SWINBURNE UNIVERSITY OF TECHNOLOGY

COS30018 – Intelligent Systems. Week 8: Collective Intelligence/Swarm Intelligence

1

Nature-inspired computing

- Nature has always served as a source of inspiration for engineers and scientists
- The best problem solver known in nature is:
  - the (human) brain that created "the wheel, New York, wars and so on" (after Douglas Adams' Hitch-Hikers Guide)
  - the evolution mechanism that created the human brain (after Darwin's Origin of Species)
- Answer 1 → neurocomputing
  - Artificial Neural Networks (Week 6)
- Answer 2 → evolutionary computing
  - Genetic Algorithms and Evolutionary Computing (Week 7)
  - Swarm Intelligence (Today)

2 / 39

2

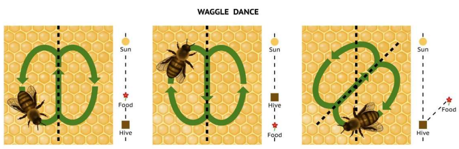
Outline

- Swarm Intelligence
- Introduction to Particle Swarm Optimization (PSO)
  - Origins
  - Concept
  - PSO Algorithm
- Introduction to Ant Colony Optimization (ACO)
  - Origin & Concept
  - ACO Algorithm

3

School of fish or flock of birds
COORDINATION??

4



Waggle dance is one of the main types of **communication** methods used by bees

5

Long trail of ants
Coordinated behaviour
(Communication-based??)

6

Swarm intelligence

- Collective system capable of accomplishing difficult tasks in dynamic and varied environments:
  - NO external guidance or control
  - NO central coordination
- Achieving a **collective performance** which could not normally be achieved by an individual acting alone

Source: http://www.scs.carleton.ca/~arpwhite/courses/95590Y/notes/SI Lecture 3.pdf

7

Swarm intelligence

- Constituting a natural model particularly suited to **distributed problem solving**
  - Particle Swarm Optimisation (**PSO**) — a way to solve optimisation problems, based on the swarming behaviour via direct communication.
  - Ant Colony Optimisation (**ACO**) – a different way to solve optimisation problems based on the way that ants indirectly communicate directions to each other.

Source: http://www.scs.carleton.ca/~arpwhite/courses/95590Y/notes/SI Lecture 3.pdf

8

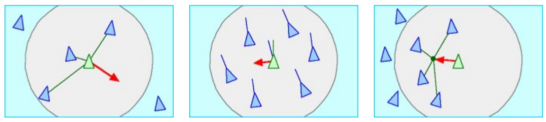
Introduction to the PSO: Origins

- Inspired from the nature social behavior and dynamic movements with communications of insects, birds and fish

9

Introduction to the PSO: Origins

- In 1986, Craig Reynolds described this process in 3 simple behaviors:



Separation
avoid crowding local flock mates

Alignment
move towards the average heading of local flock mates

Cohesion
move toward the average position of local flock mates

10

Introduction to the PSO: Origins

- Application to optimization: Particle Swarm Optimization
- Proposed by James Kennedy & Russell Eberhart (1995)
- Combines **self-experiences** with **social experiences**

11

Introduction to the PSO: Concept

- Many **particles** (called **agents** in PSO) constituting a swarm "fly" around in the search space looking for the best solution
- Each particle in search space adjusts its "flying" based on:
  - its position,
  - its own flying experience, AND
  - the flying experience of other particles

12

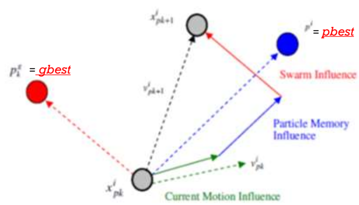
Introduction to the PSO: Concept

- Collection of flying particles (swarm) - Changing solutions
- Search area - Possible solutions
- Movement towards a promising area to get the global optimum
- Each particle keeps track:
  - its best solution, personal best, *pbest*
  - the best value of any particle, global/**neighborhood** best, *gbest*

