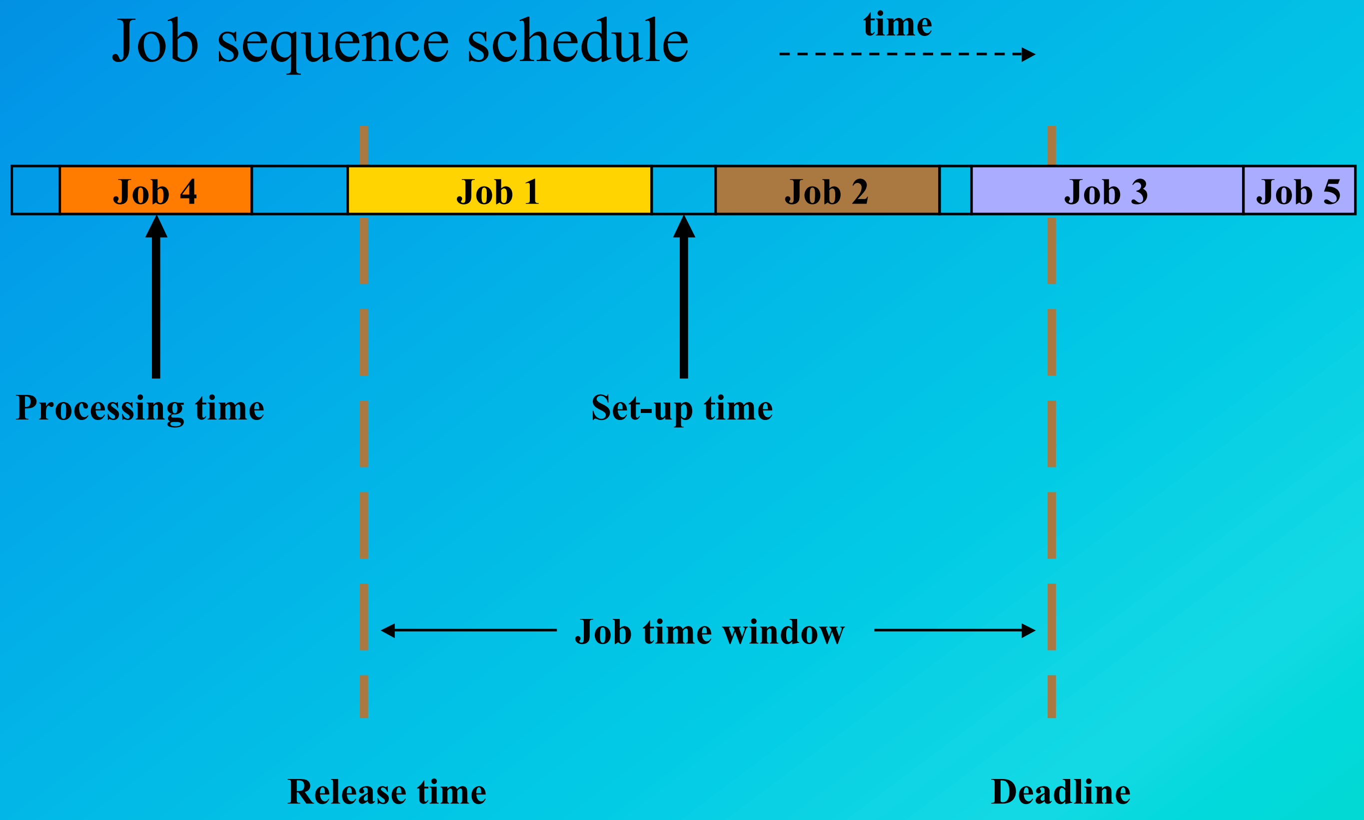
- (partial) solutions as nodes/paths in a network
- extension or modification of solutions as transitions between nodes

