Swarm Intelligence: From Natural to Artificial Systems

Eric Bonabeau, Marco Dorigo, and Guy Theraulaz

Introduction

- What is 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."

  ไม่ซับซ้อนมาก

  เป็นพฤติกรรมที่ทำร่วมกันของ agent กระทำกับ env.

- "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."

  (http://dsp.jpl.nasa.gov/members/payman/swarm/)

Some natural Swarm Intelligence systems

- **Ants**
  - find the **shortest path** to food
  - make **cemeteries**
  - **sort** their brood by size
  - etc.

- **Termites**
  - **build nests** with complex features like
    - fortified chambers
    - spiral air vents
    - fungus gardens
    - etc.

- **Bees**
  - gather pollen with high efficiency, exploiting the **nearest richest food source first**.

- **Geese**
  - **coordinate** takeoff and landing
  - flight **patterns**
  - etc.

- **Fish / Birds /…**
  - swarms

How Do Social Insects Coordinate Their Behaviour?

- communication is necessary

ผลลัพธ์จาก collective behavior ในการไป contact
คือ 'พื้นที่สะอาด'' เพราะ agent เก็บขยะ

- two types of communication
  - direct Agent สื่อสารตรง ๆ
    - antennation, trophallaxis (food or liquid exchange), mandibular contact, visual contact, chemical contact, etc.
  - indirect สื่อสารผ่าน env ไม่ต้องตะโกนบอก
    - two individuals interact indirectly when one of them modifies the environment and the other responds to the new environment at a later time

      called **<u>stigmergy</u>**

Stigmergy

- "La coordination des taches, la regulation des constructions ne dependent pas directement des oeuvriers, mais des constructions elles-memes. ***L'ouvrier ne dirige pas son travail, il est guidé par lui***. C'est à cette stimulation d'un type particulier que nous donnons le nom du **STIGMERGIE** (*stigma*, piqure; *ergon*, travail, oeuvre = oeuvre stimulante)." Grassé P. P., 1959

- ["The coordination of tasks and the regulation of constructions does not depend directly on the workers, but on the constructions themselves. ***The worker does not direct his work, but is guided by it***. It is to this special form of stimulation that we give the name **STIGMERGY** (*stigma*, sting; *ergon*, work, product of labour = stimulating product of labour)."] ขึ้นกับ env ที่ตัวเองทำงาน

Stigmergy

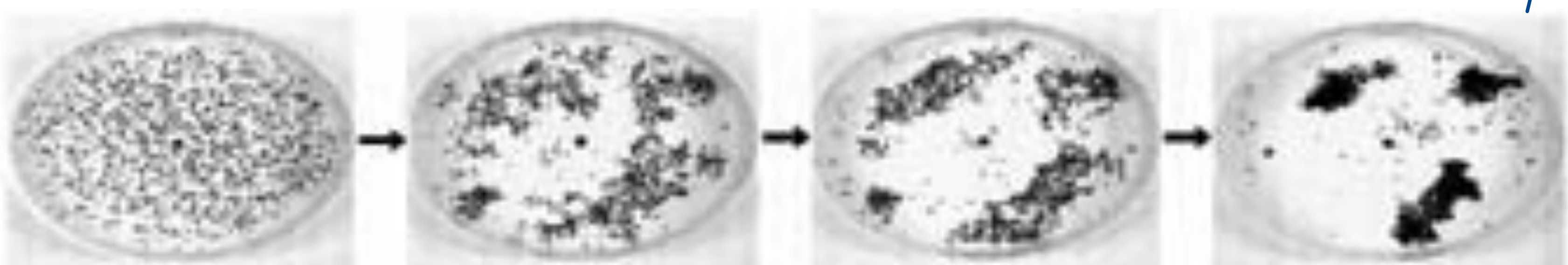
Communication through marks in the environment …
- Marks serve as a shared memory for the agents
- Promotes loose coupling
- Robust

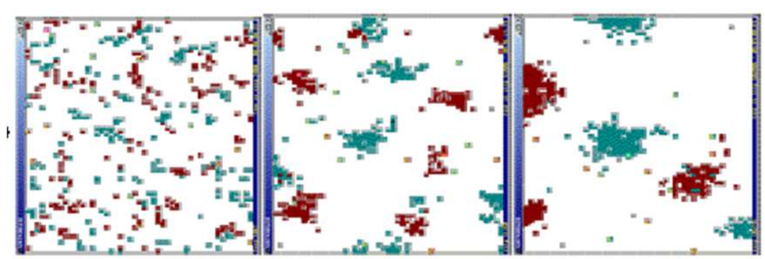
Stigmergy ~ action + sign:
- Agents put marks in the environment (inform other agents about issues of interest)
- other agents perceive these marks (influence their behavior)
- manipulate marks
- other agents perceive the marks
- etc.

