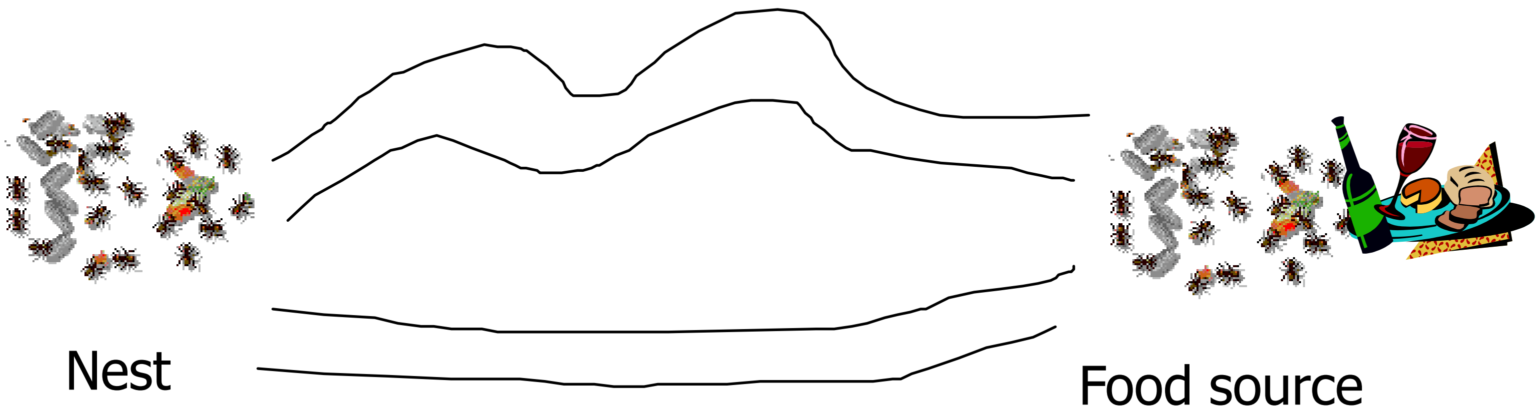
Pengenalan Distributed Artificial Intelligence

Swarm Intelligence

Swarm Intelligence

- Swarm Intelligence is a new computational and behavioral metaphor for solving distributed problems;
- It is based on the principles underlying the behavior of natural systems consisting of many agents, such as ant colonies and bird flocks .
- The approach emphasizes distributedness, direct or indirect interactions among relatively simple agents, flexibility, and robustness

Ant Foraging



DAI Applications

- Swarm Intelligence
  - Multi (collaborative) robots
  - Routing problems
  - Communication networks
  - Distributed control
  - Various forms of optimizations
  - Etc.
- Software Agents
  - Distributed data base
  - Computer networks
  - Security
  - Finance, Banks
  - E-commerce
  - Etc.

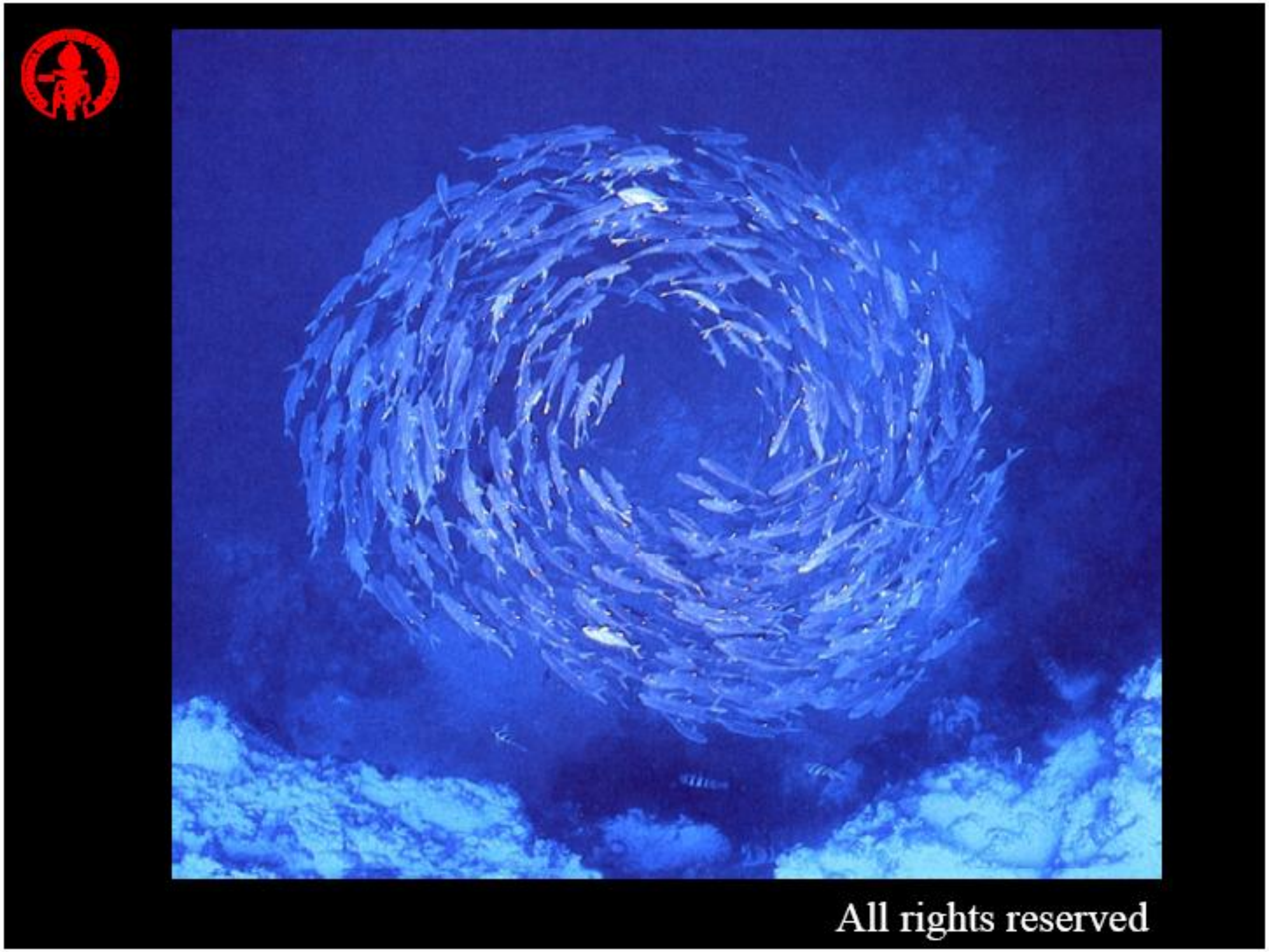
Text Books

- Swarm Intelligence : From Natural to Artificial Systems, Eric Bonabeau et al., Oxford Univ. Press, 1999
- Ant Colony Optimization, M.Dorigo, MIT Press, 2004
- Intelligent Software Agents, Richard Murch & Tony Johnson, Prentice-Hall, 1999
- Artificial Intelligence : A Modern Approach, Russel & Norvig, …., 2003

Biological Motivation

- Biological Inspiration from: social insects (ants, bees, termites) flocks of birds, herds of mammals, schools of fish, packs of wolves, pedestrians, traffic.
- Colonies of social insects can achieve flexible, reliable, intelligent, complex system level performance from insect elements which are stereotyped, unreliable, unintelligent, and simple.
- Insects follow simple rules, use simple local communication (scent trails, sound, touch) with low computational demands.
- Global structure (e.g. nest) reliably *emerges* from the unreliable actions of many.