and to define a

- heuristic “desirability” for these transitions

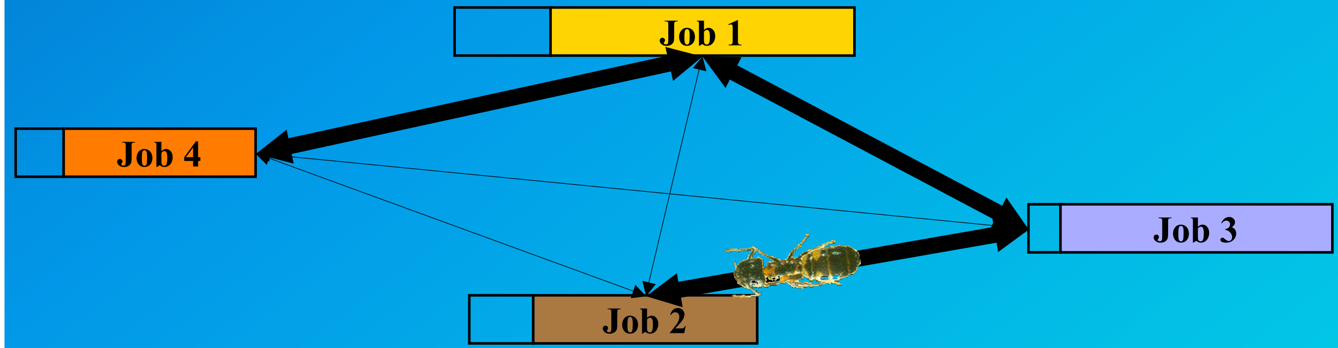
some mechanism must ensure that the constructed solutions are feasible,
i.e. that they satisfy the problem constraints

Single Machine Scheduling



ACO in Machine Scheduling

(Project with CSIRO)



Re-scheduling

Machine breakdown

Machine maintenance

Job cancellation

Job entrance

Etc….

Edge Length:	tardiness / set-up time / ...
Visibility:	earliest due date

Quadratic Assignment

another NP-hard problem...

<u>QAP Problem Statement</u>

<u>given</u>

- *n* activities
- *n* locations
- A matrix *D*=[*dij*] of dimensions *n* x *n* with distances between location *i* and location *j*
- A matrix *F*=[*fij*] of dimensions *n* x *n* with required flow between activity *i* and activity *j*

<u>find</u>

A permutation π of {1, ..., *n*} where π(*i*) is the activity assigned to location *i*
such that the transportation costs are minimized

The total transportation cost is the sum of the products of flow and distance.

$$Opt = \arg\min_{\pi} \left\{ C(\pi) = \sum_{i,j=1}^{n} \left[d_{i,j} \cdot f_{\pi(i),\pi(j)} \right] \right\}$$

AS-QAP

Min-Max Coupling Rule

try to assign activities with high flow potential to locations with low distance potential

The heuristics uses two kind of estimates:

distance potential $$d_i = \sum_{j=1}^{n} d_{i,j}$$

flow potential $$f_i = \sum_{j=1}^{n} f_{i,j}$$

distance potential (sum of distances to all other nodes)
the lower the distance potential for i the more barycentric is this node.
flow potential (sum of flow to all other activities)
the higher the flow potential, the more important it is to optimize this node.

Min-Max Rule Example

$$D = \begin{bmatrix} 0 & 10 & 4 & 2 \\ 10 & 0 & 6 & 4 \\ 4 & 6 & 0 & 1 \\ 2 & 4 & 1 & 0 \end{bmatrix} \qquad F = \begin{bmatrix} 0 & 3 & 8 & 3 \\ 3 & 0 & 2 & 4 \\ 8 & 2 & 0 & 5 \\ 3 & 4 & 5 & 0 \end{bmatrix}$$

$$d = \left[d_{i,j} \right] = \begin{bmatrix} 16 & 20 & 11 & 7 \end{bmatrix} \qquad f = \left[f_{i,j} \right] = \begin{bmatrix} 14 & 9 & 15 & 12 \end{bmatrix}$$

$$\text{Let } \; B = d^T \cdot f = \begin{bmatrix} 224 & 144 & 240 & 192 \\ 280 & 180 & 300 & 240 \\ 154 & 99 & 165 & 132 \\ 98 & 63 & 105 & 84 \end{bmatrix}$$