Marks can be
- static (environment does not manipulate marks over time)
  - e.g., flags
- dynamic (environment manipulates marks over time) มีการเปลี่ยนไป ระดับเปลี่ยนไปตามเวลา
  - e.g., pheromones, gradient fields

Example of Stigmergy: Ant cemeteries in nature

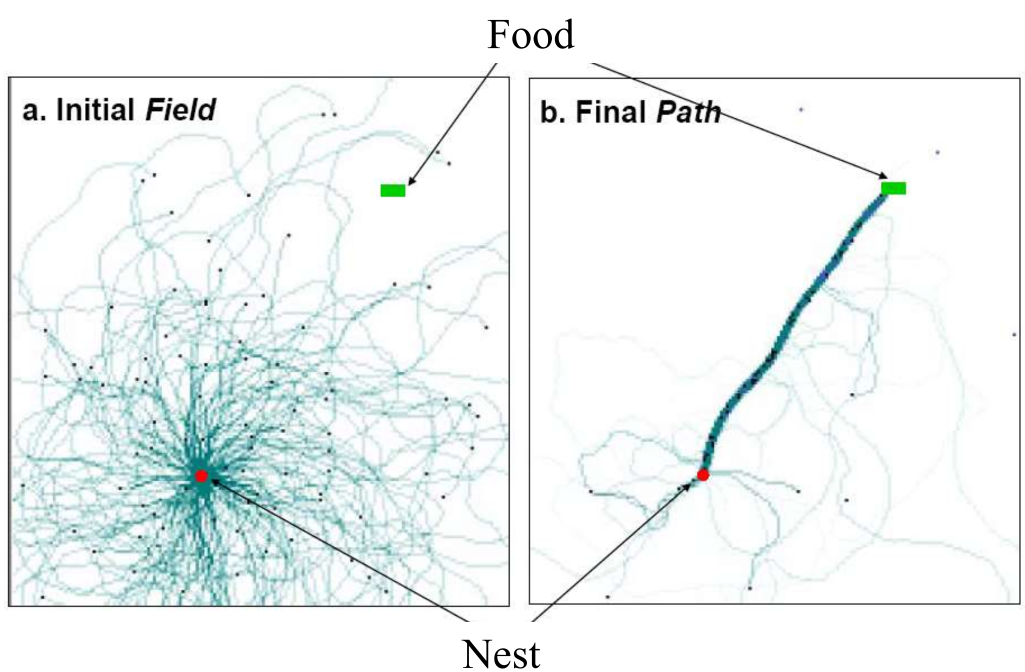
- simple behaviour rules for ants
  - wander around การเดินแบบกระจัดกระจายใน env
  - if you find a dead ant
    pick it up with probability inversely proportional to the number of other dead ants nearby โอกาสที่จะแบกมดที่ตายมีสูง แปรผกผัน กับ จ.น.มดที่ตาย
  - if you are carrying a dead ant
    put it down with probability directly proportional to the number of other dead ants nearby โอกาสที่วางมดลงแปรผันตรงกับ จ.น.มดที่ตายบริเวณนั้นๆ





Example of Stigmergy: Trail Following and Ants Foraging Behaviour

- while walking, ants and termites
  - may deposit a pheromone on the ground
  - follow with high probability pheromone trails they sense on the ground



“Artificial” Stigmergy

***Indirect communication** mediated by modifications of environmental states which are only **locally accessible** by the communicating agents*

Dorigo & Di Caro, 1999

- **Characteristics of artificial stigmergy:** !!!!!
  - **Indirect communication** สื่อสารกับ env
  - **Local accessibility** access ข้อมูลบริเวณใกล้เคียง

Foraging Strategies in Ants

- The Binary Bridge Experiment (Page 27)

  The ants choose one branch over the other due to some random fluctuations.

P_A = โอกาสที่มดเดินทาง A

- Probability of choosing one branch over the other ~

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

- The values of k and n determined through experiments.

  k = degree of attraction of an unmarked branch

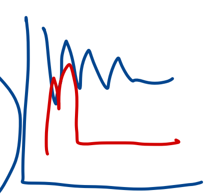
  n = choice function

n มาก การหนีเมื่อรับ เลือกทางจะมาก

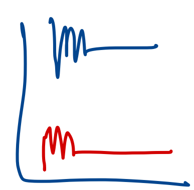
แต่ k มีผลตอนเลือกทางช่วงแรก ต้องสิ้นไปข้างล่าง –