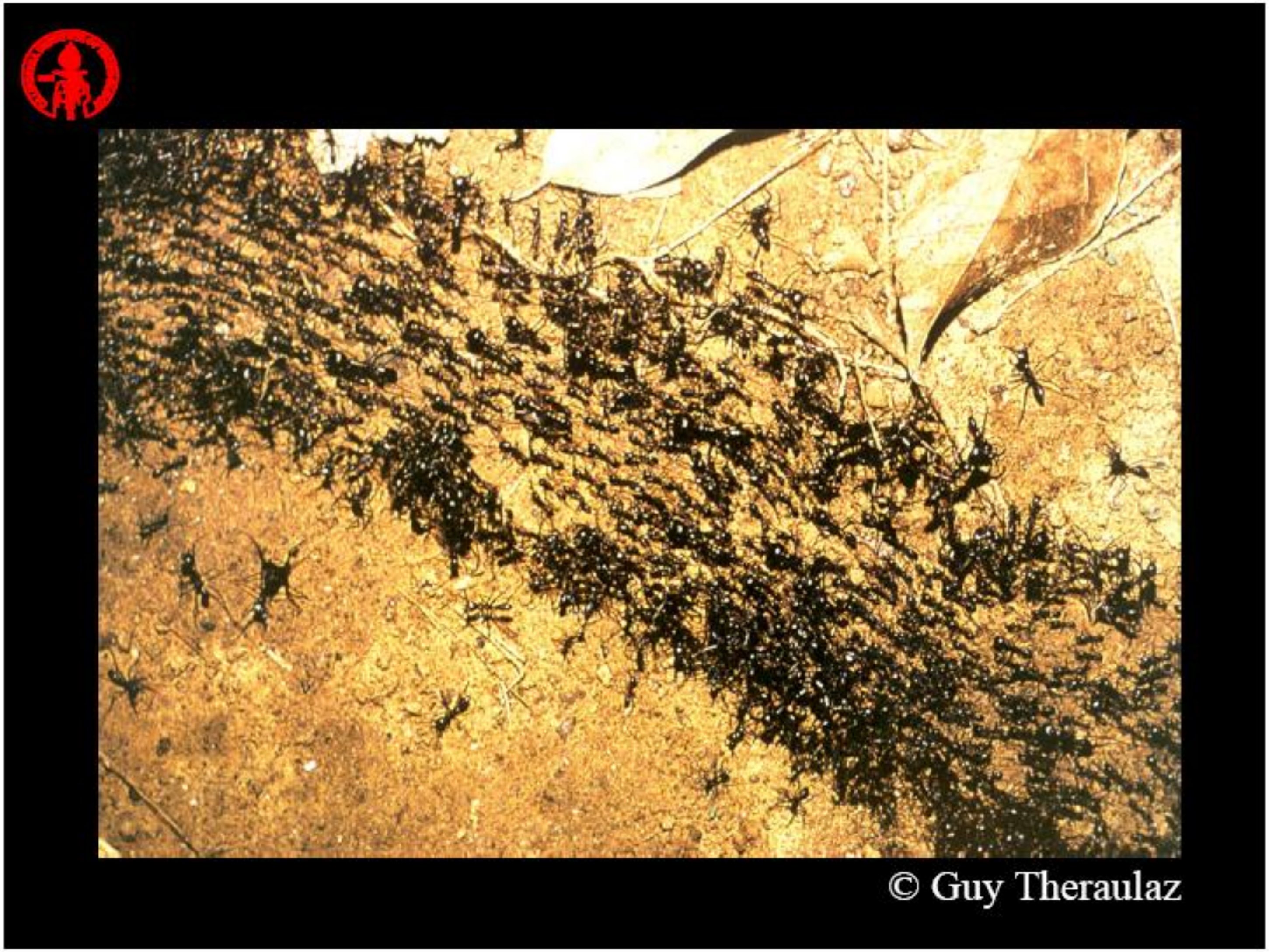

All rights reserved


All rights reserved


© Guy Theraulaz


© Pascal Goetgheluck

Human Swarms


© Guy Theraulaz


www.cloudforestalive.org


andy makely
amakely@ilsnet.com


http://vesmir.kav.cz
snimek © Pavel Hošek, Exp


© Pascal Goetgheluck


© Pascal Goetgheluck

Insect Societies

A natural model of distributed problem solving

- **Collective systems capable of accomplishing difficult tasks, in dynamic and varied environments, without any external guidance or control and with no central coordination**
- **Achieving a collective performance which could not normally be achieved by any individual acting alone**
- **Constituting a natural model particularly suited to distributed problem solving**
- **Many studies have taken inspiration from the mode of operation of social insects to solve various problems in the artificial domain**

Insect Societies

Individual simplicity and collective complexity

- The behavioural repertoire of the insects is limited
- their cognitive systems are not sufficiently powerful to allow a single individual with access to all the necessary information about the state of the colony to guarantee the appropriate division of labour and the advantageous progress of the colony
- the colony as a whole is the seat of a stable and self-regulated organisation of individual behaviour which adapts itself very easily to the unpredictable characteristics of the environment within which it evolved

Self-organisation

Systems of collective decision-making

- Insect societies have developed systems of collective decision-making operating without symbolic representations, exploiting the physical constraints of the environment in which they evolved, and using communications between individuals, either directly when in contact, or indirectly (stigmergy) using the environment as a channel of communication

- Through these direct and indirect interactions, the society self-organises and, faced with a problem finds a solution with a complexity far greater than that of the insects of which it is composed

Collective or Swarm Intelligence

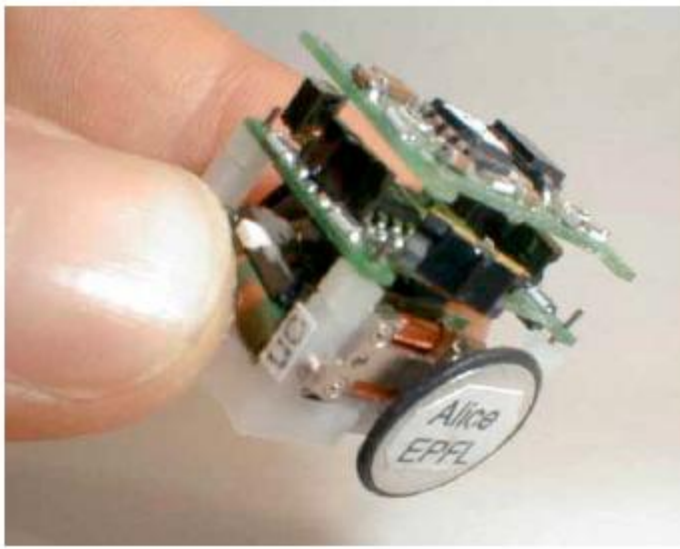
Some questions ...

