

# Maestría y Doctorado en Ciencia de la Computación

Inteligencia Artificial

Ant Colony System

Dr. Edward Hinojosa Cárdenas  
ehinojosa@unsa.edu.pe  
12 de Setiembre del 2020



UNSA

UNIVERSIDAD NACIONAL DE SAN AGUSTÍN DE AREQUIPA

# Índice



Objetivos del Curso

Computación Social

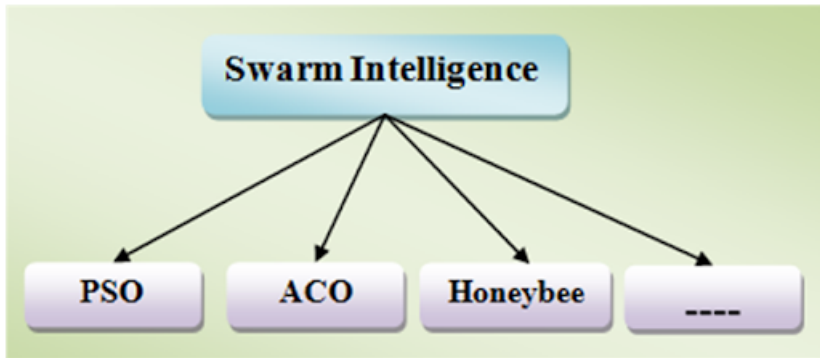
Particle Swarm Optimization

# Objetivos del Curso



- ▶ Conocer, comprender e implementar algoritmos evolutivos para resolver problemas complejos.
- ▶ **Conocer, comprender e implementar algoritmos de inteligencia de enjambre para resolver problemas complejos.**
- ▶ Conocer, comprender e implementar algoritmos inmunes artificiales para resolver problemas complejos.
- ▶ Conocer, comprender e implementar sistemas basados en lógica difusa para resolver problemas complejos.

# Computación Social





# Particle Swarm Optimization

- El Algoritmo de Optimización basado en Cúmulos de Partículas o Particle Swarm Optimization (PSO) fue propuesto por Kennedy (Psicólogo Social) y Eberhart (Ingeniero Eléctrico) en 1995 [2] y descrito en detalle en [1] y [3].



James Kennedy



Russel.C.Eberhart

# Particle Swarm Optimization



- Es una técnica metaheurística basada en poblaciones e inspirada en el comportamiento social del vuelo de las bandadas de aves.



# Particle Swarm Optimization



- ▶ Se basa en un enfoque conocido como “metáfora social” que describe a este algoritmo y que se puede resumir de la siguiente forma:
- ▶ “Los individuos que conviven en una sociedad tienen una opinión que es parte de un conjunto de creencias compartido por todos los posibles individuos”.

# Particle Swarm Optimization



- ▶ Cada individuo puede modificar su propia opinión basándose en tres factores:
  - ▶ Su conocimiento sobre el entorno (su valor de fitness).
  - ▶ Su conocimiento histórico o experiencias anteriores (su memoria).
  - ▶ El conocimiento histórico o experiencias anteriores de los individuos situados en su vecindario.

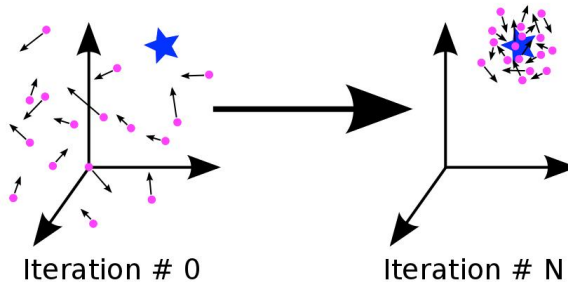




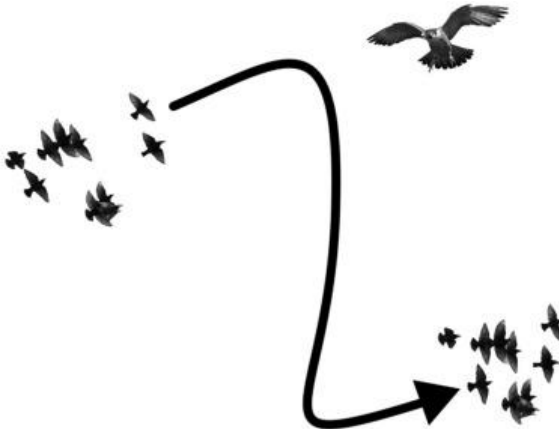


# Particle Swarm Optimization

- El algoritmo PSO explora el espacio de soluciones y encuentra soluciones de buena calidad. Eventualmente encuentra el óptimo del problema.



