Swarm Intelligence

Particle Swarm Optimization

Leslie Pérez Cáceres
leslie.perez.caceres@ulb.ac.be
Hayfa Hammami
haifa.hammami@ulb.ac.be

IRIDIA – Université Libre de Bruxelles (ULB)
Bruxelles, Belgium

Particle Swarm Optimisation

- Perturbative algorithm
  - Initial solutions must be provided
- Originally proposed for continuous problems
  - Adaptations have been propose to tackle mixed and discrete problems
- Particles share information
  - Topologies

Particle Swarm Optimisation

- Solutions: particle moving through the search space
  - Position $x[\,]=(x_1, x_2, \ldots, x_n)$
  - Velocity $v[\,]=(v_1, v_2, \ldots, v_n)$
  - Personal best position $p[\,]=(p_1, p_2, \ldots, p_n)$
  - Global best position $g[\,]=(g_1, g_2, \ldots, g_n)$

$$v_i(t+1)=v_i(t)+\psi_1 * u_1(p_i - x_i(t))+\psi_2 * u_2(g_i - x_i(t))$$

$$x_i(t+1)=x_i(t)+v_i(t+1)$$

Particle Swarm Optimisation

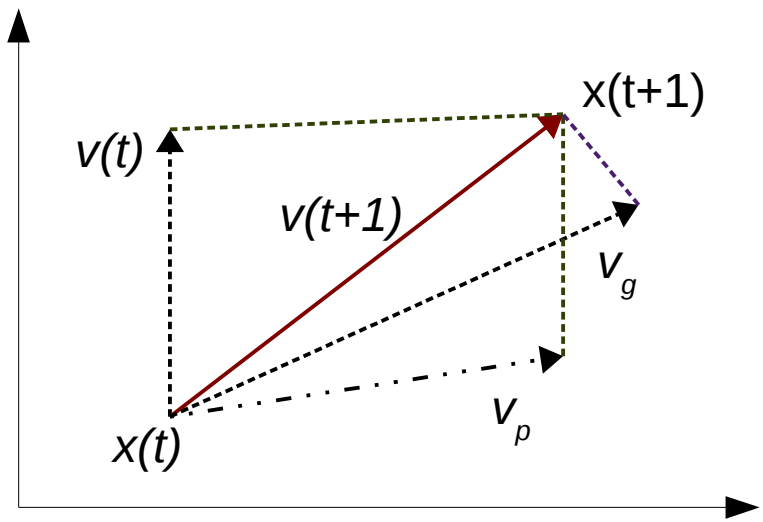
Personal influence

$$v_i(t+1) = \omega * v_i(t) + \psi_1 * u_1(p_i - x_i(t)) + \psi_2 * u_2(g_i - x_i(t))$$