- **How do insect societies manage to perform difficult tasks, in dynamic and varied environments, without any external guidance or control, and with no central coordination?**
- **How can a large number of entities with only partial information about their environment solve problems?**
- **How can collective cognitive capacities emerge from individuals with limited cognitive capacities?**

A Few Concepts

- Locality and Globality
  - based on physical proximity.
  - E.g. local and global communication.
  - E.g. local and global information.
- Flexibility is the capacity of a society to change its collective behavior [Gordon92].
- Plasticity is defined as the capability of individuals to adapt their control parameters.
- Adaptation implies not only the capacity for change, but the additional requirement that this change represents an improvement in 'fit'[Belew96].
- Centralized and Decentralized Team Control
  - Centralized: central unit coordinates the group decisional processes.
  - Decentralized: no central coordination.
- Hierarchical and Distributed Team Control
  - Hierarchical control: locally centralized.
  - Distributed control: each teammate has full decisional power.
- An intelligent individual is able to ....
  - act in its environment so that a viability condition is always satisfied.
  - maintain its identity (in a broad sense).
- A team is provided with collective intelligence if the viability of the team is required in order to achieve the viability of the individual.

Alice Micro-Robot



•Developed by G. Caprari, Autonomous System Lab, EPFL, Switzerland

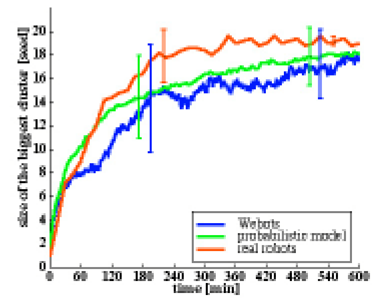
•We are working with the developers to put nose chips on the robot

•Main Features:

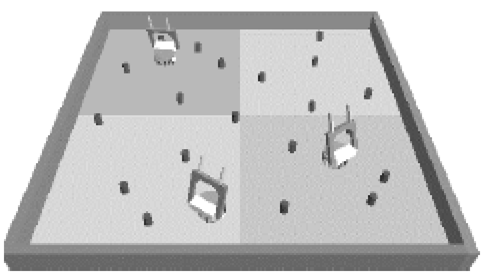
- •Modular
- •Dimensions: 22mm x 20mm x 19mm
- •Max Speed: 35 mm/s
- •Power <10mW
- •Power autonomy up to 10 hours.
- •4 proximity sensors
- •Local IR robot-robot communications
- •Low power radio comms robot-robot and robot-host PC. Range 10m.
- •PIC 16F84 with 1Kb Flash memory

Probabilistic Modelling and Collective System Optimization

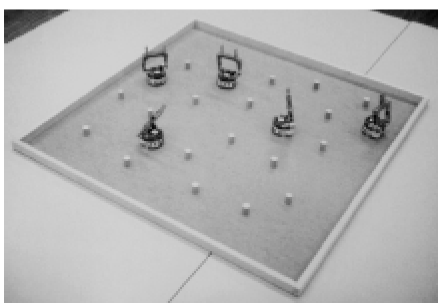
- Understanding and prediction of collective team performance based on single-robot features.
- Prediction of large swarm of robots .
- System optimization (number of robots, control parameters, body and actuator morphology, sensor and communication range and type, …).



