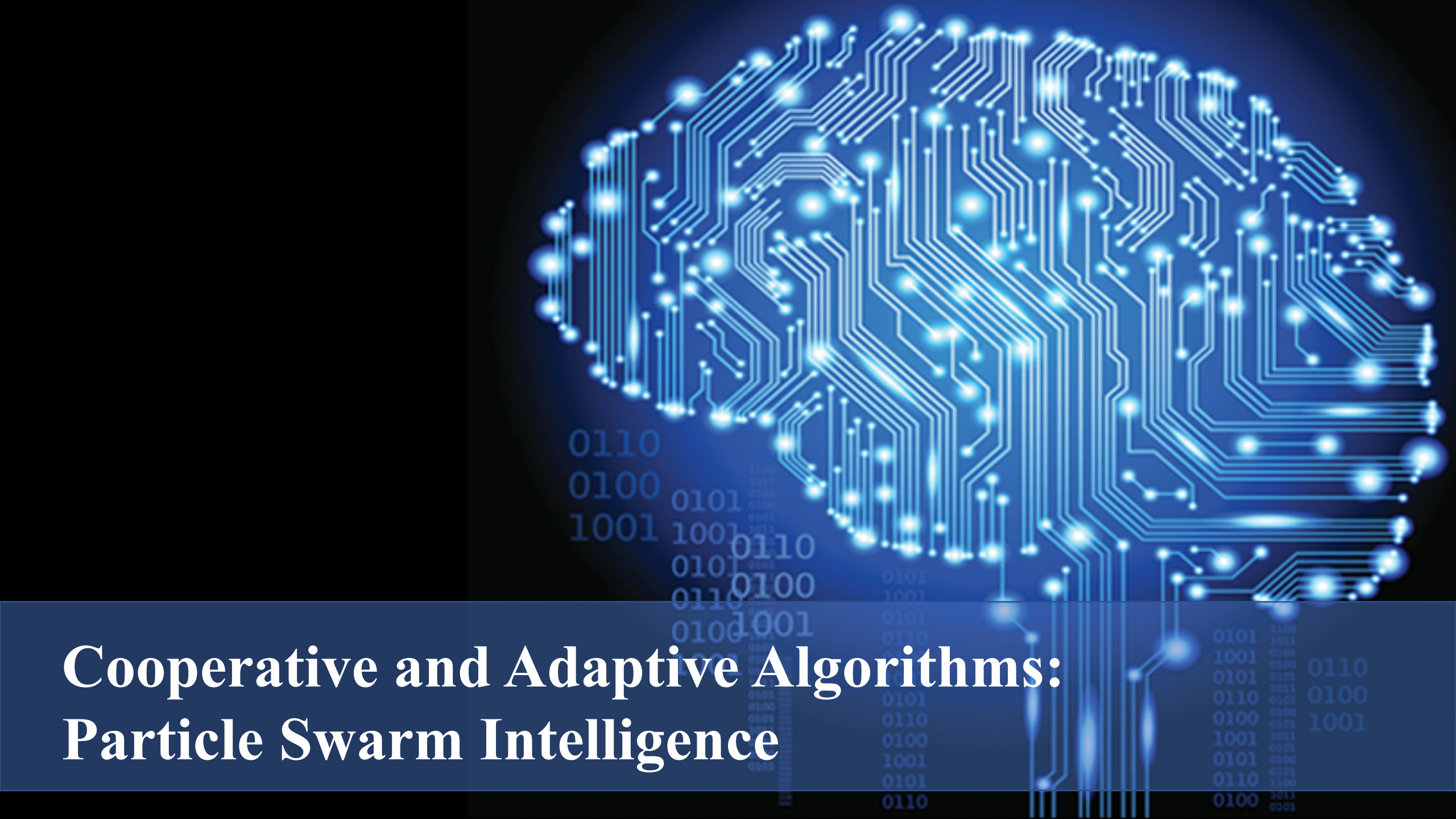


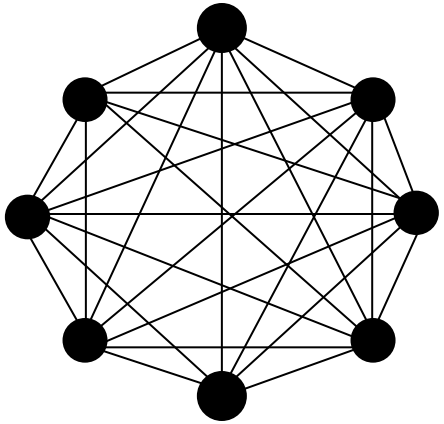
Cooperative and Adaptive Algorithms: Particle Swarm Intelligence