# Particle Swarm Optimization



# Particle Swarm Optimization



- Cada partícula (individuo) tiene una posición  $x$  (que en 2 dimensiones es determinada por un vector de la forma  $[x_1, x_2]$ ) en el espacio de búsqueda; y una velocidad  $v$  (que en 2 dimensiones es determinada por un vector de la forma  $[v_1, v_2]$ ) con la que se mueve a través del espacio.

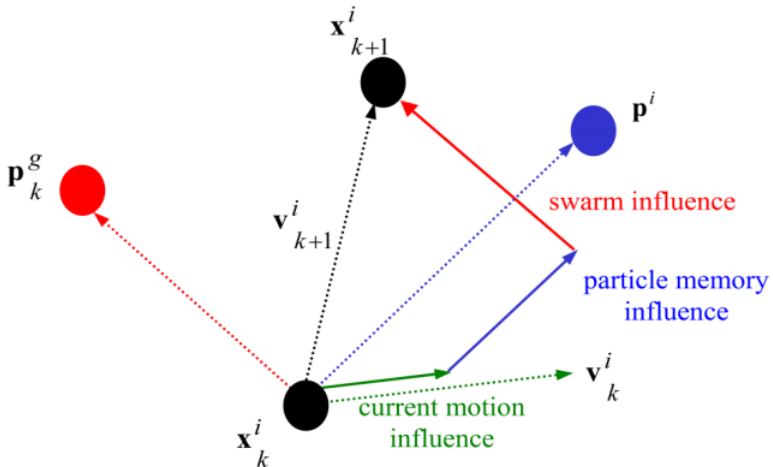


# Particle Swarm Optimization

- ▶ Además, cada partícula es influenciada por 3 factores:
  - ▶ La velocidad en que se movía (Inercia).
  - ▶ Atracción hacia la mejor localización que cada partícula (personal) ha encontrado en su historia (mejor personal).
  - ▶ Atracción hacia la mejor localización que ha sido encontrada por el conjunto de partículas (vecindario) en el espacio de búsqueda (mejor global).
    - ▶ El líder o mejor global tiene características o habilidades superiores. Los miembros del grupo confían en el líder.
    - ▶ El líder puede cambiar si un individuo presenta mejores características.

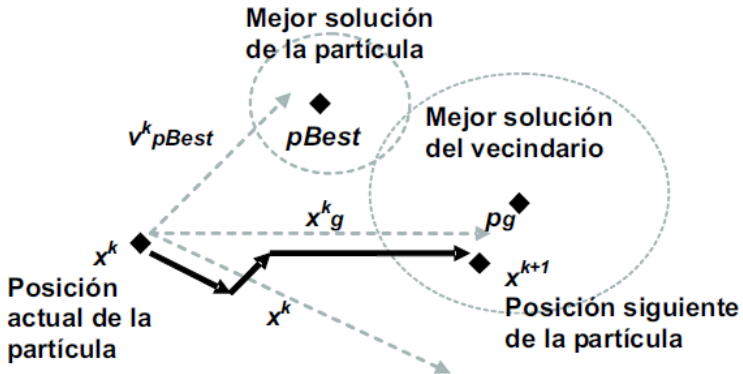


# Particle Swarm Optimization





# Particle Swarm Optimization





# Particle Swarm Optimization

## Algorithm: Canonical Particle Swarm Algorithm

```

for each particle i in the population do
    Initialise its location by randomly selecting values;
    Initialise its velocity vector to small random values close to zero;
    Calculate its fitness value;
    Set initial  $p_i^{\text{best}}$  to the particle's current location;
end
Determine the location of  $g^{\text{best}}$ ;
repeat
    for each particle i in turn do
        Calculate its velocity;
        Update its position;
        Measure fitness of new location;
        if fitness of new location is greater than that of  $p_i^{\text{best}}$  then
            | Revise the location of  $p_i^{\text{best}}$ ;
        end
    end
    Determine the location of the particle with the highest fitness;
    if fitness of this location is greater than that of  $g^{\text{best}}$  then
        | Revise the location of  $g^{\text{best}}$ ;
    end
until terminating condition;
  
```

# Particle Swarm Optimization



- ▶ Un algoritmo PSO consiste en un proceso iterativo y estocástico que opera sobre un cúmulo de partículas.
- ▶ La posición de cada partícula representa una solución potencial al problema que se está resolviendo.
- ▶ Generalmente, una partícula  $p_i$  está compuesta de tres vectores y dos valores de fitness:





# Particle Swarm Optimization

- ▶ El vector  $x_i = [x_{i1}, x_{i2}, \dots, x_{in},]$  almacena la posición actual (localización) de la partícula en el espacio de búsqueda.
- ▶ El vector  $pBest_i = [p_{i1}, p_{i2}, \dots, p_{in},]$  almacena la posición de la mejor solución encontrada por la partícula hasta el momento.
- ▶ El vector de velocidad  $v_i = [v_{i1}, v_{i2}, \dots, v_{in},]$  almacena el gradiente (dirección) según el cual se moverá la partícula.
- ▶ El valor de fitness  $fitness\_x_i$  almacena el valor de adecuación de la solución actual (vector  $x_i$ ).
- ▶ El valor de fitness  $fitness\_pBest_i$  almacena el valor de adecuación de la mejor solución local encontrada hasta el momento (vector  $pBest_i$ ).

# Particle Swarm Optimization



- ▶ El cúmulo se inicializa generando las posiciones y las velocidades iniciales de las partículas. Ambas se puede generar de forma aleatoria.
- ▶ Una vez generadas las posiciones, se calcula el fitness de cada una y se actualizan los valores de  $fitness_{x_i}$  y  $fitness_{pBest_i}$ .
- ▶ Se obtiene la mejor partícula  $g_{best}$



# Particle Swarm Optimization

- Inicializado el cúmulo, las partículas se deben mover dentro del proceso iterativo. Una partícula se mueve desde una posición del espacio de búsqueda hasta otra, simplemente, añadiendo al vector posición  $x_i$  el vector velocidad  $v_i$  para obtener un nuevo vector posición:

$$x_i \leftarrow x_i + v_i$$

*Solo se suma*



# Particle Swarm Optimization

- El vector velocidad de cada partícula es modificado en cada iteración utilizando la velocidad anterior, un componente cognitivo y un componente social. El modelo matemático resultante y que representa el corazón del algoritmo PSO viene representado por las siguientes ecuaciones:

$$v_i^{k+1} = \omega \cdot v_i^k + \varphi_1 \cdot rand_1 \cdot (pBest_i - x_i^k) + \varphi_2 \cdot rand_2 \cdot (g_i - x_i^k)$$
$$x_i^{k+1} = x_i^k + v_i^{k+1}$$



# Particle Swarm Optimization

20

- ▶  $v_i^k$ : velocidad de la partícula  $i$  en la iteración  $k$ ,
- ▶  $w$ : factor inercia.
- ▶  $\varphi_1, \varphi_2$ : son ratios de aprendizaje (pesos) que controlan los componentes cognitivo y social,
- ▶  $rand_1, rand_2$ : números aleatorios entre 0 y 1,
- ▶  $x_i^k$ : posición actual de la partícula  $i$  en la iteración  $k$ ,
- ▶  $pBest_i$ : mejor posición (solución) encontrada por la partícula  $i$  hasta el momento,
- ▶  $g_j$ : representa la posición de la mejor solución global encontrada por todas las partículas hasta la iteración actual.



# Ejemplo de Implementación del Algoritmo PSO

- Considerar los parámetros solo para la población inicial.

Minimizar	Problema	
	$f(x_1, x_2) = x_1^2 + x_2^2$	LimIx <sub>i</sub> LimSx <sub>i</sub>
	$-5 \leq x_1 \leq 5$	-5   5
	$-5 \leq x_2 \leq 5$	-5   5
	$x_1, x_2 \in \mathbb{R}$	

# Ejemplo de Implementación del Algoritmo PSO



- ▶ Considere los siguientes parámetros:
  - ▶ Tamaño de la Población: 6
  - ▶ Valores iniciales para  $v_i$  entre -1.0 y 1.0
  - ▶  $w$ : número aleatorio entre 0.0 y 1.0 para cada iteración
  - ▶  $rand_1, rand_2$ : número aleatorio entre 0.0 y 1.0 para cada nuevo individuo
  - ▶  $\varphi_1, \varphi_2$ : 2.0
  - ▶ Cantidad de Iteraciones: 100

# ¡GRACIAS!





# Bibliografía



- [1] A. P. Engelbrecht.  
*Fundamentals of Computational Swarm Intelligence.*  
John Wiley and Sons, Inc., Hoboken, NJ, USA, 2006.
- [2] J. Kennedy and R. Eberhart.  
Particle swarm optimization.  
In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 4, pages 1942–1948 vol.4, 1995.
- [3] J. Kennedy, R. Eberhart, and Y. Shi.  
*Swarm Intelligence.*  
Evolutionary Computation Series. Elsevier Science, 2001.