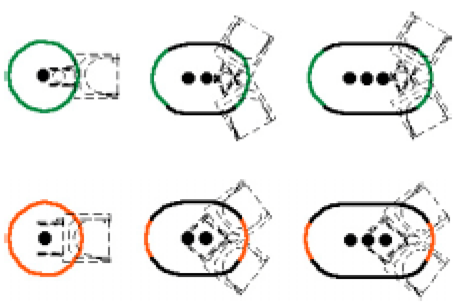
Embodied simulator

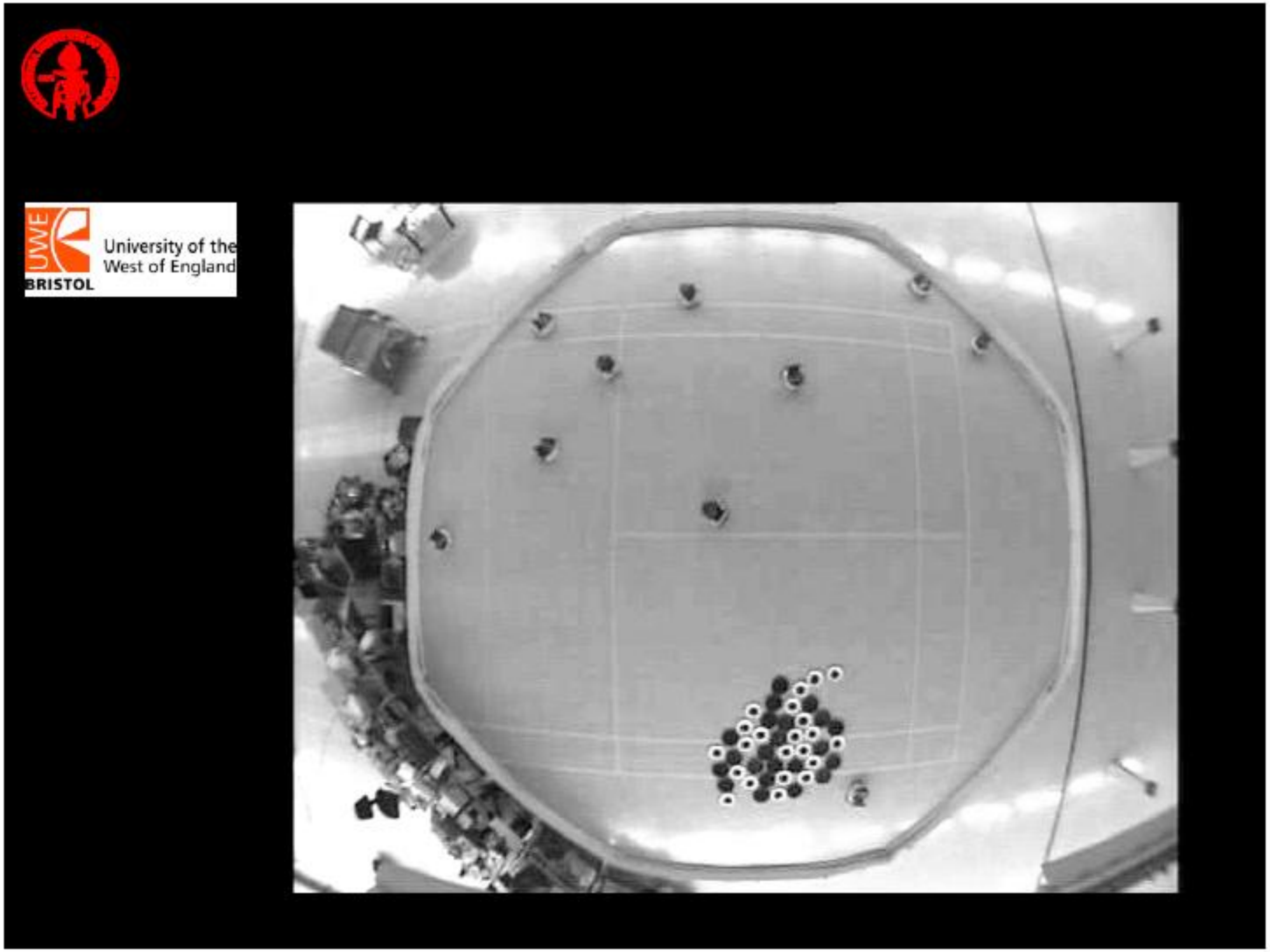


Real robots



Probabilistic model



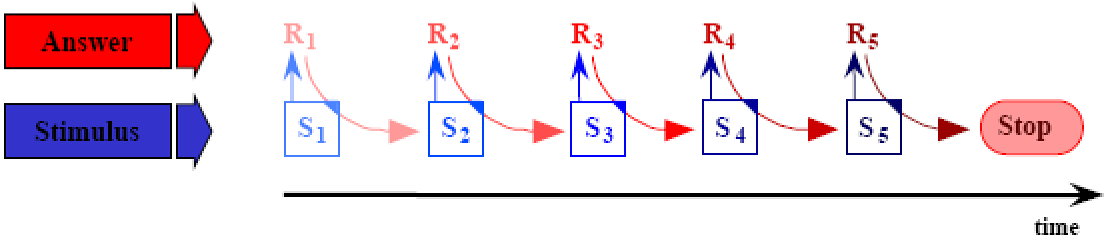

UWE
University of the
West of England
BRISTOL

Ant System/Ant Colony Optimization

Stigmergy

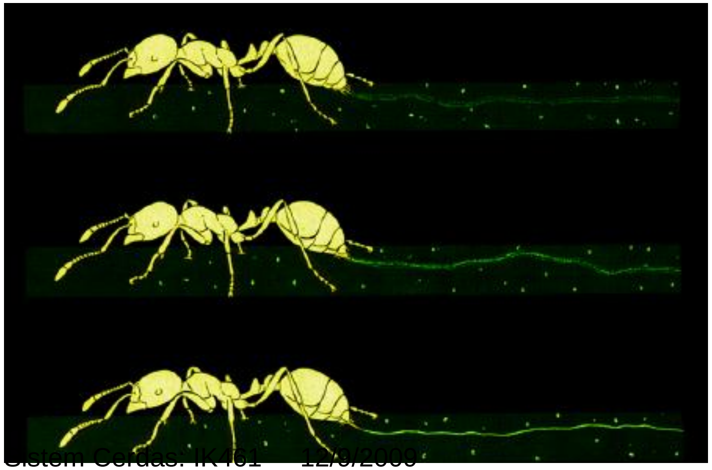
Definition

It defines a class of mechanisms exploited by social insects to coordinate and control their activity via indirect interactions.



Stigmergic mechanisms can be classified in two different categories:

- quantitative (or continuous) stigmergy
- qualitative (or discrete) stigmergy

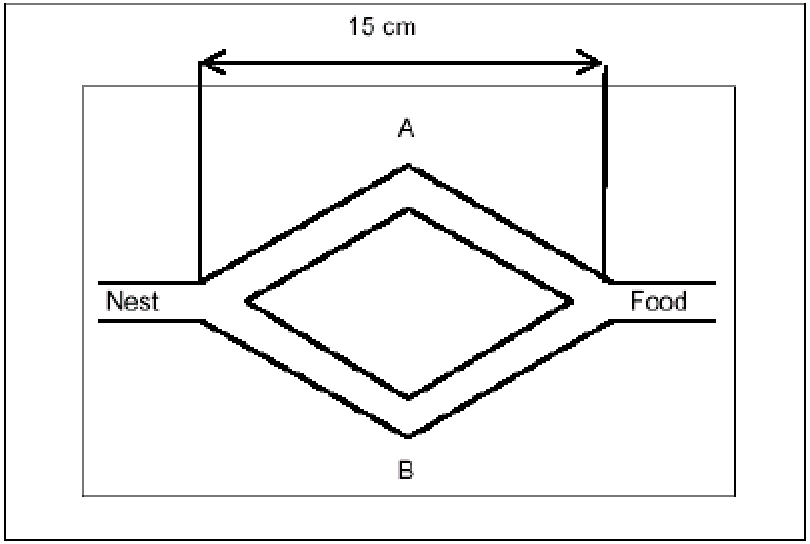


The Role of Randomness in the Organization of Foraging

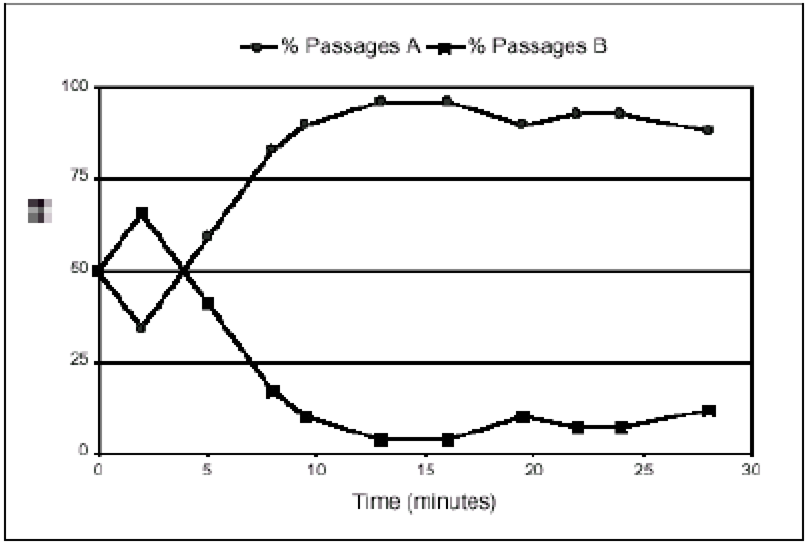
How does individual behavior with a strong stochastic component lead to **statistically predictable behavior** at the level of the colony and **collective decisions**?