Particle Swarm Intelligence

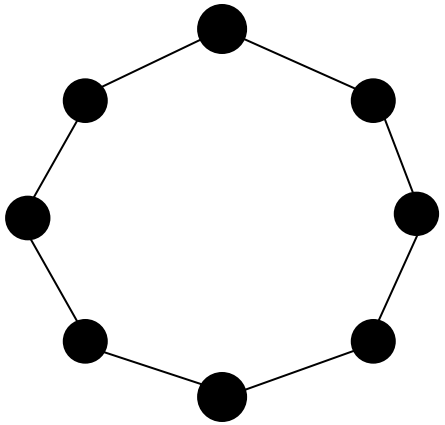
Neighborhood

PSO - Neighbourhoods

- In PSO, each particle shares its personal best information with other particles,
- Selecting a ***proper neighbourhood*** affects the ***convergence*** and also helps in avoiding getting ***stuck at local minima***,
- Different neighbourhood topologies have been introduced [9].



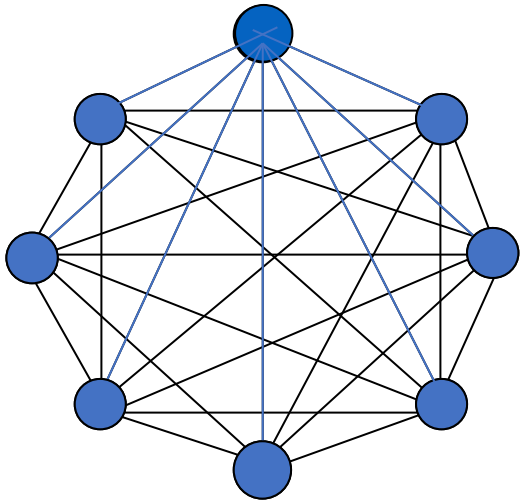
Star topology – global best



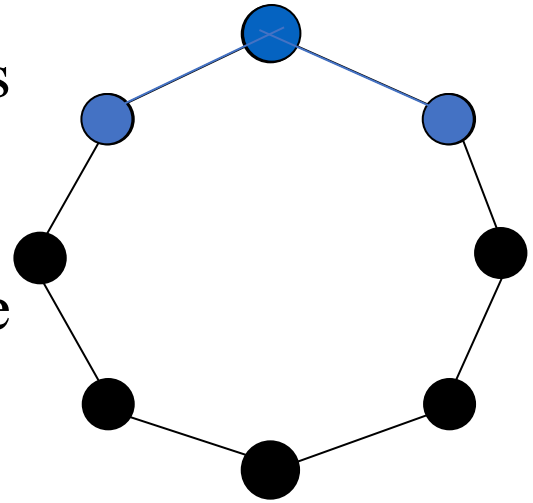
Ring topology – local best

PSO - Neighbourhoods

- *Star Topology:*
  - *gbest model:* each particle is influenced by all the other particles,
  - The fastest Propagation of information in a population,
  - Particles can get stuck easily in local minima.



- Ring Topology:
  - *lbest model:* each particle is influenced only by particles in its own neighbourhood,
  - The propagation of information is the slowest,
  - Doesn't get stuck easily in local minima but might increase the computational cost.



PSO - Neighbourhoods

- The most obvious way to select the neighbours for a certain particle *m* is to choose the particles that are closest to it in the search space (physical neighbors)

- The notion of closest is based on the ***distance in the Cartesian space***,
  - This approach might be computationally expensive as distances has to be computed each time the particle changes its position.

