

# Algoritmos de optimización bioinspirados

Introducción a Particle Swarm Optimization (PSO)

Inteligencia Artificial  
INFO257

Profesor: Jorge Maturana  
[jorge.maturana@inf.uach.cl](mailto:jorge.maturana@inf.uach.cl)

# PSO: Particle Swarm Optimization

- Se inspira del comportamiento de enjambres
- Un comportamiento complejo emerge de la interacción de varios agentes simples



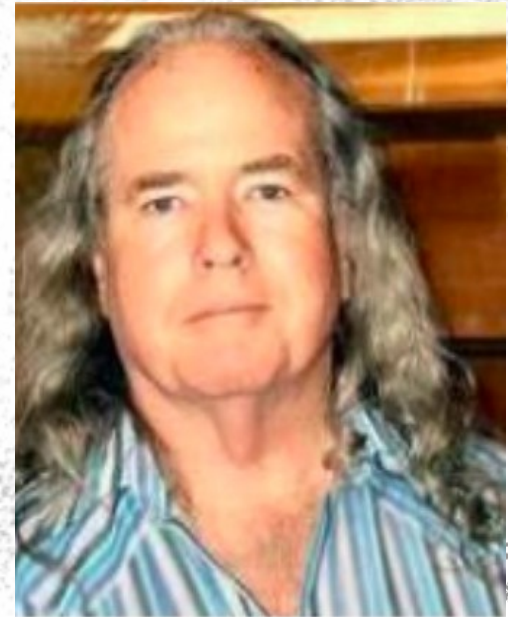
## Videos:

- [https://www.youtube.com/watch?v=V4f\\_1\\_r80RY](https://www.youtube.com/watch?v=V4f_1_r80RY)
- <https://www.youtube.com/watch?v=15B8qN9dre4>

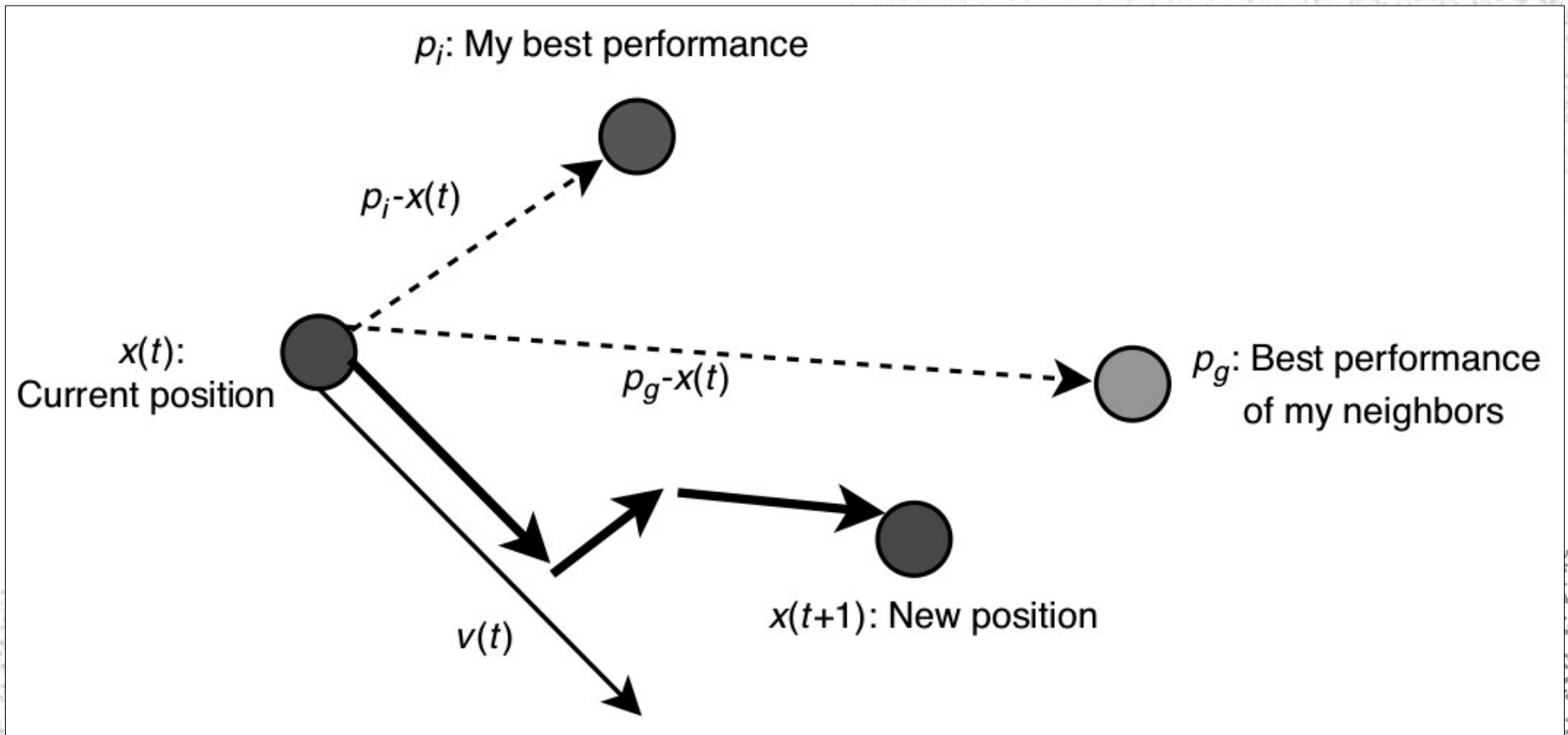


# PSO: Particle Swarm Optimization

- Propuesto por Kennedy & Eberhart en 1995
- Un conjunto de partículas se mueven intentando llegar a zonas atractivas del espacio de búsqueda
- El desplazamiento de la partícula depende de:
  - Su inercia (rapidez, dirección)
  - La mejor solución que ha encontrado
  - La mejor solución encontrada globalmente



# Movimiento de partículas



# Dos formas de actualizar posición

- La actualización de la velocidad puede plantearse de dos formas:
  - Parámetros regulan tradeoff individual/social

$$v_i(t) = v_i(t-1) + \rho_1 C_1 \times (p_i - x_i(t-1)) + \rho_2 C_2 \times (p_g - x_i(t-1))$$

- Parámetro regula inercia vs. Influencia de mejores valores encontrados

$$v_i(t) = w \times v_i(t-1) + \rho_1 \times (p_i - x_i(t-1)) + \rho_2 \times (p_g - x_i(t-1))$$

- Actualización de posición

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

# Algoritmo PSO

**Algorithm 3.14** Template of the particle swarm optimization algorithm.

Random initialization of the whole swarm ;

**Repeat**

Evaluate  $f(x_i)$  ;

**For all** particles  $i$

Update velocities:

$$v_i(t) = v_i(t - 1) + \rho_1 \times (p_i - x_i(t - 1)) + \rho_2 \times (p_g - x_i(t - 1)) ;$$

Move to the new position:  $x_i(t) = x_i(t - 1) + v_i(t)$  ;

**If**  $f(x_i) < f(pbest_i)$  **Then**  $pbest_i = x_i$  ;

**If**  $f(x_i) < f(gbest)$  **Then**  $gbest = x_i$  ;

Update( $x_i, v_i$ ) ;

**EndFor**

**Until** Stopping criteria