2- path symmetric bridge

Goss et al., 1989, Deneubourg et al., 1990

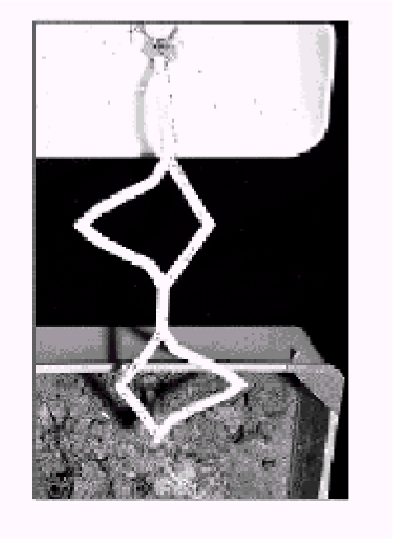


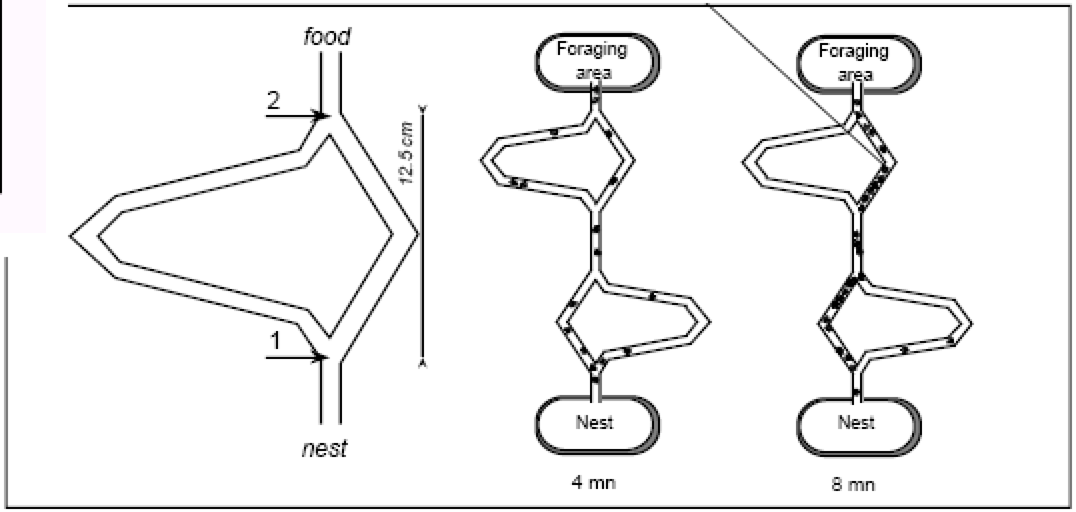
Simple bridge



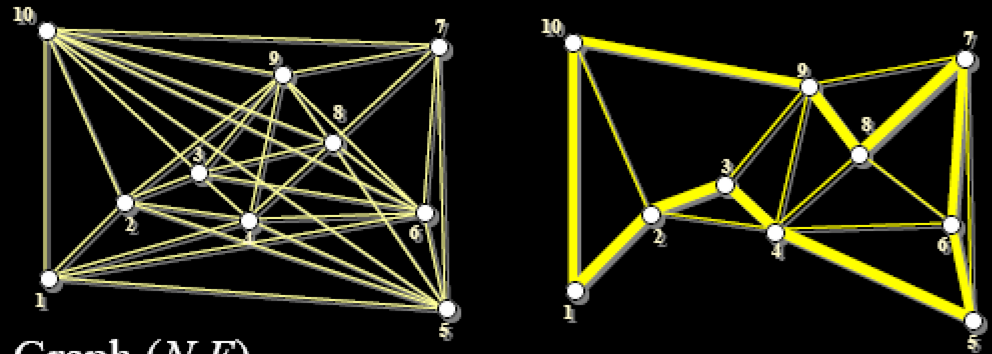
% of ant passages on the two branches

Constrained Optimization – 2 path asymmetric bridge





The Traveling Salesman Problem



Graph (N,E)
N : set of cities (nodes)
E : set of connecting roads (links)
d_{ij} : distance between city i and j

Problem: Find the shortest path which allow the salesman to visit once and only once each city in the graph

Difficulty: NP-hard problem; time for computing the shortest route grows in a nonpolinomial way with the number of cities in the network -> metaheuristics/machine-learning class (e.g., ACO, GA) provide near-optimal solutions!

How Hard are NP-Hard Problems?

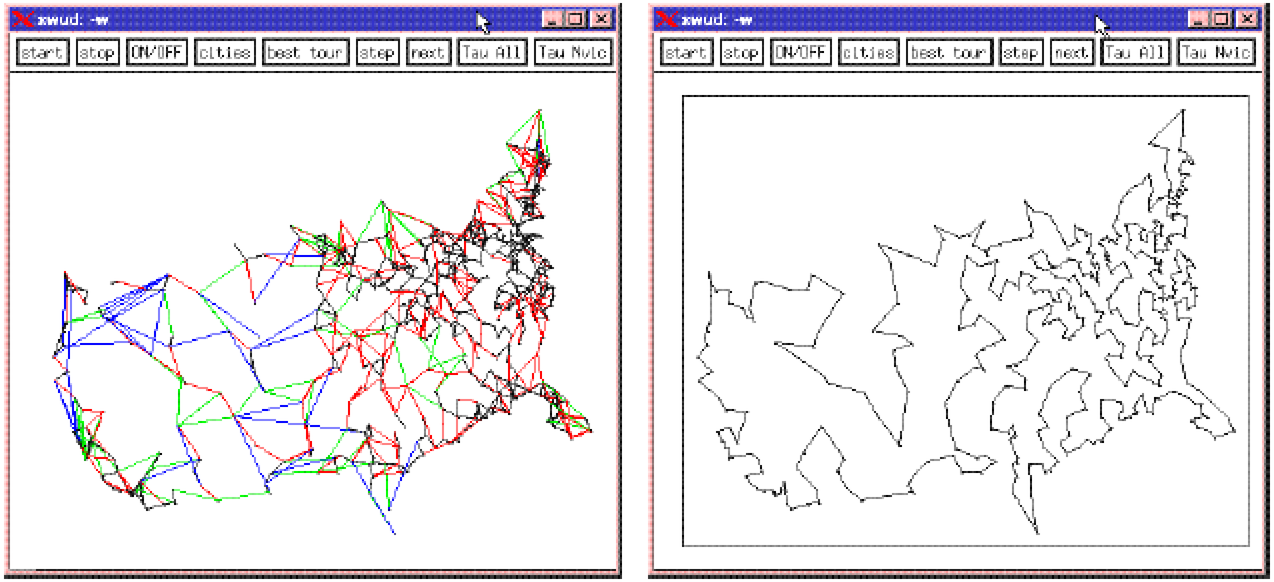
TSP – Brute force

- A 30 city tour would have to measure the total distance of be 2.65 X 10^{32} different tours. Assuming a trillion additions per second, this would take 252,333,390,232,297 years.
- Adding one more city would cause the time to increase by a factor of 31.

QAP – exact algorithms (e.g. Bixius & Anstreicher 2001)

- around 30+ max instances
- ex. 36 nodes (wiring application): 180 CPU on a 800 MHz Pentium III PC
- Same problem with ACO for QAP : 10 s on the same machine

ACS for TSP – Results on ATT532 Problem



Results

Problem name	ACS	GA	EP	SA	Optimum
Eil50 (50-city problem)	**425** (427.96) **[1,830]**	428 (N/A) [25,000]	426 **(427.86)** [100,000]	443 (N/A) [68,512]	**425** (N/A)
Eil75 (75-city problem)	**535** **(542.37)** **[3,480]**	545 (N/A) [80,000]	542 (549.18) [325,000]	580 (N/A) [173,250]	**535** (N/A)
KroA100 (100-city problem)	**21,282** **(21,285.44)** **[4,820]**	21,761 (N/A) [103,000]	N/A (N/A) [N/A]	N/A (N/A) [N/A]	**21,282** (N/A)