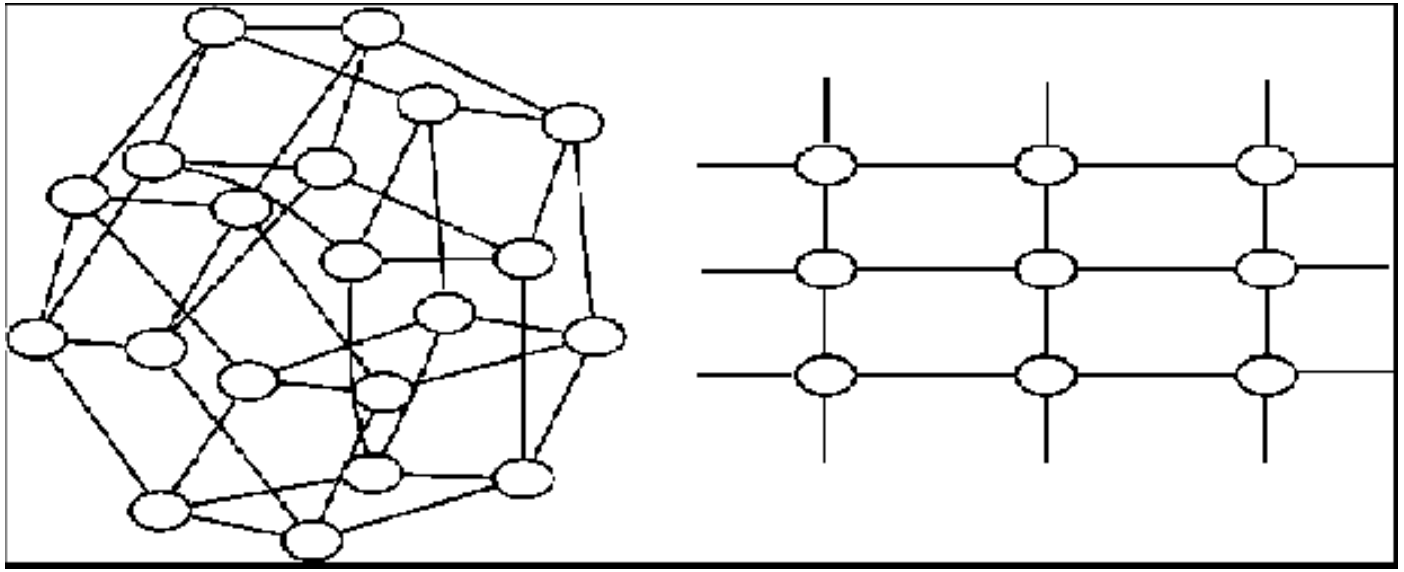
- The particles can be kept in a matrix data structure for example,
  - The most commonly used approach is to pick the particles that are stored next to *m* in the matrix (social neighbors)

- The performance is affected by the size of the neighbourhood selected (2, 4, 6, …).

PSO - Neighbourhoods

- The most successful neighbourhood structure was the square topology (Von Neumann model),

- Formed by arranging the particles in a grid and connecting the neighbours above , below and to the right and left.



The Von Neumann model [10]

The complete population

A local region

Particle Swarm Intelligence

Initialization

Initialization

- For each particle, the particle position and velocity need to be initialize.
- Particles positions can be initialized randomly in the range.
- Particles velocities can be initialized to zero or small values,
  - large values will result in large updates which may lead to divergence
- Personal best position is initialized to the particle's initial position

Initialization

- Parameters w, c_1, c_2, r_1, r_2

$$v_{t+1}^{id} = w * v_t^{id} + c_1 r_1^{id} \left(pbest_t^{id} - x_t^{id} \right) + c_2 r_2^{id} \left(Nbest_t^{id} - x_t^{id} \right)$$

- Using $c_1 = 0$ reduces the velocity model to Social-only model (particles are all attracted to *Nbest*)

- Using $c_2 = 0$ reduces to cognition-only model (particles are independent hill climbers)

Initialization

- In most applications $c_1 = c_2$
- Small values will result in smooth trajectories
- High values cause more acceleration with abrupt movement towards or past good regions
- The value of w is important to balance exploration and exploitation
  - Large values promote exploration and small promote exploitation (allowing more control to cognitive and social components)

Particle Swarm Intelligence

Convergence

PSO - Convergence

- An important question is: What are the suitable parameter values that guarantee the swarm convergence?

