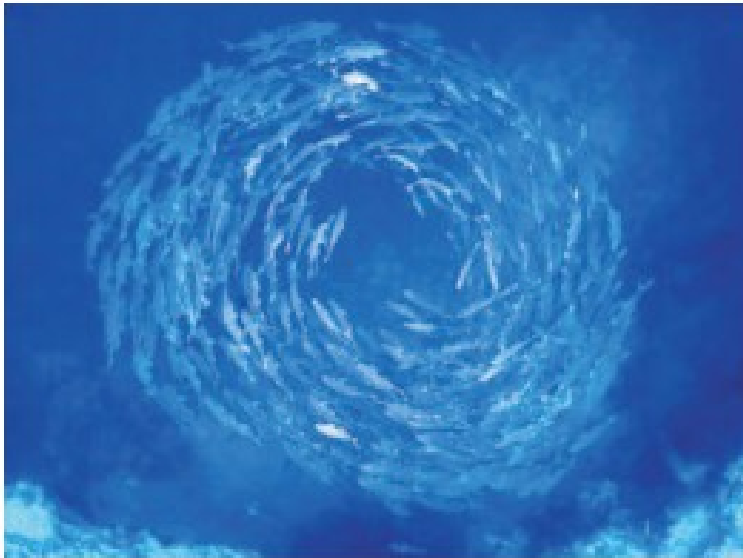


# Computación Bioinspirada

Dr. Edward Hinojosa Cárdenas  
[ehinojosa@unsa.edu.pe](mailto:ehinojosa@unsa.edu.pe)

# Algoritmo Basado en Cúmulos de Partículas (PSO)

- Un Algoritmo Basado en Cúmulos de Partículas o Particle Swarm Optimization es una técnica metaheurística basada en poblaciones e inspirada en el comportamiento social del vuelo de las bandadas de aves o el movimiento de los bancos de peces.



# Algoritmo Basado en Cúmulo de Partículas (PSO)

- Analiza las interrelaciones de los individuos con los integrantes de los grupos.
- PSO fue originalmente desarrollado por el psicólogo sociólogo James Kennedy y por el ingeniero electrónico Russell Eberhart en 1995.
- Se basaron principalmente en el trabajo de Reynolds y Heppner.

# Algoritmo Basado en Cúmulos de Partículas (PSO)

- Reynolds: Estética de las coreografías de las bandadas de aves.
- Heppner: Reglas de los vuelos sincronizados, y como se dispersan y reagrupan.

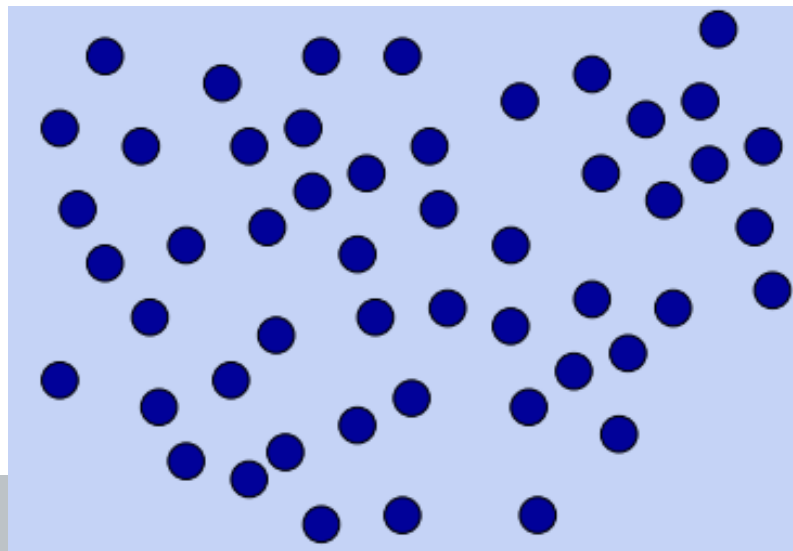


# Algoritmo Basado en Cúmulo de Partículas (PSO)

- PSO es un algoritmo metaheurístico basado en población.
- 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 (el espacio de búsqueda) compartido por todos los posibles individuos.