Results- Dorigo & Gambardella 1997

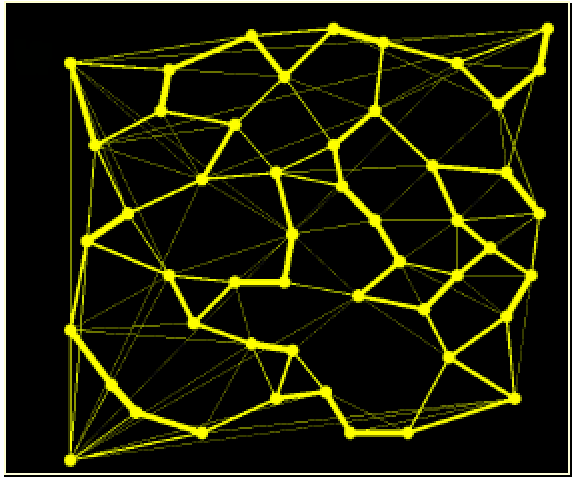
TABLE VIII

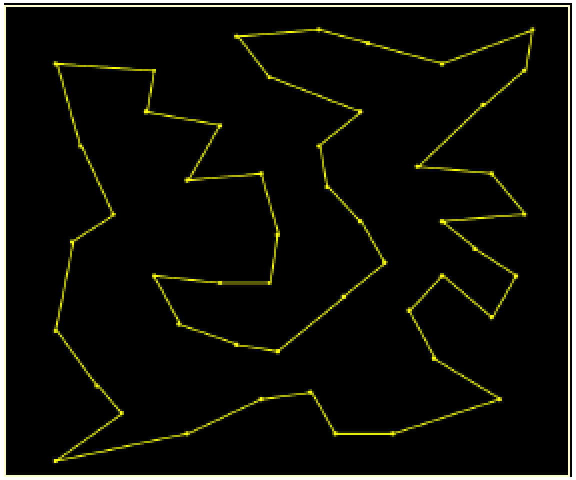
Comparison between ACS-3-opt and STSP-GA on TSP problems taken from the First International Contest on Evolutionary Optimization [5]. We report the average length of the best tour found, the average CPU time used to find it, and the relative error with respect to the optimal solution for both approaches.

Problem name	ACS-3-opt average (length)	ACS-3-opt average (sec)	ACS-3-opt % error	STSP-GA average (length)	STSP-GA average (sec)	STSP-GA % error
d198 (198-city problem)	15,781.7	238	0.01 %	15,780	253	0.00 %
lin318 (318-city problem)	42,029	537	0.00 %	42,029	2,054	0.00 %
att532 (532-city problem)	27,718.2	810	0.11 %	27,693.7	11,780	0.03 %
rat783 (783-city problem)	8,837.9	1,280	0.36 %	8,807.3	21,210	0.01 %