13

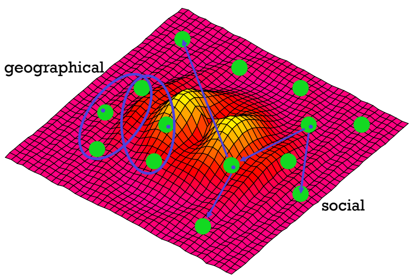
Introduction to the PSO: Concept

- Each particle modifies its position according to:
  - its current **position**: x^i_{pk}
  - its current **velocity**: v^i_{pk}
  - the vector between its current position and *pbest*
  - the vector between its current position and *gbest*



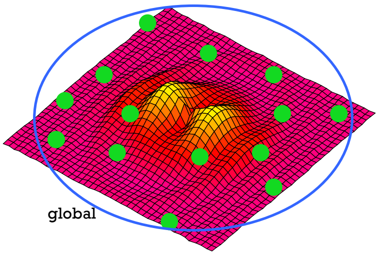
14

Introduction to the PSO: Algorithm - Neighborhood



15

Introduction to the PSO: Algorithm - Neighborhood



16

Introduction to the PSO: Algorithm - Parameters

- Algorithm parameters:
  - ***A*** : Population of agents (**agent = particle**)
  - $\boldsymbol{p_i}$: Position of agent $\boldsymbol{a_i}$ in the solution space ($\boldsymbol{a_i}$.Pos)
  - ***f*** : Objective function
  - $\boldsymbol{v_i}$: Velocity of agent's $\boldsymbol{a_i}$ ($\boldsymbol{a_i}$.Vel)
  - $\boldsymbol{V(a_i)}$: Neighborhood of agent a_i ($\boldsymbol{a_i}$.NB)

17

Introduction to the PSO: Algorithm

```
[x*] = PSO()
[aj.Pos] = Particle_Initialization();
For i=1 to it_max
  For each particle a in A do
    a.fp = f(a.Pos);
    If a.fp is better than f(a.pBest)
        a.pBest = a.Pos;
    end
  end
  For each particle a in A:   a.gBest = best p in a.NB;
  For each particle a in A do
     a.Vel = a.Vel + c1*rand*(a.pBest – a.Pos)
                   + c2*rand*(a.gBest – a.Pos);
     a.Pos = a.Pos + a.Vel;
  end
end
```

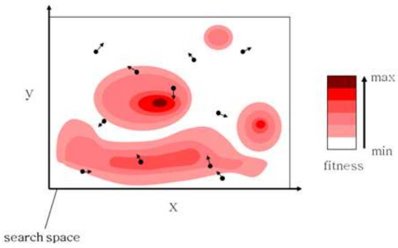
18

Introduction to the PSO: Algorithm

- Particle update rule
  a.Pos = a.Pos + a.Vel
  - with
    a.*Vel* = a.*Vel + c1*rand**(a.*pBest* – *a.Pos*)
    + *c2*rand**(a.*gBest* – *a.Pos*)
  - where
    - *a.Pos* : particle's position
    - a.*Vel* : path direction
    - *c1* : weight of local information
    - *c2* : weight of global information
    - a.*pBest* : best position of the particle
    - a.*gBest* : best position of the (neighbouring) swarm
    - *rand* : random variable