- The wrong settings can cause the particles to explode out of the search space,

- One of the simplest theoretical studies for PSO was proposed by Trelea in 2003 [7],
  - The study followed the same steps taken in dynamic systems theory,
  - It was carried using a deterministic PSO algorithm, randomness was removed.
  - The random numbers were replaced by their expected values

PSO - Convergence

- For a one-dimensional one-particle system:

$$r_1 = r_2 = \frac{1}{2}$$

$$v_{t+1} = w * v_t + \frac{c_1}{2}(pbest_t - x_t) + \frac{c_2}{2}(gbest_t - x_t)$$

$$x_{t+1} = x_t + v_{t+1}$$

- Let

$$c = \frac{c_1 + c_2}{2}$$

$$p = \frac{c_1}{c_1 + c_2} pbest + \frac{c_2}{c_1 + c_2} gbest$$

PSO - Convergence

- Hence, the equations of motion would be:

$$v_{t+1} = w * v_t + c(p - x_t)$$

$$x_{t+1} = x_t + v_{t+1}$$

- The equations could be rewritten in matrix form:

$$y_{t+1} = Ay_t + Bp$$

where

$$y_t = \begin{bmatrix} x_t \\ v_t \end{bmatrix}, A = \begin{bmatrix} 1-c & w \\ -c & w \end{bmatrix}, B = \begin{bmatrix} c \\ c \end{bmatrix}$$

PSO - Convergence

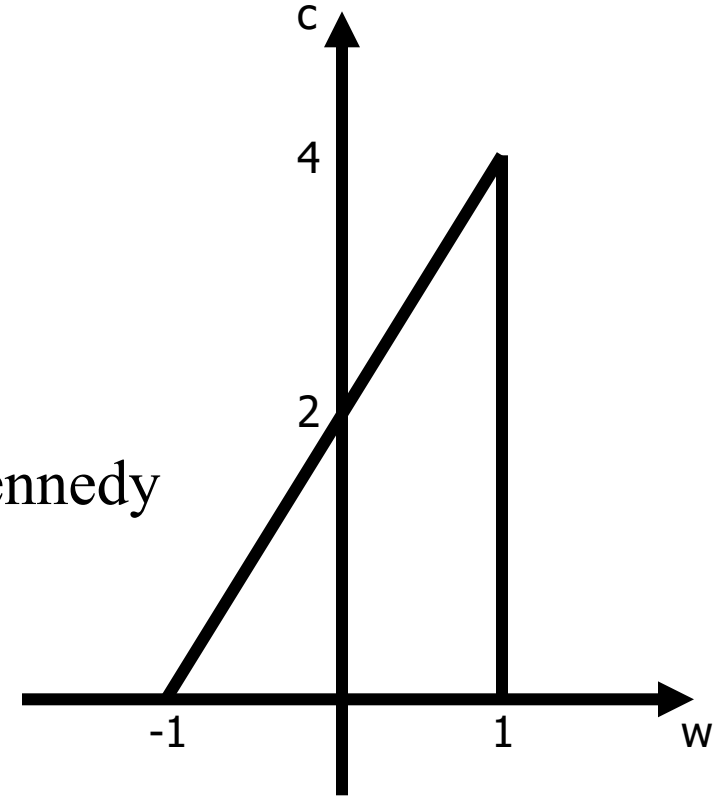
- By analyzing the characteristic equation, we can find the conditions necessary for stability:

$$w < 1,$$

$$c > 0,$$

$$2 * w - c + 2 > 0$$

- The convergence domain would be inside the triangle shown.



- The widely used set of parameters is proposed by Clerc and Kennedy in 2003 [8]:

$$w = 0.792,$$

$$c1 = c2 = 1.4944$$

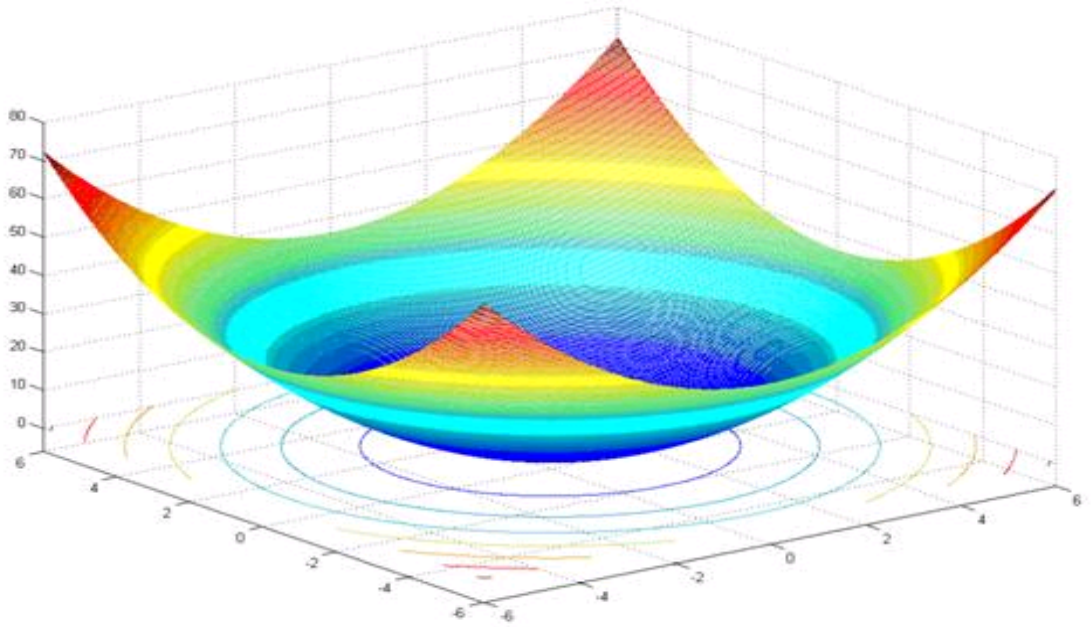
Particle Swarm Intelligence

PSO vs GA

PSO – A Comparison with GAs

- **Spherical Function:**

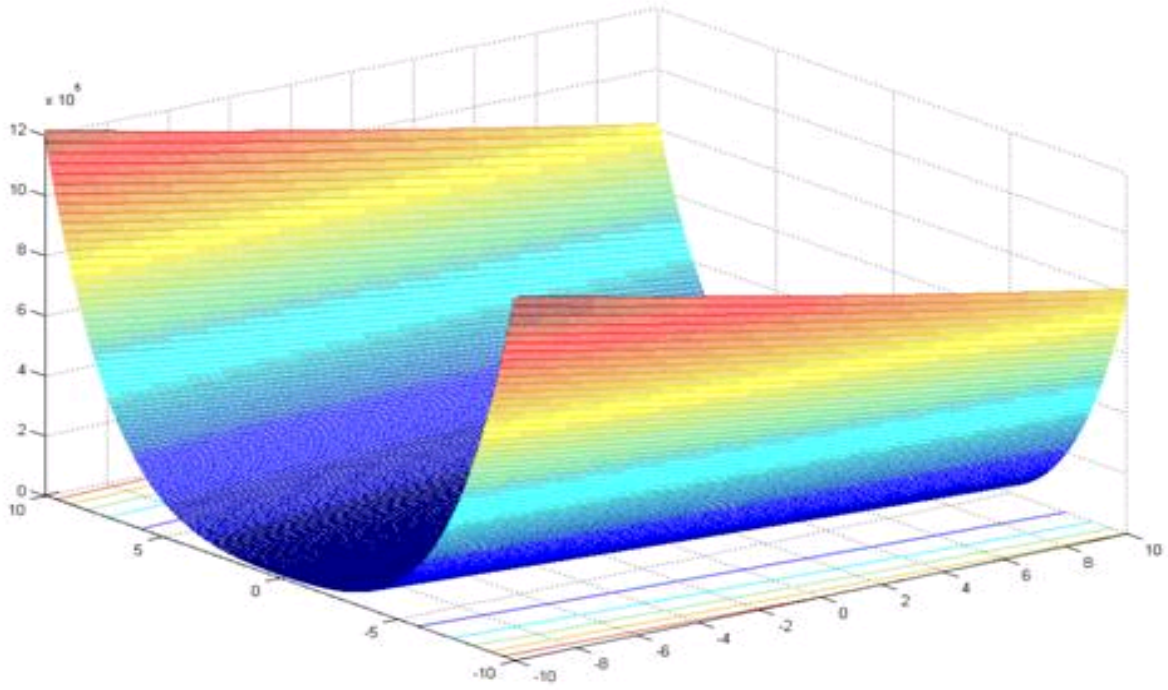