In each step do

select among the free locations the one with lowest distance potential

assign to it from the remaining activities the one with highest *Bij*

Min-Max Rule Example

$$D = \begin{bmatrix} 0 & 10 & 4 & 2 \\ 10 & 0 & 6 & 4 \\ 4 & 6 & 0 & 1 \\ 2 & 4 & 1 & 0 \end{bmatrix} \qquad F = \begin{bmatrix} 0 & 3 & 8 & 3 \\ 3 & 0 & 2 & 4 \\ 8 & 2 & 0 & 5 \\ 3 & 4 & 5 & 0 \end{bmatrix}$$

$$d = \left[d_{i,j} \right] = \begin{bmatrix} 16 & 20 & 11 & 7 \end{bmatrix} \qquad f = \left[f_{i,j} \right] = \begin{bmatrix} 14 & 9 & 15 & 12 \end{bmatrix}$$

$$\text{Let } B = d^T \cdot f = \begin{bmatrix} 224 & 144 & 240 & 192 \\ 280 & 180 & 300 & 240 \\ 154 & 99 & 165 & 132 \\ 98 & 63 & 105 & 84 \end{bmatrix}$$

In each step do

- select among the free locations the one with lowest distance potential
- assign to it from the remaining activities the one with highest *Bij*

Min-Max Rule Example

$$D = \begin{bmatrix} 0 & 10 & 4 & 2 \\ 10 & 0 & 6 & 4 \\ 4 & 6 & 0 & 1 \\ 2 & 4 & 1 & 0 \end{bmatrix} \qquad F = \begin{bmatrix} 0 & 3 & 8 & 3 \\ 3 & 0 & 2 & 4 \\ 8 & 2 & 0 & 5 \\ 3 & 4 & 5 & 0 \end{bmatrix}$$

$$d = \left[d_{i,j} \right] = \begin{bmatrix} 16 & 20 & 11 & 7 \end{bmatrix} \qquad f = \left[f_{i,j} \right] = \begin{bmatrix} 14 & 9 & 15 & 12 \end{bmatrix}$$

$$\text{Let } B = d^T \cdot f = \begin{bmatrix} 224 & 144 & 240 & 192 \\ 280 & 180 & 300 & 240 \\ 154 & 99 & 165 & 132 \\ 98 & 63 & 105 & 84 \end{bmatrix}$$

In each step do

- select among the free locations the one with lowest distance potential
- assign to it from the remaining activities the one with highest *Bij*

Min-Max Rule Example

$$D = \begin{bmatrix} 0 & 10 & 4 & 2 \\ 10 & 0 & 6 & 4 \\ 4 & 6 & 0 & 1 \\ 2 & 4 & 1 & 0 \end{bmatrix} \qquad F = \begin{bmatrix} 0 & 3 & 8 & 3 \\ 3 & 0 & 2 & 4 \\ 8 & 2 & 0 & 5 \\ 3 & 4 & 5 & 0 \end{bmatrix}$$

$$d = \left[d_{i,j}\right] = [16 \quad 20 \quad 11 \quad 7] \qquad f = \left[f_{i,j}\right] = [14 \quad 9 \quad 15 \quad 12]$$

$$\text{Let } B = d^T \cdot f = \begin{bmatrix} 224 & 144 & \text{[redacted]} & 192 \\ 280 & 180 & \text{[redacted]} & 240 \\ 154 & 99 & \text{[redacted]} & 132 \\ \text{[redacted]} & & & \end{bmatrix}$$

In each step do

select among the free locations the one with lowest distance potential

assign to it from the remaining activities the one with highest *Bij*

Probabilistic Choice

The probabilistic rule for letting the *k*-th ant couple activity *j* to location *i* is

$$p_{i,j}^{k}(t) = \frac{\left[\tau_{i,j}(t)\right]^{\alpha} \cdot \left[\eta_{i,j}\right]^{\beta}}{\sum_{l \in J_i^k} \left[\tau_{i,l}(t)\right]^{\alpha} \cdot \left[\eta_{i,j}\right]^{\beta}}$$

where

J_i^k is the set of actvities that ant k still has to assign in step i

$\eta_{i,j} = b_{i,j}$

$\tau_{i,j}(t)$ is the amount of pheromone on the coupling of acitivity i to location j