ถ้า k มาก

n น้อย



n มาก



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

กรณี $k = 0$: $P_A = \dfrac{A_i^n}{A_i^n + B_i^n}$

มดตัว 1 $\dfrac{0^1}{0^1 + 0^1} = A$

มดตัว 2 $\dfrac{1^2}{1^2 + 0^2} = \dfrac{1}{1} = 1 = A$

3 $\dfrac{2^2}{2^2 + 0^2} = \dfrac{4}{4} = 1 = A$

กรณี $k = 5$: $\dfrac{(5 + A_i)^2}{(5 + A_i)^2 + (5 + B_i)^2}$

มด 1 $= \dfrac{5}{5 + 5} = \dfrac{1}{2} = A$

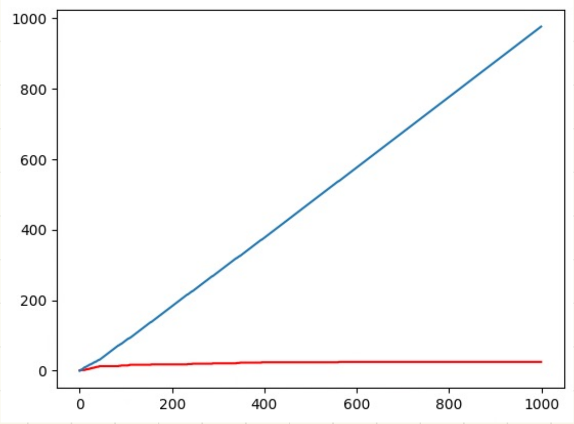
2 $= \dfrac{(5 + 1)^2}{(5+1)^2 + (5+0)^2} = \dfrac{36}{36 + 25} = \dfrac{36}{61} = B$

3 $= \dfrac{(5+1)^2}{(5+1)^2 + (5+1)^2} = \dfrac{36}{36} = \dfrac{1}{2} = A$

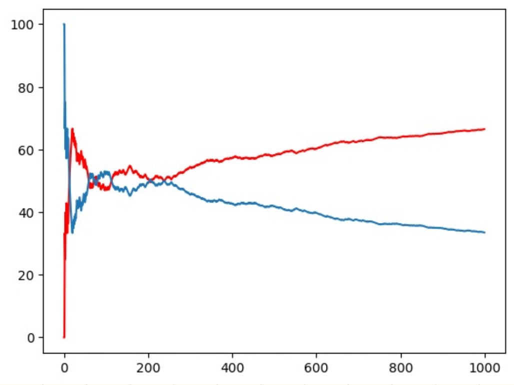
k มาก การ dominated จะน้อยลง สลับไปมาได้มากขึ้นเรื่อยๆ

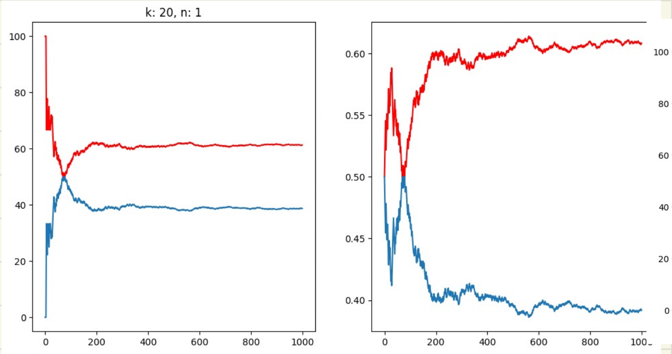
```
k = 5
n = 2
A = 0       As = []
B = 0       Bs = []

for i in range(1000):

    Pa = (k+A)**n / ((k+A)**n + (k+B) n)        A ・0.7

    r = random.random()        r คือตัดสินใจว่าจะไป A

    if r <= Pa:          ถ้าไม่ใช้ r ใช้ PB = 1 - PA หรือ PA > 0.5
        A += 1                 มันจะไม่มีการสุ่มไหมคะ เหมือน softmax
    else :
        B += 1
    As. append(A)
    Bs. append(A)
```

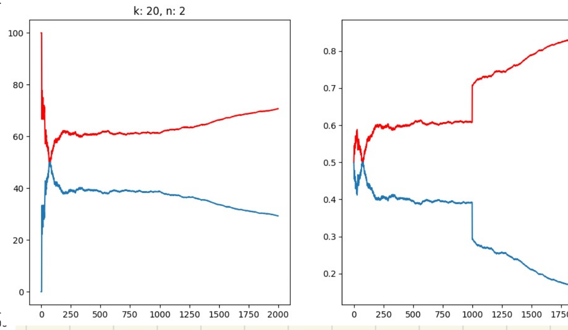
```
plt.plot([i for i in range(len(As))], As, 'r')
plt.plot([i for i in range(len(Bs))], Bs, '
```