$$f(x) = \sum_{i=1}^{d} x_i^2$$



PSO – A Comparison with GAs

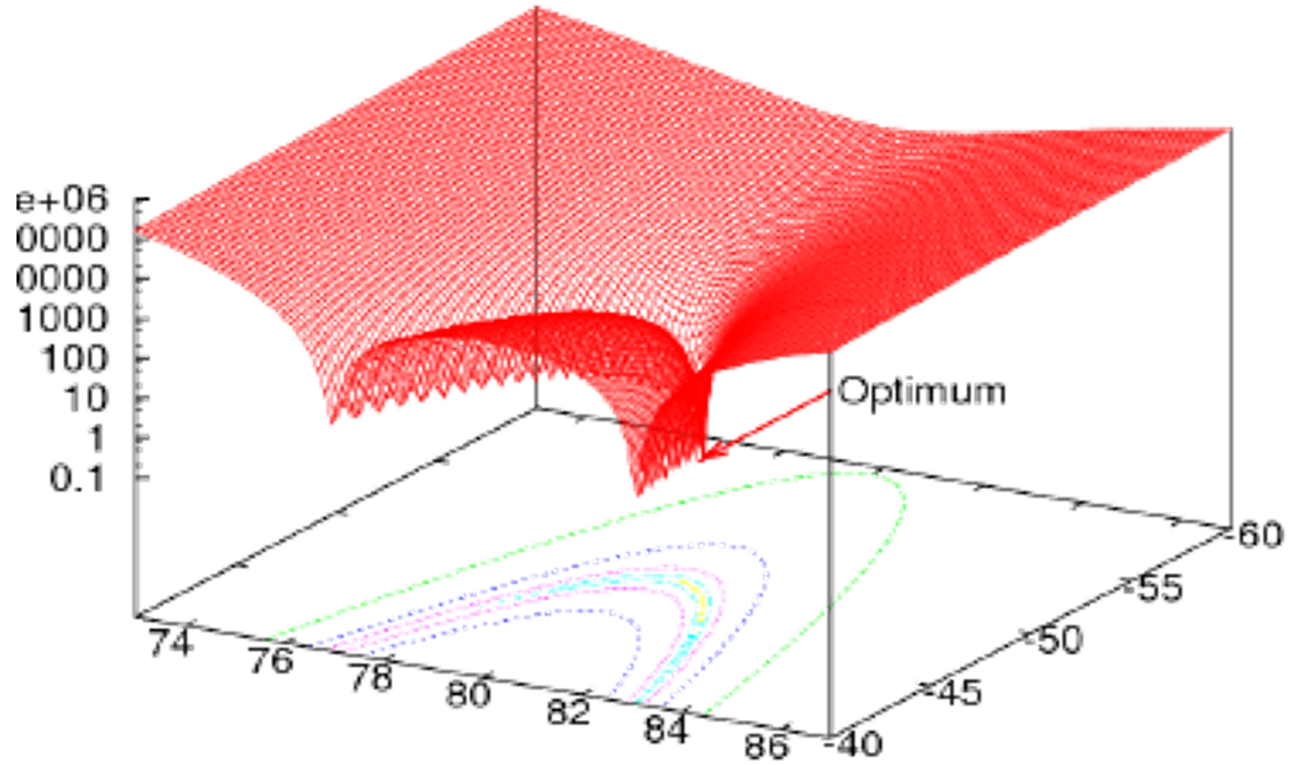
- **Rosenbrock Function:**

$$f(x) = \sum_{i=1}^{d-1} \left[(1 - x_i)^2 + 100 \left(x_{i+1} - x_i^2 \right)^2 \right]$$