Inertia

Social Influence

$$x_i(t+1) = x_i(t) + v_i(t+1)$$

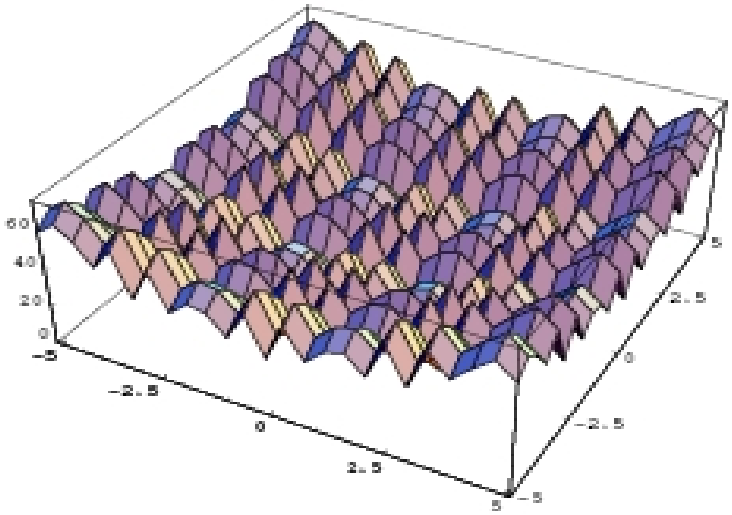


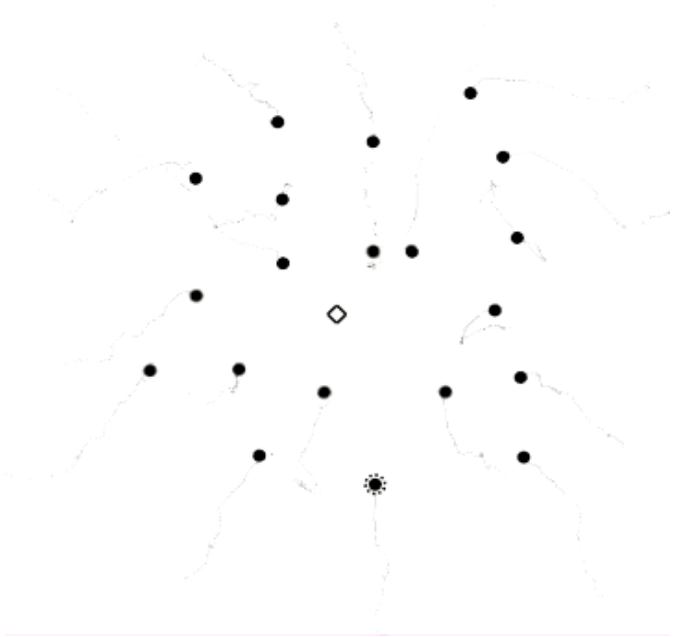
Particle Swarm Optimisation

```
Initialize particles
While(!termination)
  Update global best        #Topology dependent
  Update velocity
  Update current position
  Update personal best
End while
Return best solution
```

Particle Swarm Optimisation

$$F_{\text{Rastrigin}} = \sum_{i=1}^{m} ({x_i}^2 - 10\cos(2\pi x_i) + 10)$$

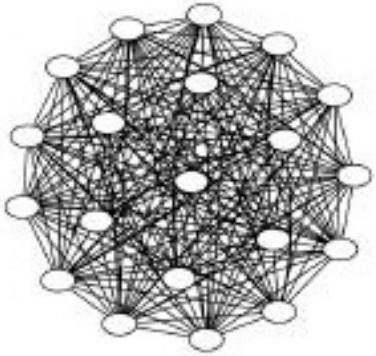




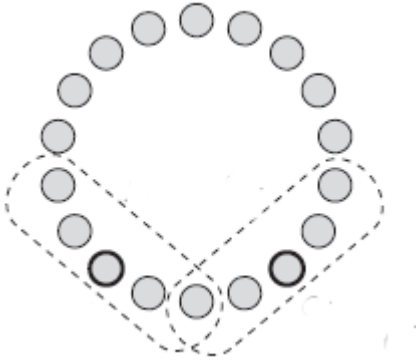
http://cg.kw.ac.kr/kang/pso/

Particle Swarm Optimisation: Topologies

- Define a neighbourhood for the particles.
  - Gbest: all particles are neighbours.



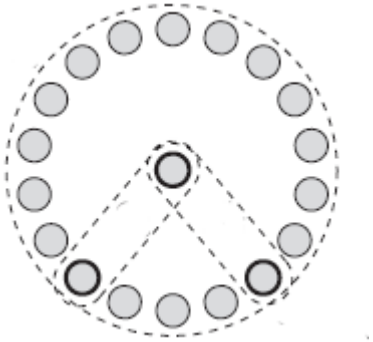
  - Ring: Each particle has n other neighbours.



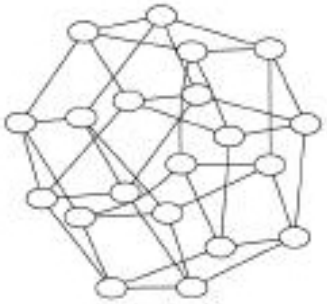
m=4

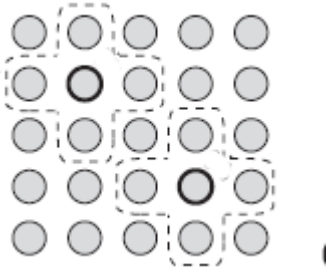
Particle Swarm Optimisation: Topologies

- Star: one central particle.