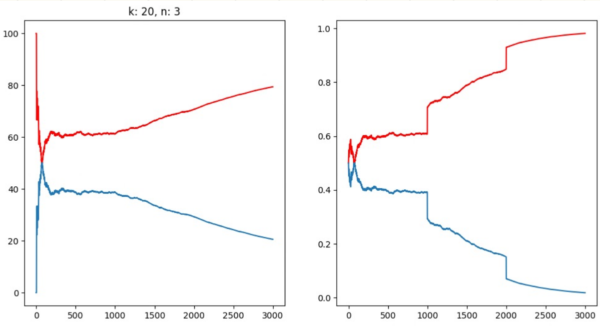


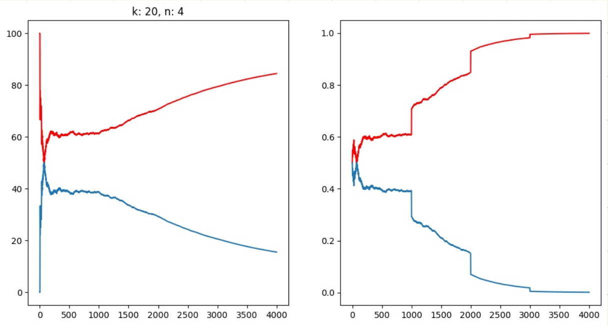
กราฟการคิด % $\frac{A*100}{A+B}$

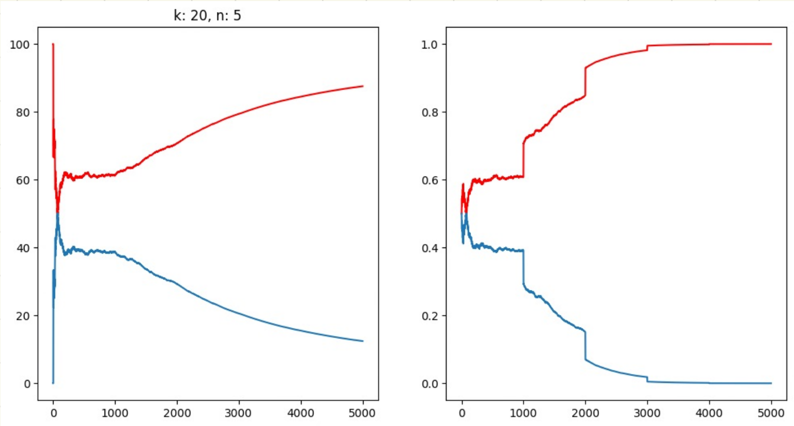



k: 20, n: 1


k: 20, n: 2


k: 20, n: 3


k: 20, n: 4


k: 20, n: 5

Foraging Strategies in Ants

- Ants deposit pheromone on the paths that they cover and this results in the building of a solution (optimal path).

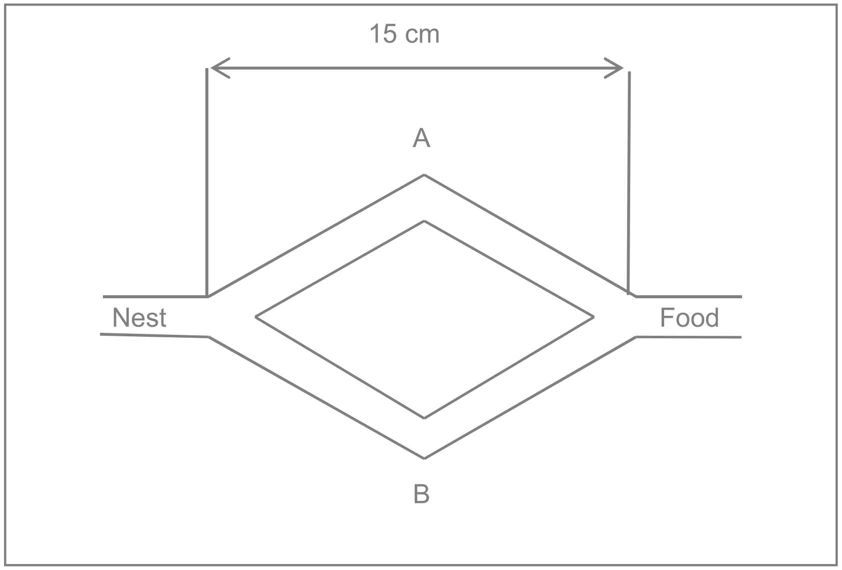
- In SI and optimization, concept of pheromone evaporation is used.
- Helps in avoiding suboptimal solutions – local optima.
- May differ from how it takes places in the real world.

Foraging Strategies in Ants

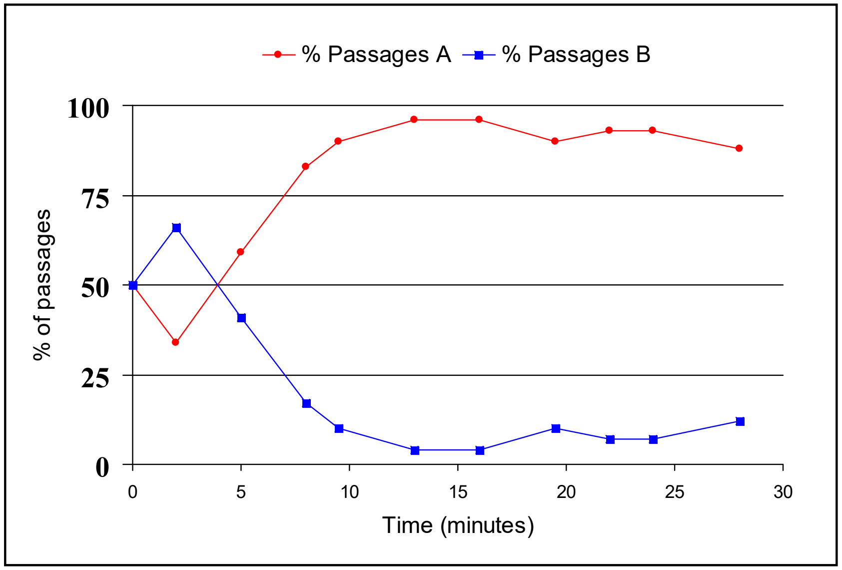
- Inter-nest Traffic studied – a case of natural optimization
- Similarity with MST shown by Aron et al.
- Other experiments done – effect of light vs dark, chemical vs visual cues.