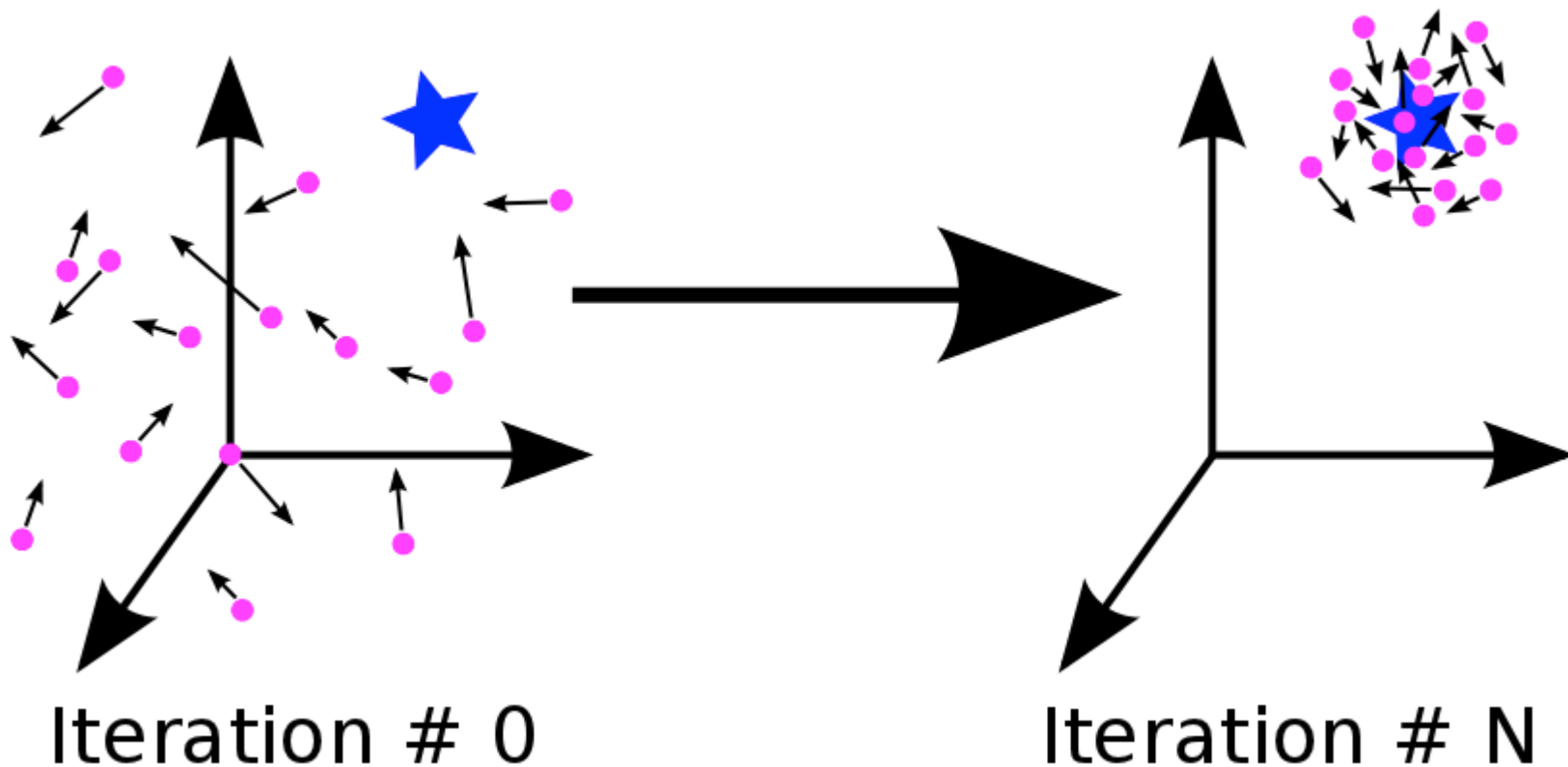
# Algoritmo Basado en Cúmulos de Partículas (PSO)

- 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.