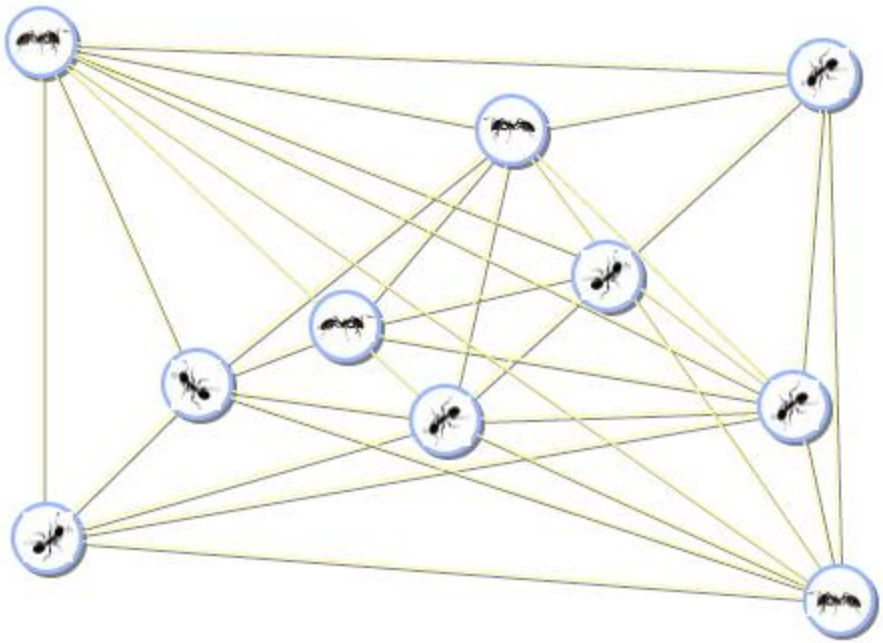
AS for TSP – Results 50 cities

Example of solution found on Eil50 problem





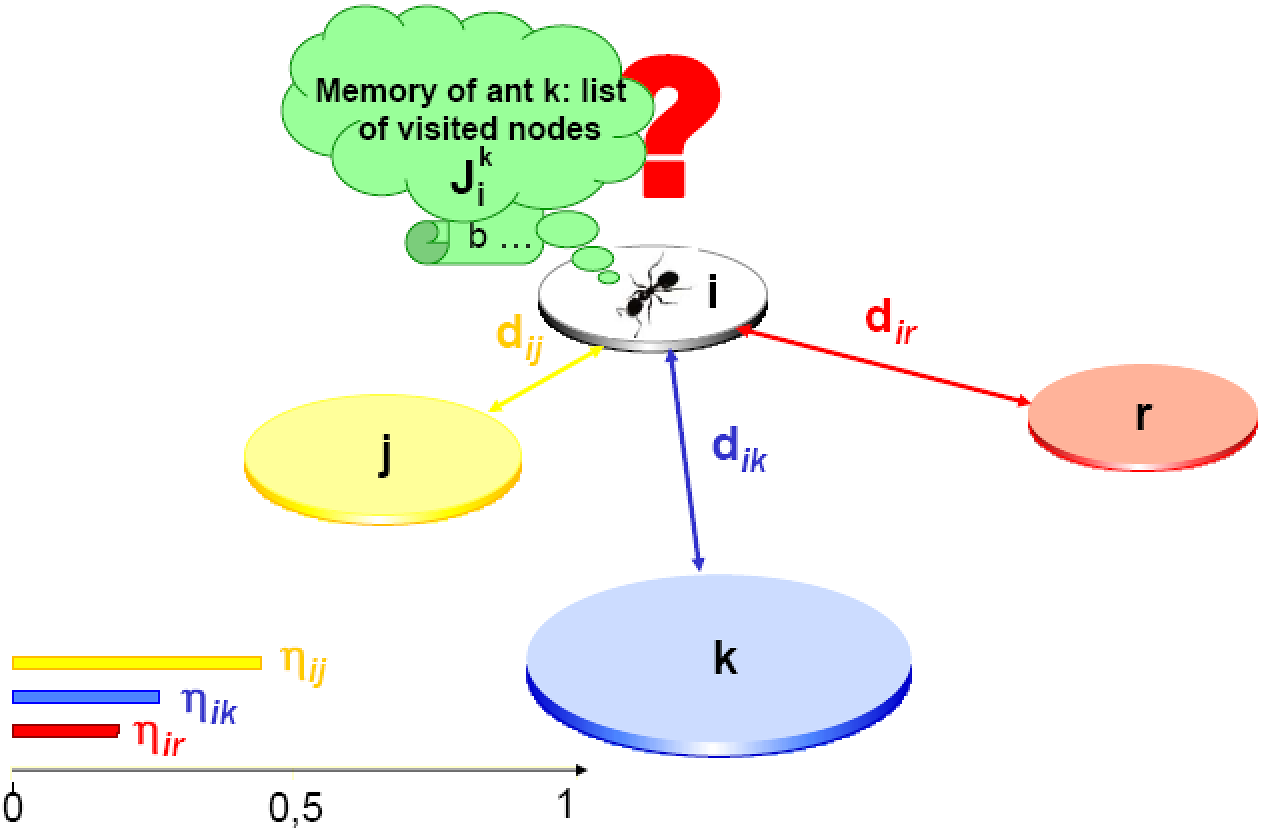
AS for TSP - Overview



$b_i(t)$, $(i = 1 \dots n)$: number of ants at the node i at the iteration t

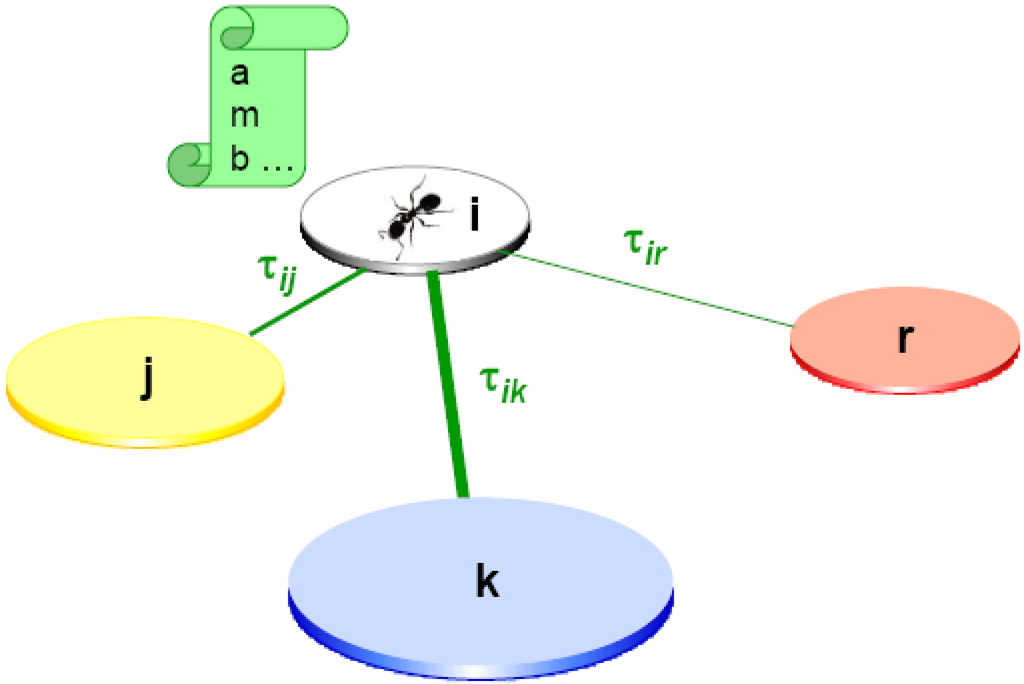
$m = \sum_{i=1}^{n} b_i(t)$ = constant: total number of ants

AS for TSP- Individual Ant Behavior



The inverted value of the distance $\eta_{ij} = 1/d_{ij}$ between nodes i and j is called visibility; this information (heuristic desirability) is static, i.e. not changed during the problem solution

AS for TSP- Individual Ant Behavior

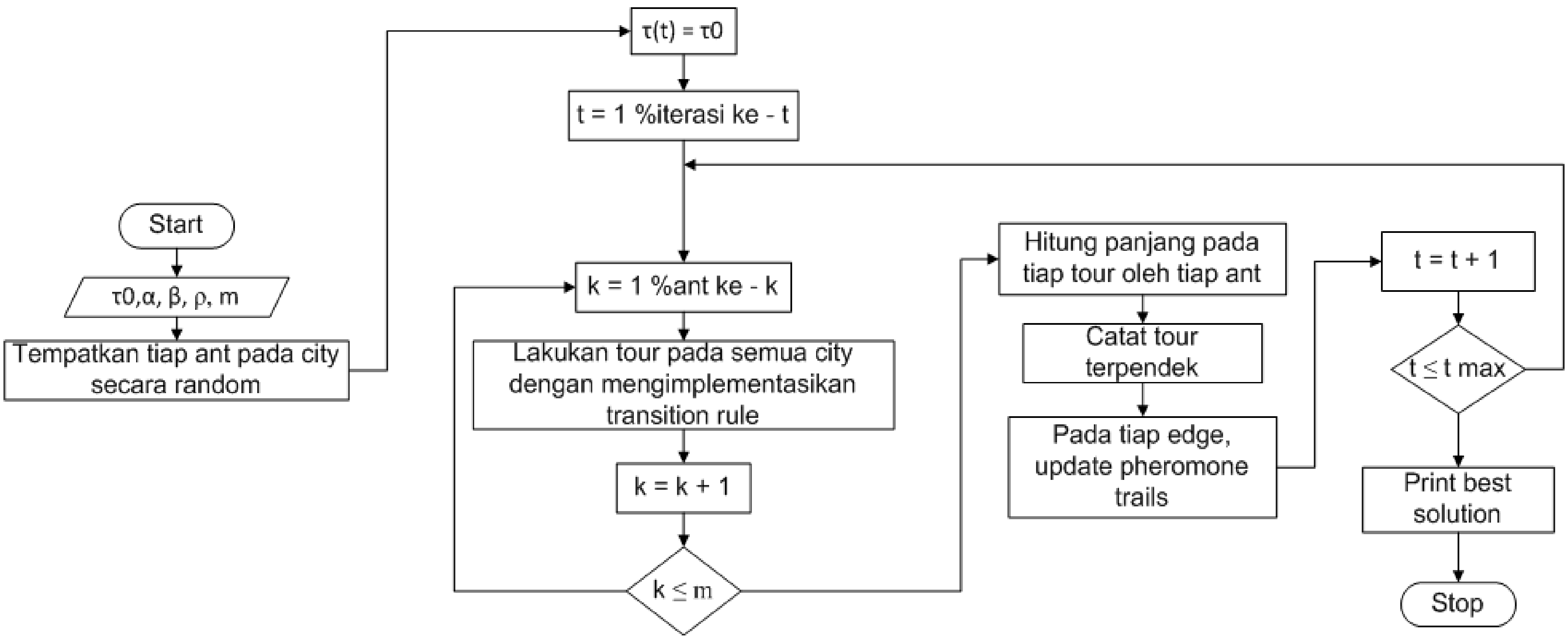


τ_{ij}, quantity of virtual pheromone deposited on the link between the node i and j

AS for TSP - Algorithm

Loop * t = 1*\
    Place one ant on each node *there are n = |N| nodes *
    For k := 1 to *m* * each ant builds a tour, in this case m=n*
        For step := 1 to *n* * each ant adds a node to its path *
            Choose the next node to move by applying a probabilistic ***state transition rule***
        End-for
    End-for
    Update pheromone trails
Until End_condition *t = t_{max} *\

Flow Chart Ant System Untuk TSP



/* Inisialisasi*/

For every edge(I,j) do

$\tau_{ij}(0) = \tau o$

End for

For k=1 to m do

Place ant k on a randomly chosen city

End For

Let T^{+} be the shortest tour found from beginning and L^{+} its length

/*main loop*/

For t = 1 to t_{max} do

For k = 1 to m do

Build tour $T^k(t)$ by applying n – 1 times the following step:

Choose the next city j with probability

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

Where i is the current city

End For

For k = 1 to m do

Compute the length $L^k(t)$ of the tour $T^k(t)$ produced by ant k

End For

If an improved tour is found then

Update T^+ and L^+

End if

For every edge (i,j) do

Update pheromone trails by applying the rule:

$\tau_{ij}(t) \leftarrow (1-\rho).\tau_{ij}(t) + \Delta\tau_{ij}(t) + e\Delta\tau_{ij}^e(t)$ *where*

$$\Delta\tau_{ij}(t) = \sum_{k=1}^{m} \Delta\tau_{ij}^k(t),$$

$$\Delta\tau_{ij}^k(t) = \begin{Bmatrix} Q/L^k(t) \text{ if } (i,j) \in T^k(t); \\ 0 \quad otherwise \end{Bmatrix} \text{ and } \Delta\tau_{ij}^e(t) = \begin{Bmatrix} Q/L^+(t) \text{ if } (i,j) \in T^+; \\ 0 \quad otherwise \end{Bmatrix}$$

End For

For every edge (i,j) do

$\tau_{ij}(t+1) = \tau_{ij}(t)$

End For

End For

Print the shortest tour T^+ and its length L^+

Stop

Kontruksi Ant System

1. Kontruksi Graph sebagai representasi masalah
2. Fungsi heuristik (η).
3. Probabilistik Transition Rule.
4. Pheromone update (τ).
5. Metode pemberhentian.