- Conclusion here: some colonies have networks of nests several hundreds of meters in span – it is possible this is close to a MST.

Ants Foraging Behaviour
Example: The Double Bridge Experiment



Simple bridge



% of ant passages on the two branches

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

Raid Patterns of Army Ants

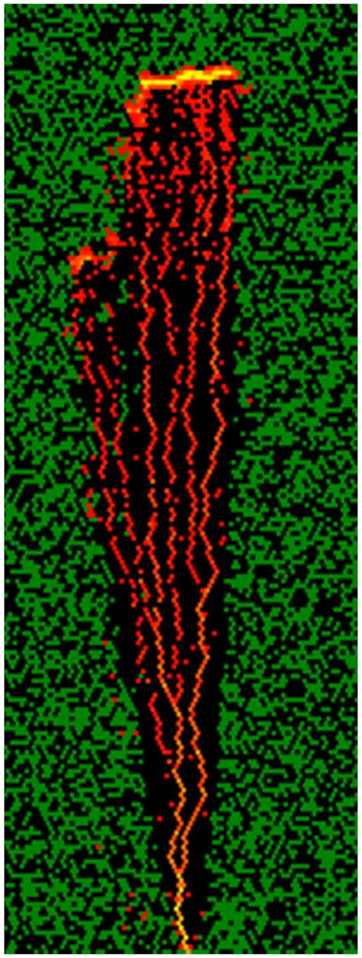
- An example of powerful, totally decentralized control.
- Example : Eciton burchelli can consist of as many as 200,000 workers.
- These individuals are blind, communication via pheromone.

Raid Patterns of Army Ants

- 3 species of ants have a common ancestor.
- Can the foraging behavior be explained through a different environment in each case?
- Deneubourg et al. modeled the behavior of these ants.
- Used a 2-D grid
- Had several rules like:
- 1 ant deposits 1 unit of pheromone per each visited site while returning to its nest.
- Maximum number of ants per site

Raid Patterns of Army Ants

- Pheromone disappearance rate at each site
- Movement of an ant from one site to the other based on a probabilistic mechanism shown earlier.
- Particular food distribution in the network
- A well-defined raid pattern is observed.
- Some similarity with the actual observations.



Ant Colony Optimization (ACO)

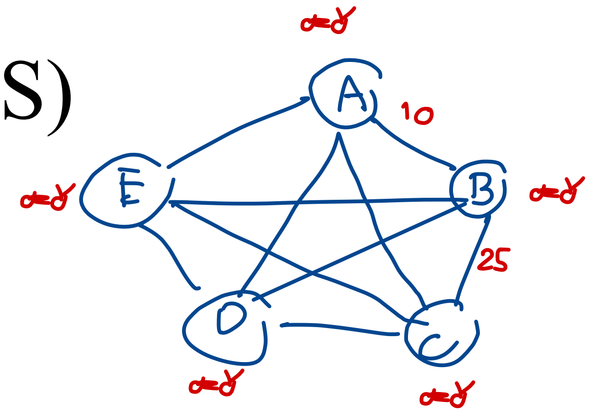
- We now come to more rigorous mathematical models.
- TSP has been a popular problem for the ACO models.
  - several reasons why TSP is chosen…..

- 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.

① ปล่อยมดเท่า จ.น. เมือง

ปัญหาคือ เมื่อเจอปัญหาใหม่ ต้องนิยาม algorithm ใหม่ สร้างมดใหม่

Ant System (AS)



- Used to solve TSP
- Transition from city i to j depends on:

1. Tabu list – list of cities not visited list ของเมืองที่ยังไม่เคยไป
2. Visibility = $1/d_{ij}$; represents local information – heuristic desirability to visit city j when in city i.
3. Pheromone trail $T_{ij}(t)$ for each edge – represents the learned desirability to visit city j when in city i. ระดับ $\tau_{i \to j} \neq \tau_{j \to i}$

- Generally, have several ants searching the solution space.

  m = n

Ant System (AS)

ถ้า τ มาก η มาก => อยากไป

τ น้อย η น้อย => ไม่อยากไป

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

ระดับ τ eta $\eta_{ij} = \dfrac{1}{d_{ij}}$ ระยะทาง

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

- Alpha and beta are adjustable parameters.