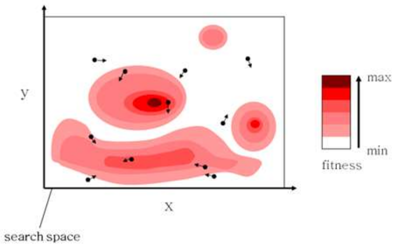
19

Introduction to the PSO: Algorithm - Example



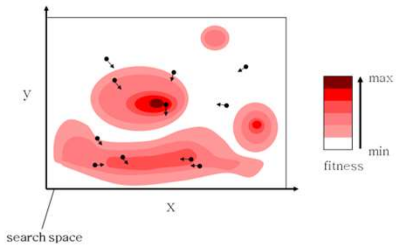
20

Introduction to the PSO: Algorithm - Example



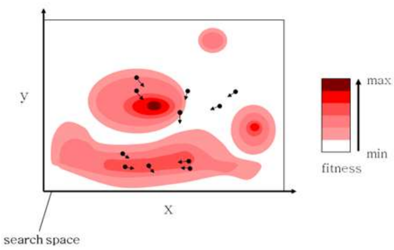
21

Introduction to the PSO: Algorithm - Example



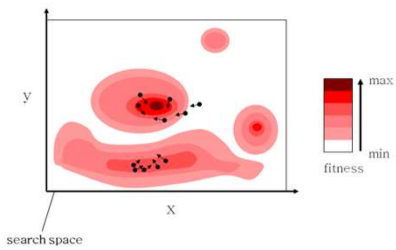
22

Introduction to the PSO: Algorithm - Example



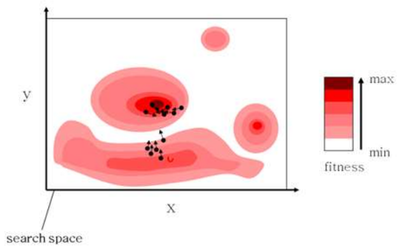
23

Introduction to the PSO: Algorithm - Example



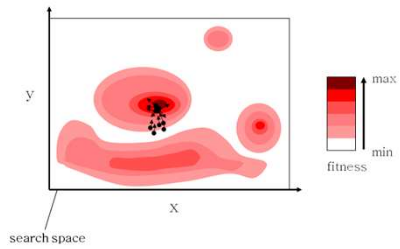
24

Introduction to the PSO: Algorithm - Example



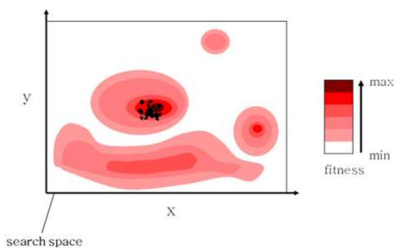
25

Introduction to the PSO: Algorithm - Example



26

Introduction to the PSO: Algorithm - Example



27

PSO: Applications

- **Suitable for problems whose solutions can be mapped to an R^n space.**
  - Energy management: what is a user's optimal consumption profile given the multiple objectives: Minimizing cost, maximizing comfortability, minimizing environmental damages, etc.
  - Aircraft surface design: safety vs cost vs practicality
  - Vehicle routing problems
  - Structural engineering problems
  - … and even
    - weights of an ANN

28

Presentation Outline

- Swarm Intelligence
- Introduction to Particle Swarm Optimization (PSO)
  - Origins
  - Concept
  - PSO Algorithm
- Introduction to Ant Colony Optimization (ACO)
  - Origin & Concept
  - ACO Algorithm

29

Collective intelligence as emergent property of many individuals operating simple rules

For *Lasius Niger* ants, [Franks, 89] observed:

- regulation of nest temperature within 1 degree celsius range;
- **forming bridges;**
- raiding specific areas for food;
- building and protecting nest;
- **sorting brood and food items;**
- **cooperating in carrying large items;**
- emigration of a colony;
- **finding shortest route from nest to food source;**
- preferentially exploiting the richest food source available.

The ACO algorithm is inspired by this:

30

A key concept: Stigmergy

Stigmergy is:

indirect communication via interaction with the environment [Gassé, 59]

- A problem gets solved bit by bit ..
- Individuals communicate with each other in the above way, affecting what each other does on the task.
- Individuals leave *markers* or *messages* – these don't solve the problem in themselves, but they affect other individuals in a way that helps them solve the problem …
  - e.g. as we will see, this is how ants find shortest paths.

31

Stigmergy in Ants

Ants are behaviorally unsophisticated, but collectively they can perform complex tasks.

Ants have *highly developed sophisticated sign-based* ***stigmergy***

- They communicate using pheromones;
- They **lay trails of pheromone** that can be followed by other ants.

- If an ant has a **choice of two pheromone trails** to follow, one to the NW, one to the NE, but the NW one is *stronger* – which one will it follow?

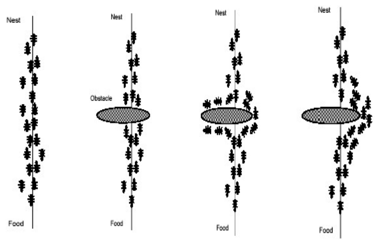
32

ACO Concept: Pheromone Trails

- Individual ants lay pheromone trails while travelling from the nest, to the nest or possibly in both directions.
- The pheromone trail gradually evaporates over time.
- But pheromone trail strength accumulate with multiple ants using path.