- Von Neumann: Network of 2 dimensions, each particle connected up and down, left and right.

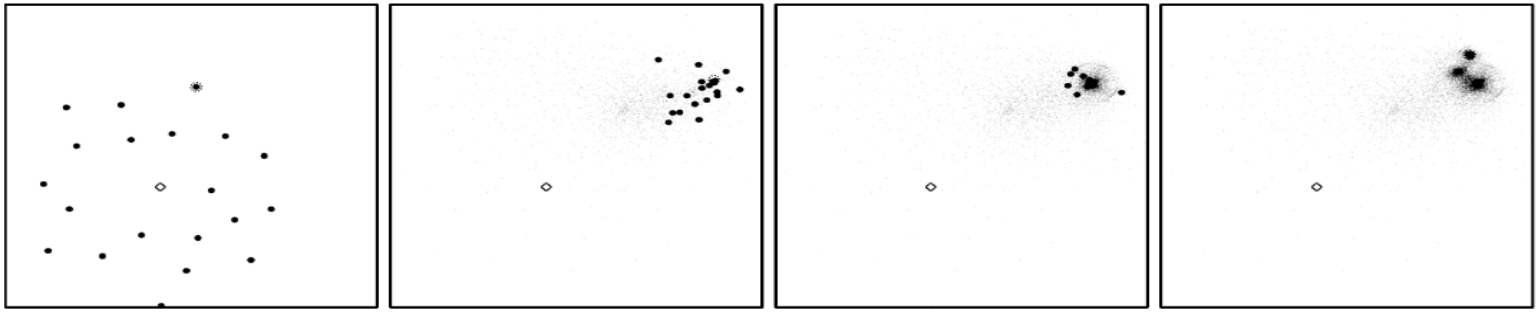




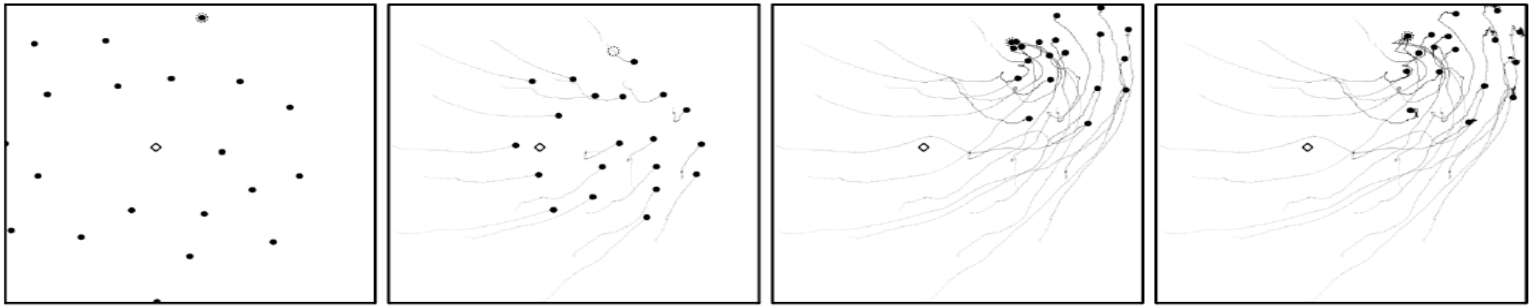
Particle Swarm Optimisation: Inertia

$$v_i(t+1)=\omega * v_i(t)+\psi_1 * u_1(p_i-x_i(t))+\psi_2 * u_2(g_i-x_i(t))$$

- Controlling the balance of the search
  - Inertia is small → more intensification



  - Inertia is big → more exploration



Particle Swarm Optimisation: Inertia

- Inertia can be varied during the search:
  - Example: Privilege exploration in initial iterations and exploitation at the end.

$$v_i(t+1)=\omega(t)*v_i(t)+\psi_1*u_1(p_i-x_i(t))+\psi_2*u_2(g_i-x_i(t))$$

  - Linear:

$$\omega(t)=(\omega(0)-\omega(T))\ \frac{T-t}{T}+\omega(T)$$

  - No lineal

$$\omega(t+1)=\alpha\,\omega(t')$$