Ant System (AS)

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

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

Ant System (AS)

- Pheromone update :

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

ถ้า L มาก Q / L น้อย ก็เติมฟีโรโมนน้อย

L น้อย —น— มาก เติมเยอะ

- T is the tour done at time t by ant k, L is the length, Q is a heuristic parameter. ระยะทางเดินของมดตัวนั้น
- Pheromone decay: การระเหยของ Phyromone

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

ระดับการระเหย

เช่น 20% ก็เป็น 1 - 0.2

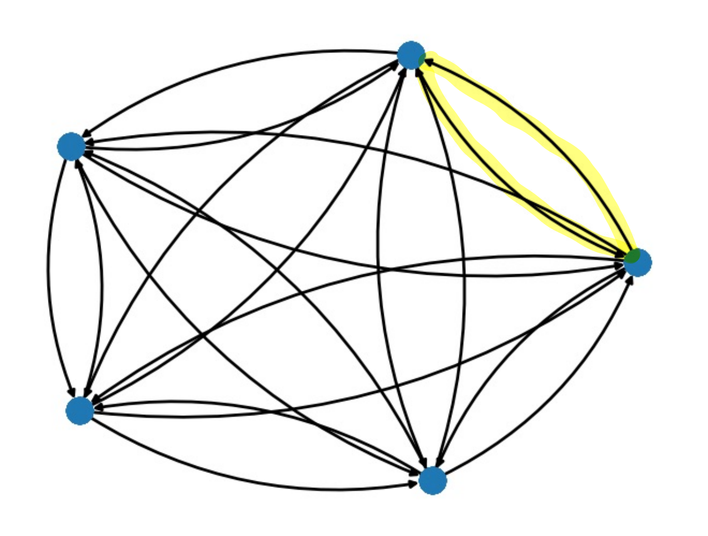
มดเดินผ่าน จะมีการเติมฟีโรโมนเพิ่ม

เช่น A → B เดินผ่าน

ต้องเติม $\Delta\tau_{AB}$ ทุกตัว

$D_c + D_d + D_e$

ทางที่ใช้ฟีโรโมน ไป-กลับ ไม่เท่ากัน $D_{ij} \neq D_{ji}$



เติม pheromone ทุกสเต็ปเดินหลัง update τ

Ant System (AS)

- Modifications to the algorithm:
- Elitist scheme borrowed from GA
- Use the elitist to update its own tour (T+) edges for pheromone deposition.
- Could extend the same concept to “e” elitists ants.

- Results …..?
- Does not perform as well as other methods – the ones mentioned are TS (Tabu Search) and SA.

Ant System (AS)

- Does not converge to a single solution – is that a good criteria?

- However, they conclude that the "nonconvergence" property is interesting –

1. It tends to avoid trappings in local optima.
2. Could be used for dynamic problems.

- So next …..ACS

Ant Colony System (ACS)

$$j = \arg\max_{u \in J_k^i} \{[\tau_{ij}(t)].[\eta_{iu}]^{\beta}\} \; if \; q \le q_o \; j = J$$

- Modifications to AS.
- New transition rule:

  q_o is a parameter that can be tweaked
- It is similar to tuning temperature in SA.
- J is a city randomly selected according to the probability calculated previously.
- This helps ACS to improvise on the best solutions.

Ant Colony System (ACS)

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

- Pheromone update rule (new):
- However, only applied to the best ant.
- The change in the pheromone concentration = 1/L+.

- Local updates done as follows:

$$\tau_{ij}(t) = (1-\rho).\tau_{ij}(t) + \rho\tau_0$$

Ant Colony System (ACS)

- To improves its search methodology, uses a candidate list of cl closest cities, considers these first, considers other cities only when the list is exhausted.
- Example cl = 15 on Page 51.

- ACS-TSP has been applied on problems of various sizes.
- ACS-TSP has been shown to be superior over other methods like GA, SA, EP for problems of size 50 – 100 cities.
- For larger size problems………

Ant Colony System (ACS)

- Use a local search method in conjunction with ACS-TSP.
- Called as 2-opt, 3-opt – refers to the number of edges exchanged iteratively to obtain a local optima.

- Has been shown to be comparable to the best techniques available (GA).

- Other methods for improvement-
- Elitism, worst tours (pheromone removed), local search enhancement.

The Quadratic Assignment Problem (QAP)

- Find pi such that the following is minimized:

$$C(\pi) = \sum_{i,j=1}^{n} d_{ij} f_{\pi(i)\pi(j)}$$

- QAP has shown to be NP-hard.
- d's are the distance between the nodes and f's are the flows between nodes.
- The problem is similar to TSP.
- distance potentials and flow potentials.

The Quadratic Assignment Problem

- Associate the minimum total flow at a node with the maximum total potential and so on : min-max coupling rule.
- This is a good heuristic, but does not give the optimal results.