33

ACO Concept: Pheromone Trails



Ant Algorithms – (P.Koumoutsakos – based on notes L. Gamberdella (www.idsia.ch)

34

Ant Colony Optimisation Algorithms: Basic Ideas

- Ants are *agents* that interact with its environment (and leave its information on the env.)
- Typical environment in ACO is a graph.

Ants:

Move along between nodes in a graph.

They choose where to go based on pheromone strength (and maybe other things)

An ant's path represents a specific candidate solution.

When an ant has finished a solution, pheromone is laid on its path, according to quality of solution.

This pheromone trail affects behaviour of other ants by `stigmergy' …

35

Some nice online demo:

- http://www.theprojectspot.com/tutorial-post/ant-colony-optimization-for-hackers/10

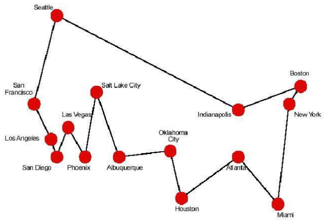
36

Travelling Salesman Problem (TSP)

TSP PROBLEM : Given N cities, and a distance function d between cities, find a tour that:

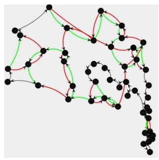
1. **Goes through every city once and only once**
2. **Minimizes the total distance.**

- Problem is NP-hard
- Classical combinatorial optimization problem to test.

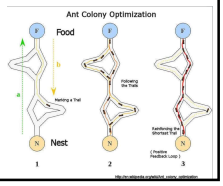


37

ACO: Algorithm



```
[x*] = ACO()
  [a_i State] = Ant_Colony_Initialization();
  while not terminated:
    For each ant a in A do:
      a.generateSolutions();
      a.transitionToNewState();
    end
    pheromoneUpdate();
  end
```



38

ACO for the Traveling Salesman Problem – generateSolution()

$$p_{xy}^k = \frac{(\tau_{xy}^{\alpha})(\eta_{xy}^{\beta})}{\sum_{z \in \mathrm{allowed}_y} (\tau_{xz}^{\alpha})(\eta_{xz}^{\beta})}$$

Note: Ants don't have to start from the same city.
-> *Parallelized computation*.

p_{xy}^k: The probability that the k^{th} ant currently at city x would move to (*allowable*) city y

τ_{xy}: the amount of pheromone deposited for transition from x to y

η_{xy}: the heuristics for how desirable it is to go from x to y (*e.g.*, $\frac{1}{d_{xy}}$, where d_{xy} is distance between x and y)

α and β parameters to control the influence of τ_{xy} and η_{xy}, respectively.

39

ACO for TSP – Pheromone update

$$\tau_{xy} \leftarrow (1-\rho)\tau_{xy} + \sum_{k}^{m} \Delta\tau_{xy}^k$$

$$\Delta\tau_{xy}^k = \begin{cases} Q/L_k & \text{if ant } k \text{ uses curve } xy \text{ in its tour} \\ 0 & \text{otherwise} \end{cases}$$

τ_{xy}: the amount of pheromone deposited for transition from x to y

ρ : the **pheromone evaporation** coefficient

m: the number of ants, and

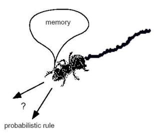
$\Delta\tau_{xy}^k$: the amount of pheromone deposited on link xy by the k^{th} ant.

Q: a constant

L_k: cost for the route taken by the k^{th} ant (*e.g., total distance of that route)*

40

ACO State Transition Rule



Next city is chosen between the **not visited** cities according to a *probabilistic* rule

Exploitation: the best edge is chosen

Exploration: each of the edges in proportion to its value

41

Summary

- Swarm intelligence:
  - Achieving a collective performance which could not normally be achieved by an individual acting alone: **PARALLELIZING THE SEARCH**
- PSO has a memory (for storing *pbest* and *gbest*)
- There is no selection in PSO
  - → all particles survive for the length of the run
  - → PSO is the only EA that does not remove candidate population members
- PSO is: Simple in concept, easy to implement, and computationally efficient
- ACO is an approach for solving hard **combinatorial optimization problems**.
- Artificial ants implement a randomized construction heuristic which makes probabilistic decisions.
- The accumulated search experience is taken into account by the adaptation of the pheromone trail.

42