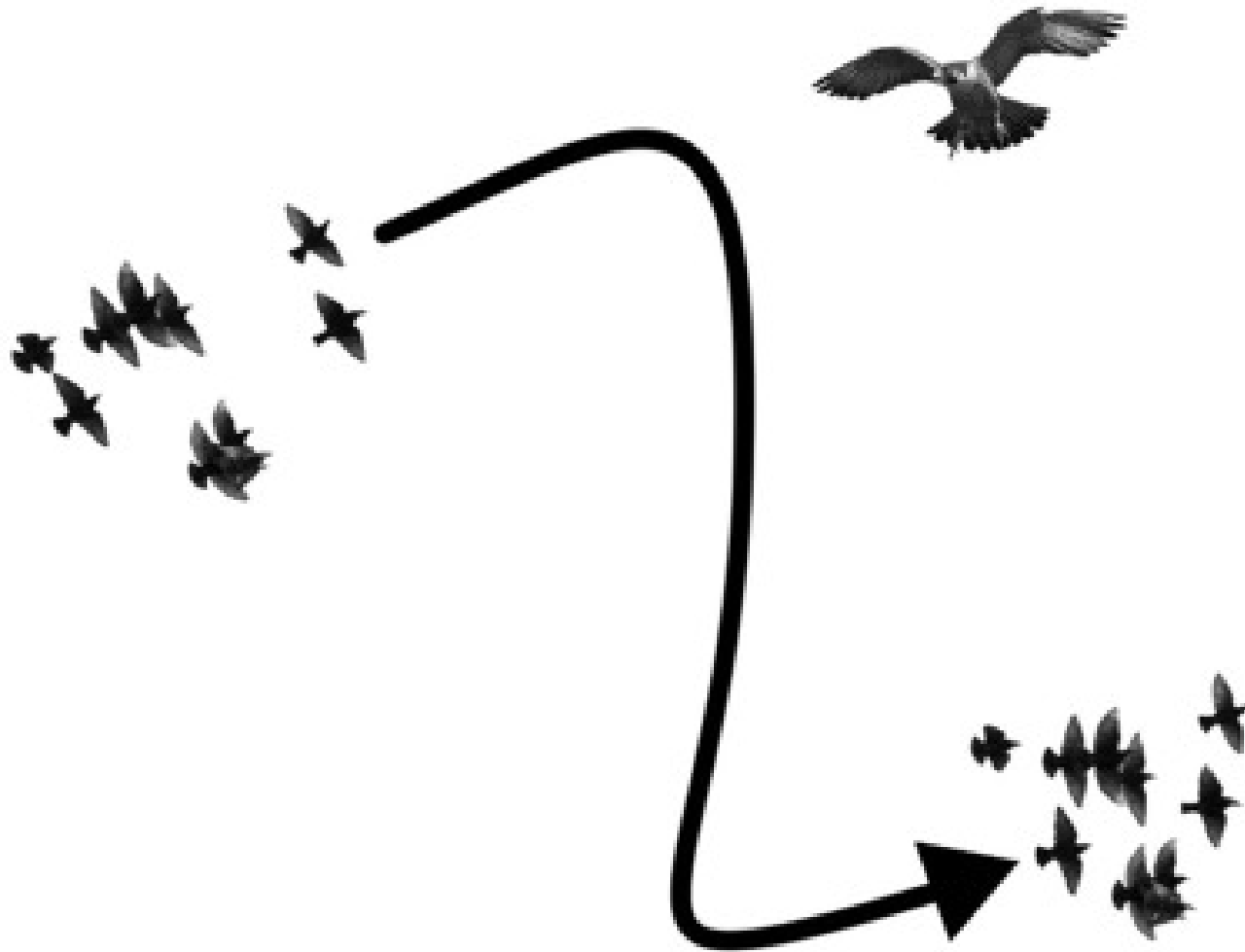


# Algoritmo Basado en Cúmulos de Partículas (PSO)

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



# Algoritmo Basado en Cúmulos de Partículas (PSO)





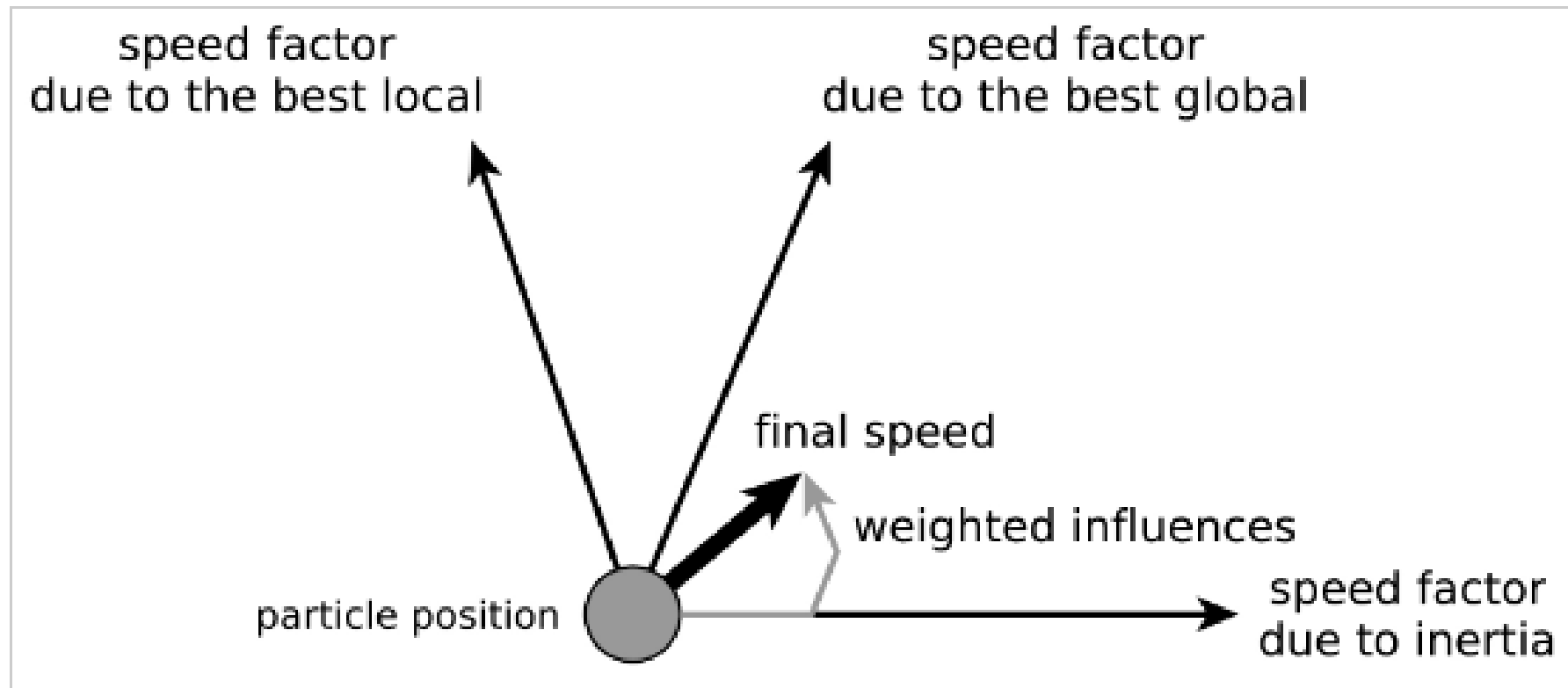
# Algoritmo Basado en Cúmulo de Partículas (PSO)

- Cada partícula (individuo) tiene una posición,  $x$ , (que en 2 dimensiones vendrá determinado por un vector de la forma  $(x_1, x_2)$ ), en el espacio de búsqueda y una velocidad,  $v$ , (que en 2 dimensiones vendrá determinado por un vector de la forma  $(v_x, v_y)$ ) con la que se mueve a través del espacio.
- Además, como partículas de un mundo real físico, tienen una cantidad de inercia, que los mantiene en la misma dirección en la que se movían, así como una aceleración (cambio de velocidad), que depende principalmente de dos características:

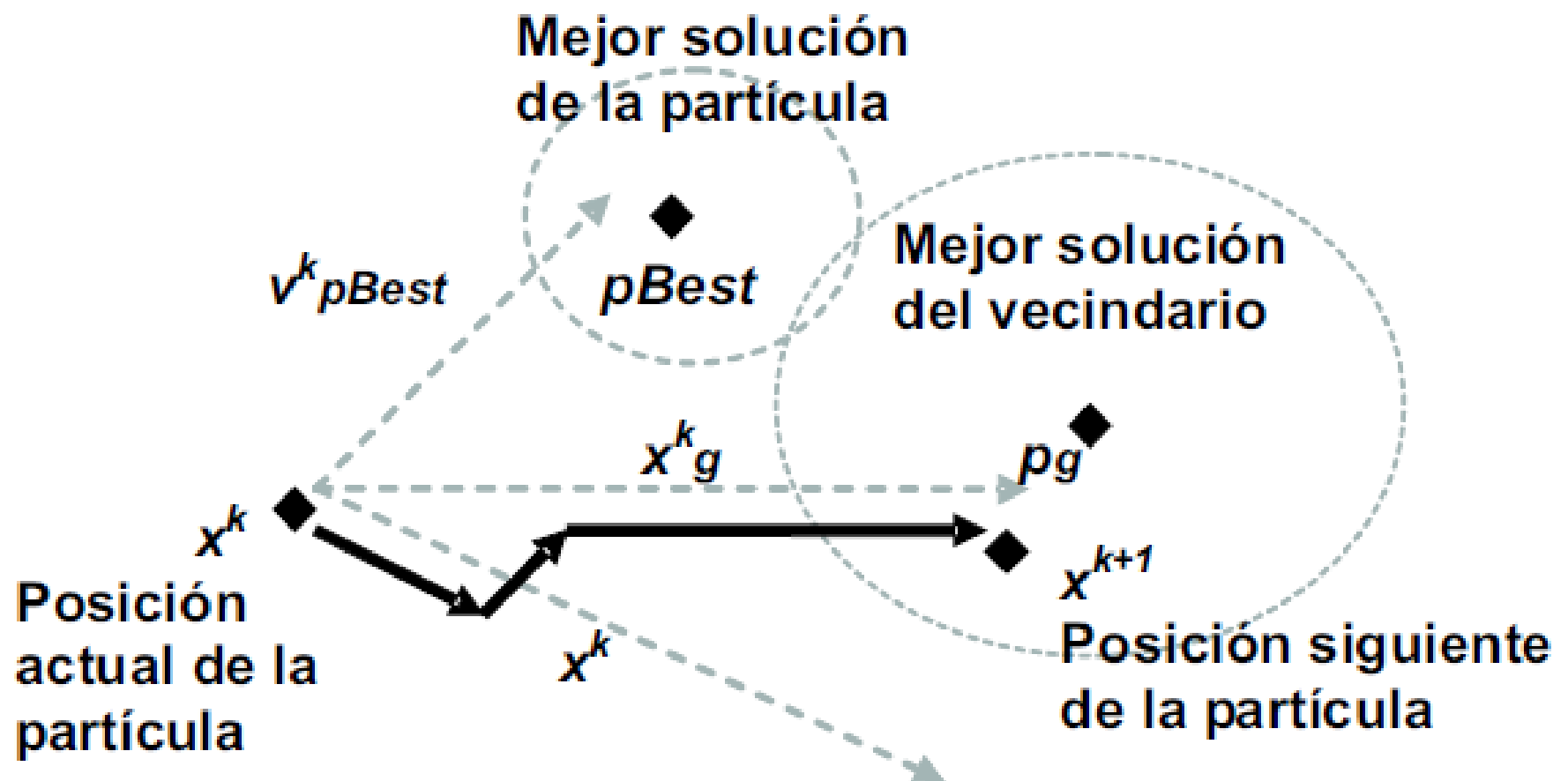
# Algoritmo Basado en Cúmulo de Partículas (PSO)

- Cada partícula es atraída hacia la mejor localización que ella, personalmente, ha encontrado en su historia (mejor personal).
- Cada partícula es atraída hacia la mejor localización que ha sido encontrada por el conjunto de partículas en el espacio de búsqueda (mejor global).