PSO – A Comparison with GAs

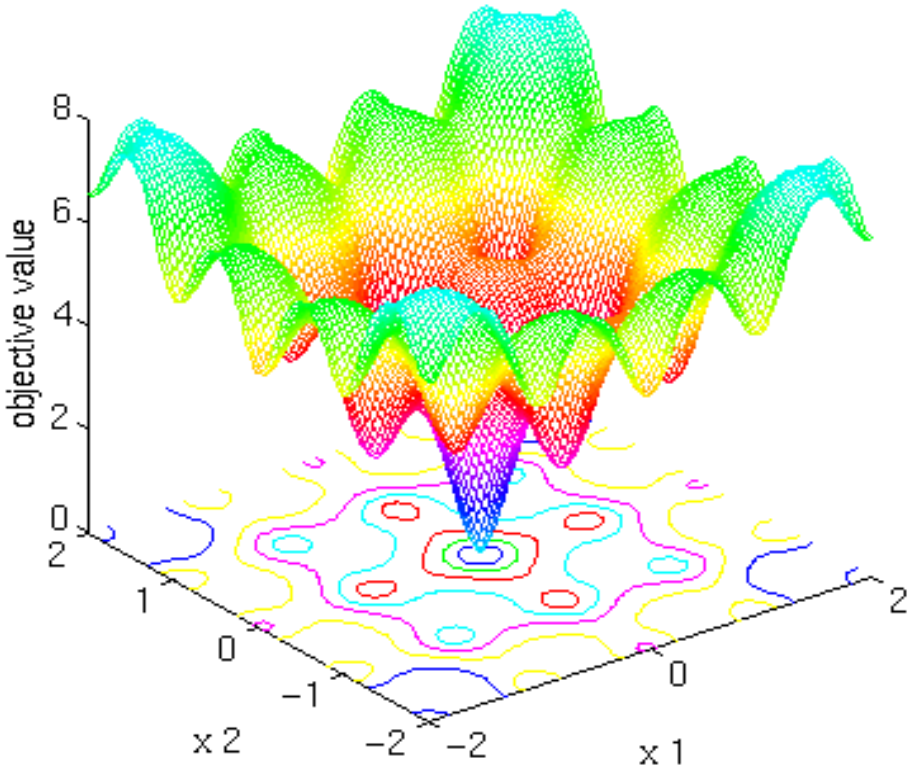
- **Rosenbrock Function:**



PSO – A Comparison with GAs

- **Ackley Function:**

$$f(x) = -20\exp\left(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d}x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{d}\cos(2\pi x_i)\right) + 20 + e$$

