ECM3412/ECMM409
Nature Inspired Computation
Lecture 12

Swarm Intelligence 2: Particle Swarm Optimisation

Today

- Brief recap of the principles of Swarm Intelligence
- Recap of the important points of flocking behaviour
- Particle Swarm Intelligence Algorithm
- Some applications

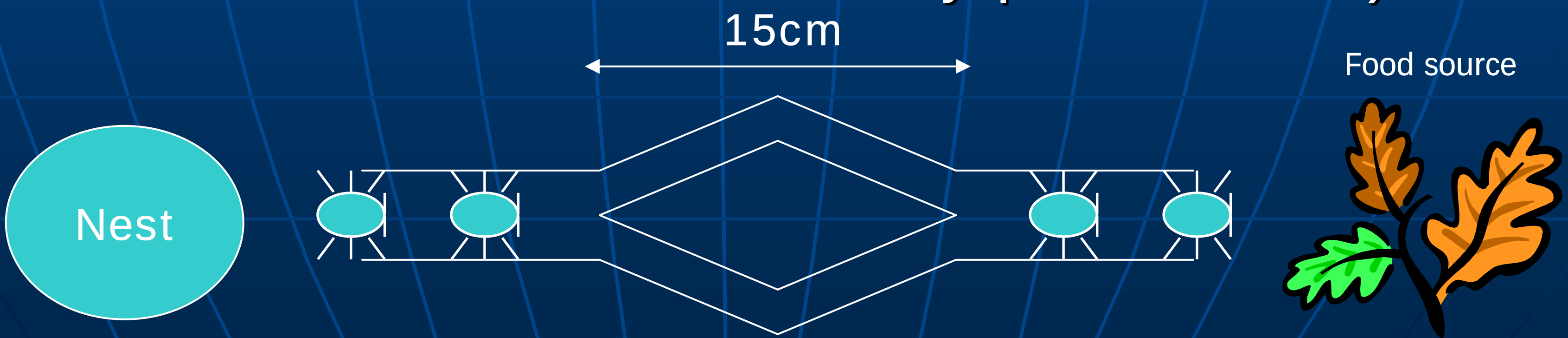
Swarm Intelligence Recap

- Emergent ‘intelligent’ behaviour not possible by an individual

- Arises as a consequence of interactions between individuals in an environment

- Interaction can be direct (through vision etc...)

- Or indirect such as ants and stigmergy (the modification of the environment by pheromones)



Ultimate Swarm Intelligence?

- Ultimate swarm intelligence? (Levine, 1998)
  - Slime-mold (*Dictyostelida and Acrasida*)
  - Food abundant – mold is an amoeba
  - Food scarce – turns into multicellular organism (slug) capable of movement...

Slime mold *Physarum*

Characteristics of Flocking

Rapid directed movement of the whole flock

Reactivity to predators (flash expansion, fountain effect)

Reactivity to obstacles

No collisions between flock members

Coalescing and splitting of flocks

Tolerant of movement within the flock, loss or gain of flock members

No dedicated leader

Different species can have different flocking characteristics – easy to recognise but not always easy to describe

Benefits of Flocking: Energy Saving

V-formations in birds:

- geese flying in Vs can extend their range by over 70%
- each bird rides on the vortex cast off by the wing-tip of the one in front
- individual geese fly 24% faster than flocks

Particle Swarm Optimisation

- Introduced by Kennedy and Eberhardt in 1995
- Particle – each particle is a solution
  - No mass or volume
  - BUT! Velocity and acceleration do apply
- Swarm – velocity matching not included
- Optimisation – discovery of near-optimal solution by means of a population of individuals

The ‘Cornfield Vector’ / Rooster Effect

- How does swarming/flocking get us to a near optimal solution?
- When food is left for birds, often within minutes/hours there are a number of birds at that food source (similar to our optimal solution).
- Therefore we can define a ‘fitness function’ which represents the quality of a solution.
- As we will see later, the knowledge of individuals is incorporated into the algorithm.