# Algoritmo Basado en Cúmulos de Partículas (PSO)



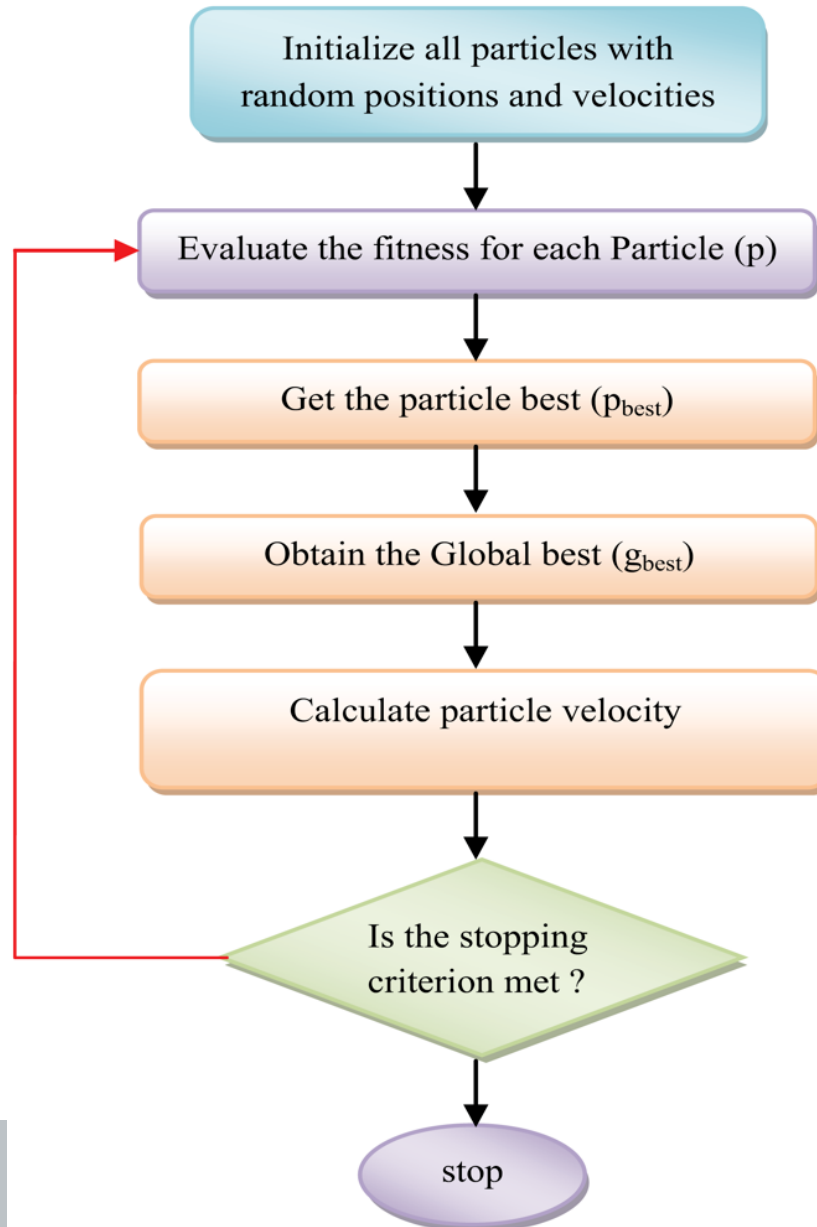
# Algoritmo Basado en Cúmulos de Partículas (PSO)



# Algoritmo Basado en Cúmulo de Partículas (PSO)

- 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.