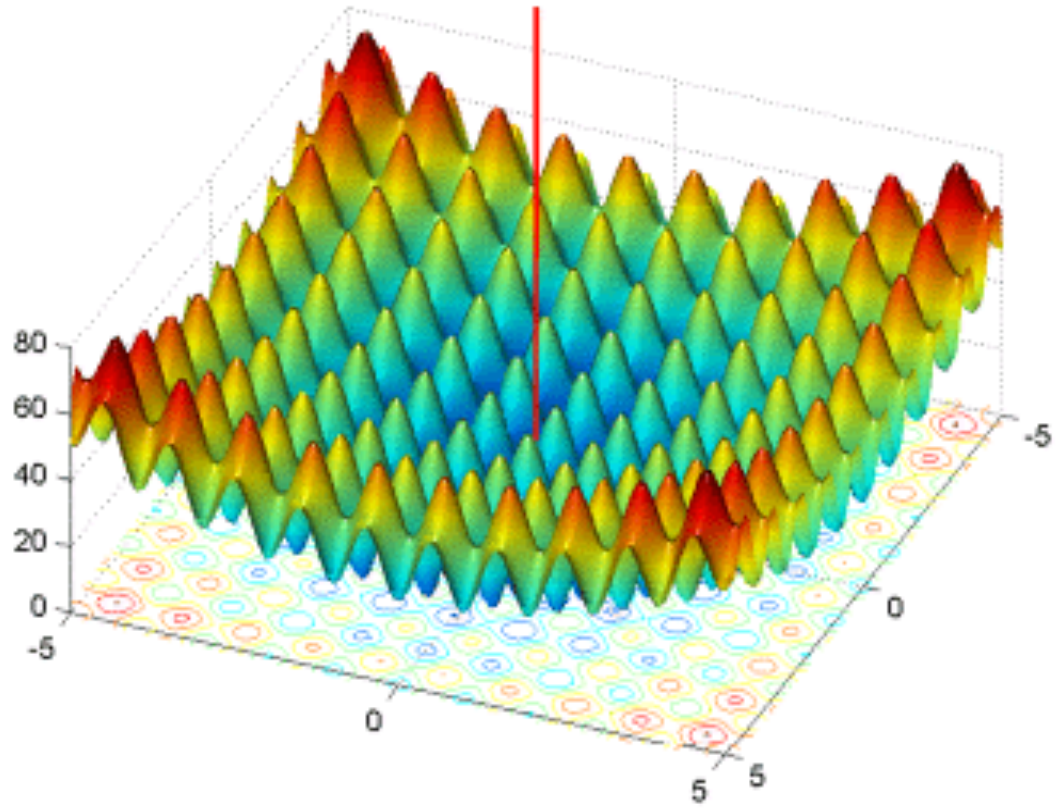


PSO – A Comparison with GAs

- **Rastrigin Function:**

$$f(x) = \sum_{i=1}^{d} x_i^2 - 10\cos(2\pi x_i) + 10$$

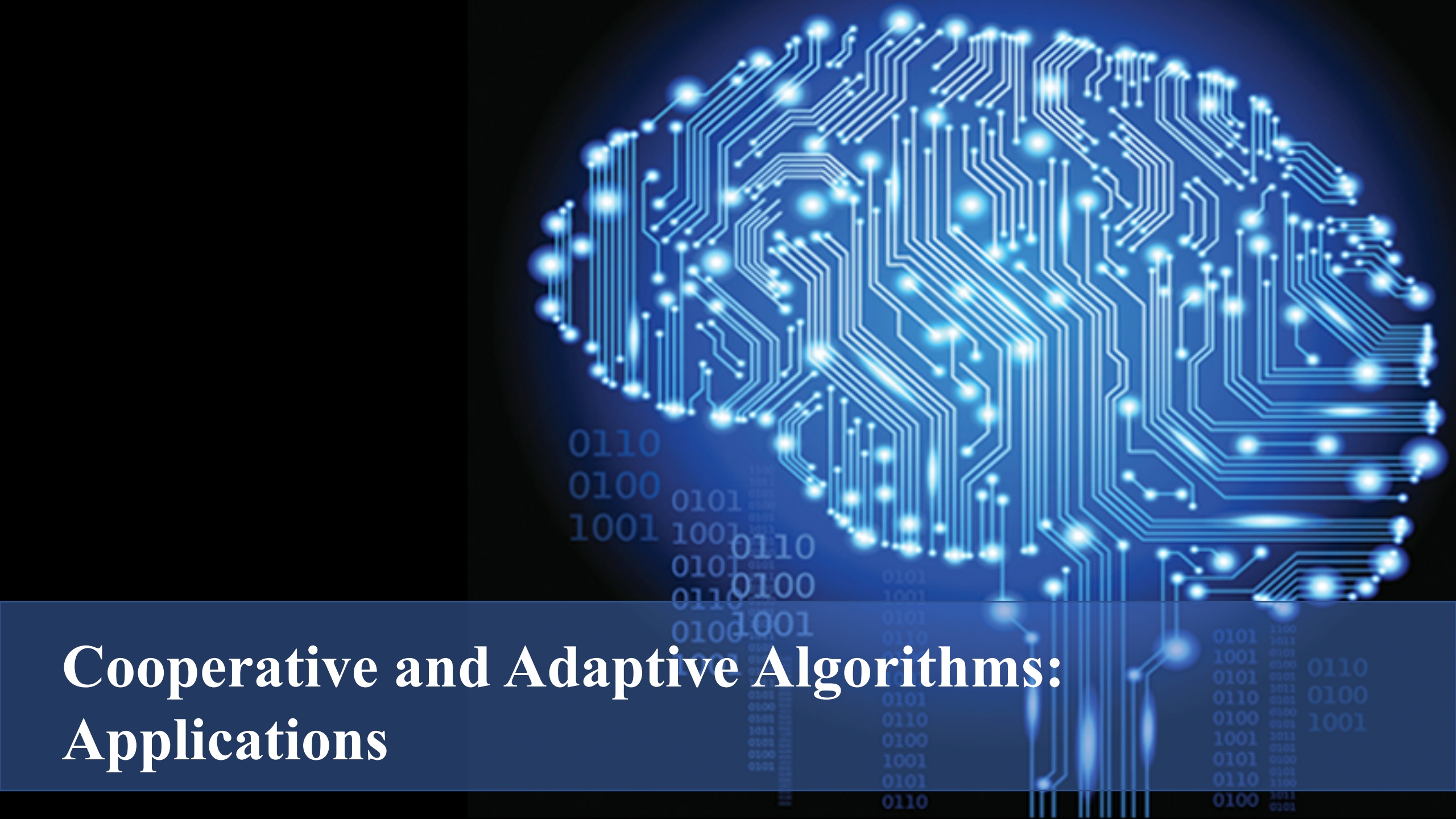
Global minimum at [0 0]



PSO - A comparison with GAs

- A comparison is made between PSO and GAs using the four functions (Spherical, Rosenbrock, Ackley and Rastrigin):
  - Both algorithms use 10 particles (individuals) and run for 1000 iterations, using Clerc and Kennedy parameters.
  - For a dimensionality of 10,
  - The results are the averages reported over 20 runs.

Benchmark	GA - Elitism	PSO - *gbest*
Spherical	0.0099	**2.3273e-17**
Rosenbrock	**5.5760**	9.6467
Ackley	0.9451	**0.3047**
Rastrigin	38.2186	**18.7730**



Cooperative and Adaptive Algorithms: Applications