Basic PSO

Create and initialize N random particles (solutions)on the search space

For Each Timestep

For Each Individual

Update the position of the particle by adding a velocity to the current particle position

Next Individual

Next Timestep

Position Updating

- How do we decide on a new position for a particle?
- Based on its velocity like this:

$$x_{ij}[t+1] = x_{ij}[t] + v_{ij}[t+1]$$

Where x is the position of particle i in dimension (variable) j

Velocity Updating

- How do we determine the velocity of a particle?
- By using this function:

$$v_{ij}[t+1] = v_{ij}[t] + c1 * rand() * (pbestx_{ij} - presentx_{ij}) + c2 * rand() * (gbestx_{ij} - presentx_{ij})$$

- Where c1 & c2 are constants and rand() is a function which returns a random number between 0 and 1.
- As before i is the individual and j is the dimension.
- pbestx is that particle's previous best position
- gbestx is the population's previous best position
- It's not as complicated as it looks!

Velocity Updating

- Break the function into sections:
- $v_{ij}[t+1] = v_{ij}[t]$
  - New particle velocity for particle i in dimension j is equal to the old velocity plus...
- $+c1*rand()*(pbestx_{ij}-presentx_{ij})$
  - A random weighting of the previous best encountered position for individual i minus the current position, plus...
- $+c2*rand()*(gbestx_{ij}-presentx_{ij})$
  - A random weighting of the previous best encountered position for the whole population minus the current position.

Velocity Updating

- Why rand()?
  - As with GAs, to stop the swarm converging too quickly.
- Why c1 & c2?
  - These constants can be used to change the weighting between personal and population experience.
- Why pbest?
  - This is the component which draws individuals back to their previous best situations
- Why gbest?
  - This is the component where individuals compare themselves to others in their group

PSO and Social Behaviour

- Kennedy and Eberhardt discuss PSO in the context of human social behaviour

- The *pbest* parameter is the cognitive component:
  - How well the individual is doing based on performance in the past.
- The *gbest* parameter is the social component
  - How well the individual is doing based on the performance of other individuals.

Variations – Neighbourhood Based PSO

- Similar to the previous global-based PSO
- *lbest* computed instead of *gbest* where lbest is the best solution in the current neighbourhood
- Neighbourhoods are normally not defined by closeness of decision variables
  - *No matching required*
  - *Good solutions are spread throughout the population*
- *gbest v lbest?*
  - *gbest will tend to converge earlier as every particle has access to the best individual*
  - *lbest has larger diversity and less likely to get trapped in local minima*

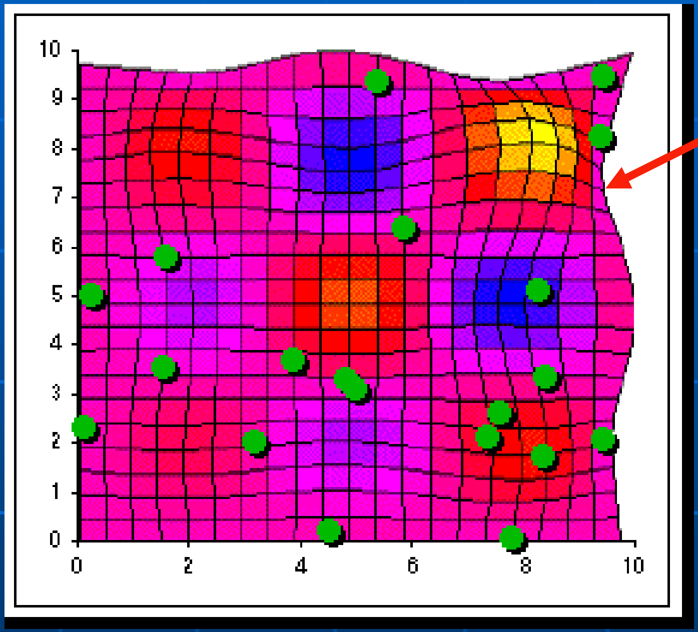
PSO Variables

- Swarm size
  - Analogous to population size in GAs
- Neighbourhood size
  - *lbest* (as opposed to gbest) algorithm uses a neighbourhood to calculate social component
  - Neighbourhood size a variable
- Number of iterations
- Acceleration coefficients (c1 & c2)
  - Manage the relationship between pbest and gbest/lbest
  - Low values may mean slow progress
  - High values mean the algorithm may miss good solutions

PSO Termination Criteria

- Max Iterations
- Acceptable solution found
- No improvement for N iterations
- Normalized swarm radius close to zero
- Compute the slope of fitness for the global best - is it flat?