Note: This is identical to the AS-TSP transition rule with ηij=bij

Trail Update

Trail update is similar to AS-TSP with

$$\tau_{i,j}(t+1) = (1-\rho)\cdot\tau_{i,j}(t) + \sum_{k=1}^{m}\left[\Delta\tau_{i,j}^{k}(t)\right]$$

and

$$\Delta\tau_{i,j}^{k}(t) = \begin{cases} Q / C^{k}(t) & \text{for each edge } (i, j) \text{ visited by ant } k \text{ in iteration } t \\ 0 & \text{otherwise} \end{cases}$$

where $C^k(t)$ is the value of the objective function for ant k at iteration t.

Evaluation of AS-QAP

	Nugent (7)	Nugent (12)	Nugent (15)	Nugent (20)	Nugent (30)	Elshafei (19)	Krarup (30)
SA	**148**	**578**	**1150**	**2570**	6128	17937024	89800
TS	**148**	**578**	**1150**	**2570**	**6124**	**17212548**	90090
GA	**148**	588	1160	2688	6784	17640584	108830
ES	**148**	598	1168	2654	6308	19600212	97880
SC	**148**	**578**	**1150**	**2570**	6154	**17212548**	**88900**
AS-QAP	**148**	**578**	**1150**	2598	6232	18122850	92490
AS-LS	**148**	**578**	**1150**	**2570**	6146	**17212548**	89300
AS-SA	**148**	**578**	**1150**	**2570**	6128	**17212548**	**88900**

Comparison of AS applied to QAP with other meta-heuristics

AS-LS and AS-SA (improved with local search)

no reasonable indication of computation time in the source

Telecom Routing

Call routing

- is a dynamic on-line problem (call loads change permanently)

    needs an adaptive solver

        quasi static: re-routing only after exceptions (e.g. link failure)

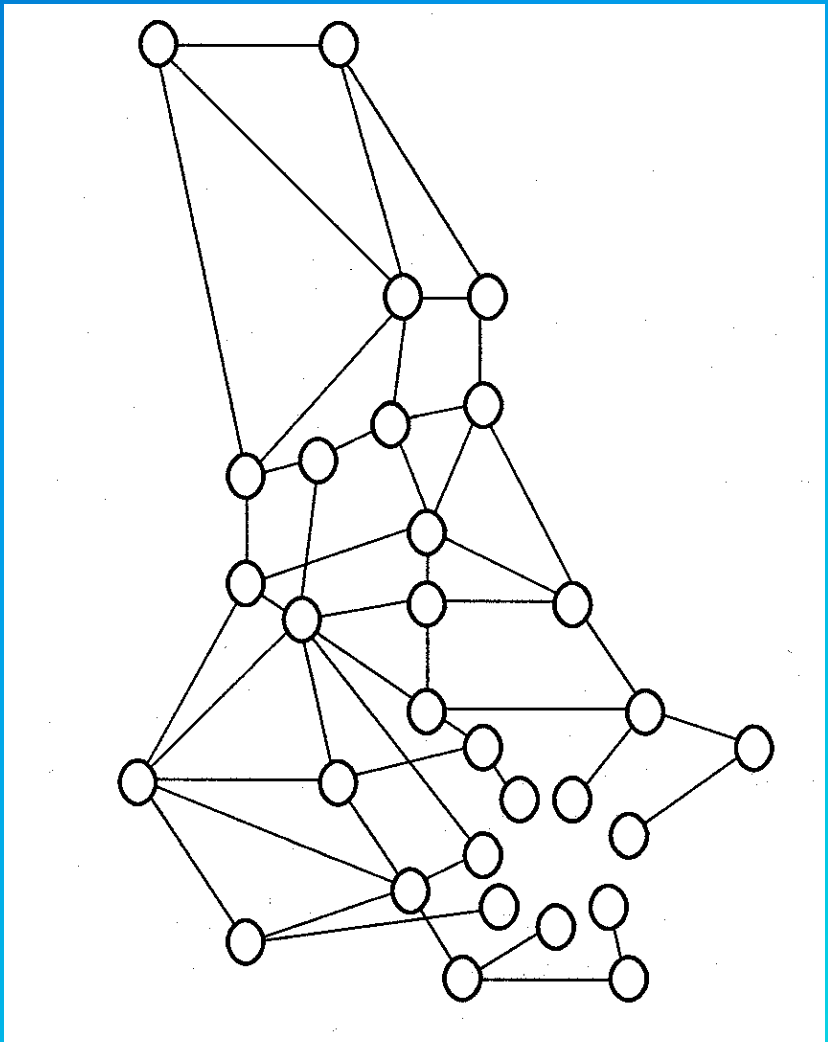
        fully dynamic: autonomous re-routing according to cost function