- Hence AS-QAP proposed.
- The transition rule – the probability that the kth ant chooses activity j as the activity to assign to location i is:

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

The Quadratic Assignment Problem

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

- Same pheromone update rule as AS-TSP.
- Here the change is equal to $Q/C^k(t)$ though – hence low coupling (C) value means a stronger pheromone trail.

- Results :
- GA, ES < AS-QAP < TS, SA

- Improvements…..

Hybrid Ant System (HAS)

- Departs radically from previously described ACO algorithms.
- Three procedures:

1. Pheromone-trail-based modification
2. Local search
3. Pheromone trail updating

…..kind of the same idea as ACS.

Hybrid Ant System (HAS - QAP)

$$p_{ij}^{k} = \frac{\tau_{i\pi^{k}(j)} + \tau_{j\pi^{k}(i)}}{\sum_{l=1}^{n} (\tau_{i\pi^{k}(l)} + \tau_{l\pi^{k}(i)})}$$

- Over here, each ant represents a solution like in GA, SA etc.
- It moves to another solution by applying R swaps.
- Example R = n/3.
- And the probability of moving from one point in solution space to the other is given above.

Hybrid Ant System (HAS - QAP)

- Local search:
- After a new solution is obtained, do a local search to get a lower point in solution space.
- This point may not necessarily be the local optima (why?)

- Pheromone-trail updating is done as follows:

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

- Here the change at each time step = 1/C(pi)+.

Hybrid Ant System (HAS - QAP)

- Intensification – keeping new best solutions in memory and replacing the current ones with them; again similar to elitism.

- Diversification: All pheromone trail values are reinitialized if no improvement is made in S generations – example S = n/2.
- How does HAS-QAP perform ?
- The results are that it performs comparable to other methods.
- However, it does not do so well for regular problems – reason?
- Does good for problems that have a irregular structure.

Other 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
- Shortest common supersequence

PRECAI => Big AI conf.

Particle Swarm Optimisation. คล้าย GA

GA มีประชากรที่รอดไปยังรุ่นต่อไป

PSO ประชากรไมมีกันตาย

Particle Swarm Optimization

- Particle Swarm Optimization (PSO) mimics the collective intelligent behavior of " unintelligent " creatures.
- It was developed in 1995 by James Kennedy and Russell Eberhart [Kennedy, J. and Eberhart, R. (1995). "Particle Swarm Optimization", *Proceedings of the 1995 IEEE International Conference on Neural Networks*, pp. 1942-1948, IEEE Press.] (http://dsp.jpl.nasa.gov/members/payman/swarm/kennedy95-ijcnn.pdf)
- It has been applied successfully to a wide variety of search and optimization problems.
- In PSO, a swarm of *n* individuals communicate either directly or indirectly with one another in each search directions (gradients).

Particle Swarm Optimization
The Anatomy of a Particle

- A particle (individual) is composed of:
  - Three vectors:
    - The **x-vector** records the current position (location) of the particle in the search space,
    - The **p-vector** records the location of the best solution found so far by the particle, and ตำแหน่งของผึ้งที่ผ่านมาที่ดีที่สุด
    - The **v-vector** contains a gradient (direction) for which particle will travel in if undisturbed. ค.เร็วของทาง ($\vec{v} + \vec{x}$)
  - Two fitness values: ค่าความดี
    - The **x-fitness** records the fitness of the x-vector, and ต.น.ของ x ที่ดีสุด
    - The **p-fitness** records the fitness of the p-vector. fitness ที่ดีสุดของ local best

g-finess => fitness global best สามารถแชร์ global best ร่วมกันได้

$$\mathbf{I_k}$$

$X = \langle x_{k0}, x_{k1}, \ldots, x_{kn-1} \rangle$ state, คำตอบ

$P = \langle p_{k0}, p_{k1}, \ldots, p_{kn-1} \rangle$ local best ที่เก็บตำแหน่งที่ดีที่สุด

$V = \langle v_{k0}, v_{k1}, \ldots, v_{kn-1} \rangle$

x_fitness = ?

p_fitness = ?

Particle Swarm Optimization
Swarm search

Velocity calculation

โมเมนตัม เหมือนการวางแผน pheromone เช่น จอ ง่อ คงเหลือ

$$v_{id(t)} = \omega \underbrace{v_{id(t-1)}}_{\text{v รอบที่แล้ว}} + \underbrace{c_1 \times rand() \times \left(p_{id} - x_{id}\right)}_{\text{ใช้ local best ของมันตัวนั้นๆ}} + \underbrace{c_2 \times Rand() \times \left(p_{gd} - x_{id}\right)}_{\text{ใช้ global best}}$$

Position update

$$x_{id(t)} = x_{id(t-1)} + v_{id(t)}$$

update ท.น. นี้

x_{id} – current value of the dimension “d” of the dividual “i”

v_{id} – current velocity of the dimension “d” of the individual “i”.

P_{id} – optimal value of the dimension “d” of the individual “i” so far.

P_{gd} – current optimal value of the dimension “d” of the swarm.

c_1, c_2 – acceleration coefficients.