PSO Example1

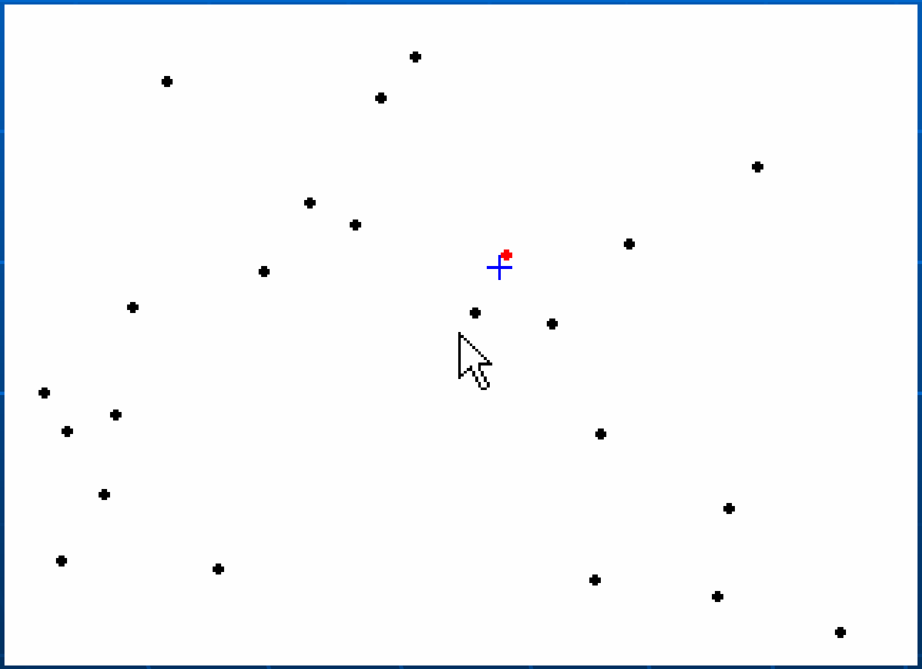


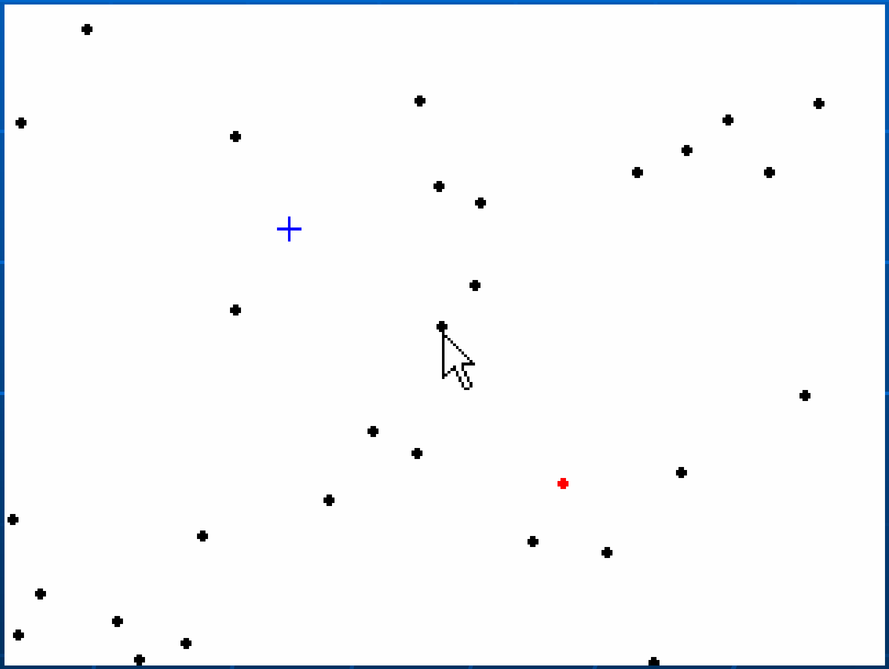
Global optimum

- Credit: Maurice Clerc

PSO Example2

- Simple Problem (De Jong F1)
- More difficult problem (De Jong F5)





PSO Applications

- Theoretically any optimisation task that EAs can be applied to. PSO has been applied to:
  - Design of:
    - Antennae, aircraft wings, amplifiers, controllers, circuits etc...
  - Scheduling and planning of:
    - Operational planning, task assignment, TSP

PSO Summary

- Based on the principles of flocking/swarming
- Has a simple function to update the velocities of particles and therefore create new solutions
- Has an intuitive set of parameters
- Can be modified to be used on discrete and dynamic optimisation problems