- must satisfy several objectives simultaneously:

    Maximum performance + minimum cost

(in the following it is assumed that link quality is independent of link load)

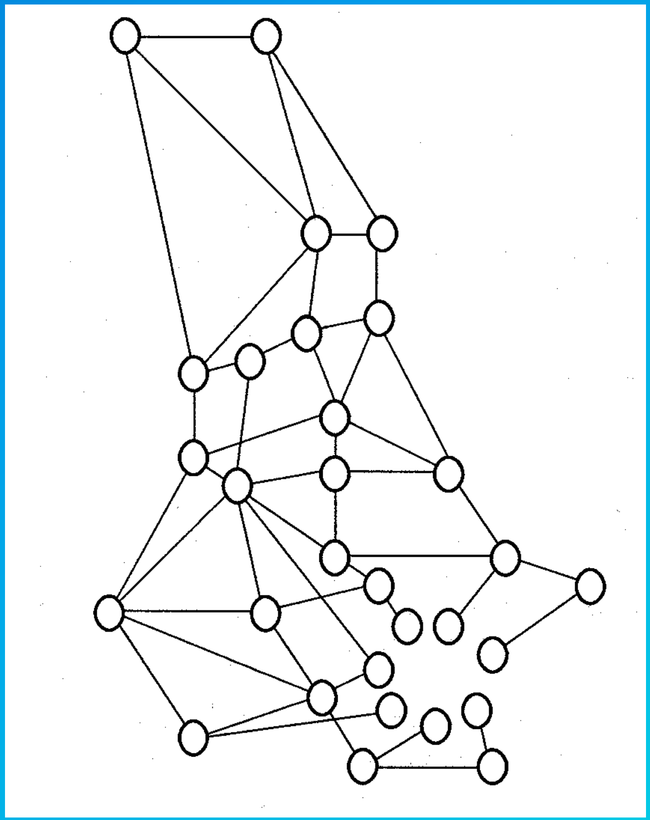
BT Backbone Network



Each node *nj* in the network is characterized by

- total capacity *Cj*
- spare capacity *Sj*
- routing table *Rj*

Telecom Routing (BT Backbone)



Idea:

packages are routed leaving pheromone traces
Where the amount of pheromone deposited
Is moderated by the congestion encountered

Note that this is a dynamic routing which adapts
When demands change.

Nature Inspired Meta-Heuristics

	Simulated Annealing	*Tabu Search*	*Neural Networks*	*Evolutionary Computation*	*Ant Colonies*
Practical Results	✓	➶	➶	➶	❍
Theoretical Framework	✓	➶	➶	➶	❍
Commerical Products	➶	➶	➶	❍1)	-2)
Complexitiy Studies	➶	➶	❍	❍	-

1) should now be updated to ➶
2) should now be updated to ❍

from:
A. Colorni et al. “Heuristics from Nature for Hard Combinatorial Optimization Problems”. in Int. Transactions in Operational Research, Vol. 3, No. 1, pp 1-21.

Summary

In this section we we have looked at swarm-based optimization

- Emergent Swarm Behaviour
- Ant System Meta-Heuristics
- Ant Colony Meta-Heuristics
- Applications of Ant Colony Optimization