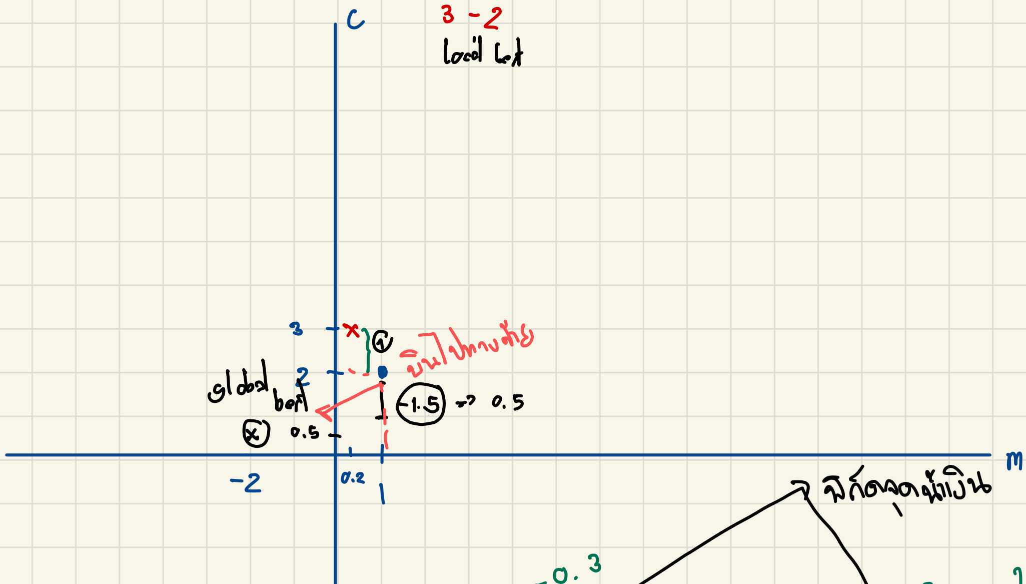
ω - inertia weight factor

$$v_{id(t)} = \omega v_{id(t-1)} + \underbrace{c_1 \times rand() \times (p_{id} - x_{id})} + \underbrace{c_2 \times Rand() \times (p_{gd} - x_{id})}$$

$y = mx + c$

m	c

x	1	2



v	0.1	0.2

-0.3   -3

ลบแกน x

$$V = 0.8\begin{bmatrix}0.1\\0.2\end{bmatrix} + 2(0.6)\left(\begin{bmatrix}0.2\\3\end{bmatrix} - \begin{bmatrix}1\\2\end{bmatrix}\right) + 2(0.4)\left(\begin{bmatrix}-2\\0.5\end{bmatrix} - \begin{bmatrix}1\\2\end{bmatrix}\right)$$

ลบแกน y

บินทางซ้าย

v มาก มีโอกาสบินข้ามจุด ซึ่ง control โดยค่า c trace_2 เก็บค่าความดี

ยังพยายามบินโดยมี
local best เรื่อยๆ

INERTIA = 0.1
C_1 = 0.8 // local
C_2 = 0.005 // global

ผึ้งบินวนแถวๆ จุดเริ่มต้น

INERTIA = 0.1
C_1 = 0.005
C_2 = 0.8

เริ่มมีการขยับ สุดท้ายอยู่ที่จุดสูงสุด
แต่ยังมีตัวที่บินมั่วๆ

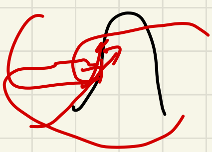
0.8
0.005
0.8

บินวนไม่หยุด

0.8
0.05
0.05

ไม่ได้มาตรงกลาง ยังเอียงมาข้างนึง

เพิ่ม ω เพื่อบินสำรวจ อยากได้แค่
ω น้อยๆ



ยังมาข้างเขา
ลงจุดต่ำอีกฝั่ง

C มาก มีโอกาสบินหลุดกราฟ

ถ้าอยาก explore ในกราฟ ต้องทำให้มันเยอะมากๆ

ถ้า c_1 มาก

จะต้องบินกลับ local ตัวเอง

Particle Swarm Optimization
Swarm Search

- In PSO, particles never die!
- Particles can be seen as simple agents that fly through the search space and record (and possibly communicate) the best solution that they have discovered.
- Initially the values of the velocity vectors are randomly generated with the range [-Vmax, Vmax] where Vmax is the maximum value that can be assigned to any v_{id}.
- Once the particle computes the new Xi and evaluates its new location. If x-fitness is better than p-fitness, then $P_i = X_i$ and p-fitness = x-fitness.

Particle Swarm Optimization
The algorithm

1. Initialise particles in the search space at random.
2. Assign random initial velocities for each particle.
3. Evaluate the fitness of each particle according a user defined objective function.
4. Calculate the new velocities for each particle.
5. Move the particles.
6. Repeat steps 3 to 5 until a predefined stopping criterion is satisfied.