Fungsi Heuristik

- Jarak antar kota i dan j:

$$d_{ij} = [(x_i - x_j)^2 + (y_i - y_j)^2]^{1/2}$$

- Nilai heuristik (η):

$$\eta_{ij} = \frac{1}{d_{ij}}$$

Probabilistik Transition Rule

- Transition rule:

$$P_{ij}^{k}(t) = \frac{\left[\tau_{ij}(t)\right]^{\alpha}\left[\eta_{ij}\right]^{\beta}}{\sum_{t\in j_i^k}\left[\tau_{ij}(t)\right]^{\alpha}\left[\eta_{ij}\right]^{\beta}}$$

- Transition rule untuk TSP:

$$P_{ij}^{k}(t) = \frac{\left[\tau_{ij}(t)\right]^{\alpha}\left[1/d_{ij}\right]^{\beta}}{\sum_{t\in j_i^k}\left[\tau_{ij}(t)\right]^{\alpha}\left[1/d_{ij}\right]^{\beta}}$$

Dimana:

α dan β = parameter yang mengontrol bobot relatif.

$\tau_{ij}(t)$ = pheromone trail pada edge yang menghubungkan city i ke j.

Pheromone Update

- Formula:

$$\tau_{ij}(t) \leftarrow (1-\rho).\Delta\tau_{ij}(t) + e.\Delta\tau_{ij}^{e}(t)\ \mathrm{dim}\,ana$$

$$\Delta\tau_{ij}(t) = \sum \Delta\tau_{ij}^{k}$$

$$\Delta\tau_{ij}^{k}(t) = \begin{cases} Q/L^{k}(t), jika(i,j) \in T^{k}(t) \\ 0, sebaliknya \end{cases}$$

dan

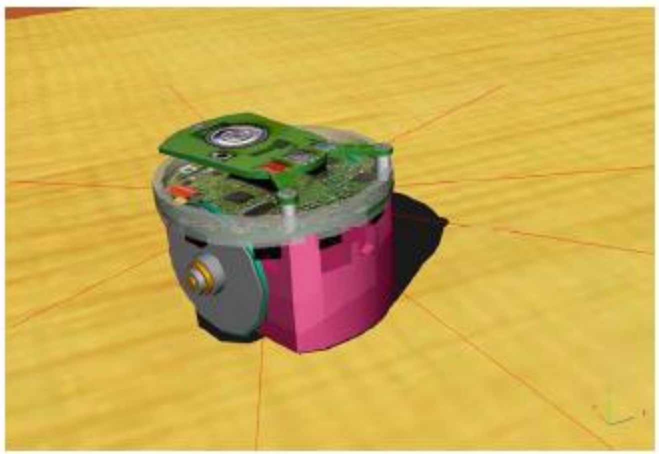
$$\Delta\tau_{ij}^{k}(t) = \begin{cases} Q/L^{+}, jika(i,j) \in T^{+} \\ 0, sebaliknya \end{cases}$$

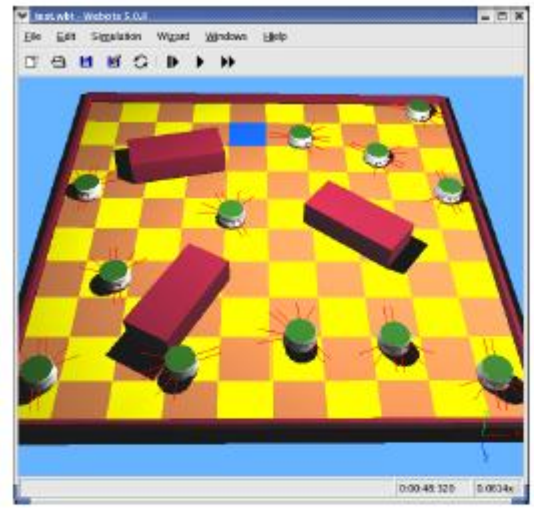
$$\tau_{ij}(t+1) = \tau_{ij}(t)$$

Catatan:
ρ = evaporation ([0..1]).
k = semut ke k.
t = iterasi ke t.
T^{+} = the best solution.
T^{k} = solusi terbaik pd suatu iterasi

Motivasi

E-Puck Simulated Robot





- Discrete sensor and actuators
- Noise and nonlinear characteristics faithfully reproduced
- Single and multi-robot simulator
- Different levels of simulation (different accuracy/faithfulness trade-offs)

Motivasi

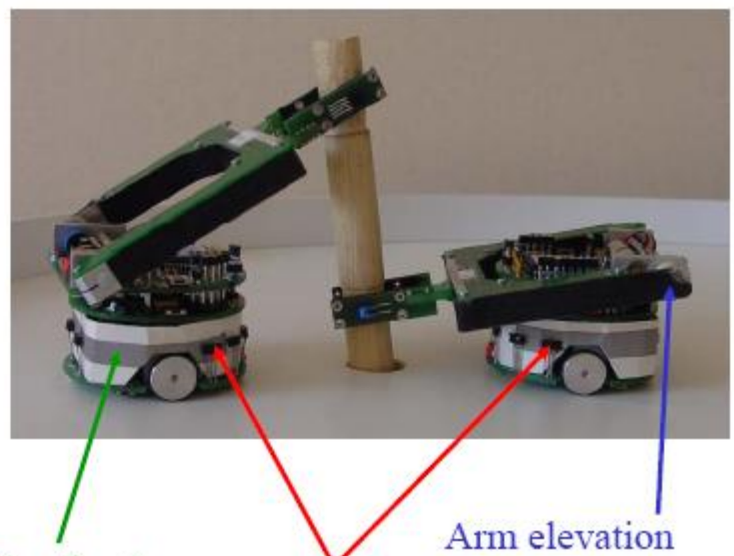
A Case Study: Stick-Pulling

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Physical Set-Up

- 2-6 robots
- 4 sticks
- 40 cm radius arena

Collaboration via indirect communication



Motivasi

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Boundary Coverage of Structures

- Case study: turbine inspection
- Goal: complete sensor **coverage** of the turbine/compressor blades
- Technical challenges **limit** possible designs of robotic sensors
- Test-bed: 40 Alice II
- Could pave the way for similar applications in coverage/inspection of engineered or natural, **regular structures** with heavily constrains on robotic equipment