# Flujograma del Algoritmo PSO



# Descripción del Algoritmo PSO

- 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:

# Descripción del Algoritmo PSO

- El vector  $x_i = \langle x_{i1}, x_{i2}, \dots, x_{in} \rangle$  almacena la posición actual (localización) de la partícula en el espacio de búsqueda.
- El vector  $pBest_i = \langle p_{i1}, p_{i2}, \dots, p_{in} \rangle$  almacena la posición de la mejor solución encontrada por la partícula hasta el momento.
- El vector de velocidad  $v_i = \langle v_{i1}, v_{i2}, \dots, v_{in} \rangle$  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  $pBest_i$ ).

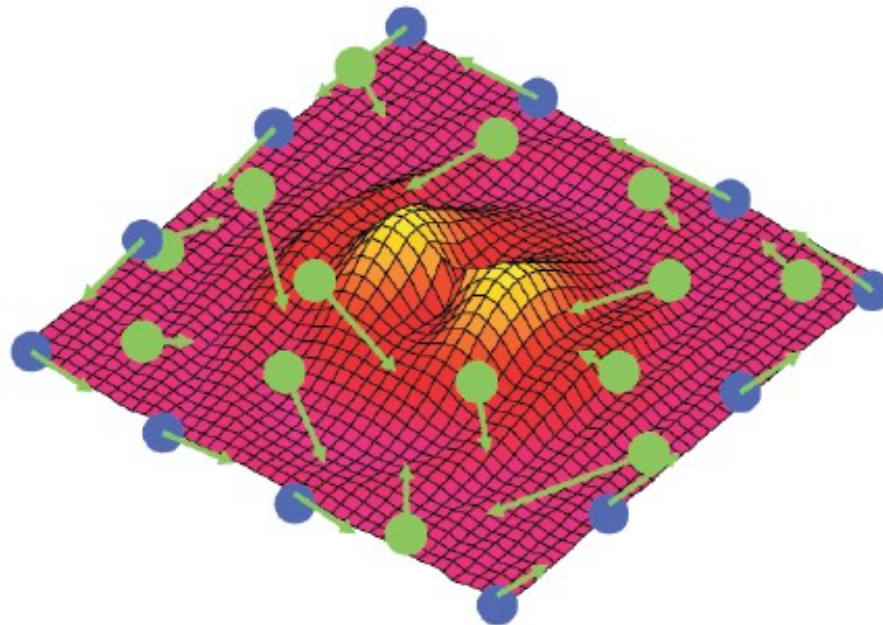


# Descripción del Algoritmo PSO

- El cúmulo se inicializa generando las posiciones y las velocidades iniciales de las partículas. Las posiciones se pueden generar aleatoriamente en el espacio de búsqueda (quizás con ayuda de un heurístico de construcción), de forma regular o con una combinación de ambas formas.
- Una vez generadas las posiciones, se calcula el fitness de cada una y se actualizan los valores de  $fitness_{x_i}$  y  $fitness_{pBest_i}$ .

# Descripción del Algoritmo PSO

- Las velocidades se generan aleatoriamente, con cada componente en el intervalo  $[-v_{\max}, v_{\max}]$ , donde  $v_{\max}$  será la velocidad máxima que pueda tomar una partícula en cada movimiento.



# Descripción del Algoritmo PSO

- 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:

$$x_i \leftarrow x_i + v_i$$

# Descripción del Algoritmo PSO

- Se obtiene la mejor partícula  $g_i$ .
- 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:

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

# Descripción del Algoritmo PSO

$v_i^k$  : velocidad de la partícula  $i$  en la iteración  $k$ ,

$\omega$  : 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,

$g_i$  : representa la posición de la partícula con el mejor  $pBest\_fitness$  del entorno de  $p_i$  ( $lBest$  o *localbest*) o de todo el cúmulo ( $gBest$  o *globalbest*).

# PSO: Ejemplo 01

- Minimizar la siguiente función:

Problema			
Minimizar	$f(x_1, x_2) = x_1^2 + x_2^2$	LimI $x_i$	LimS $x_i$
	$-5 \leq x_1 \leq 5$	-5	5
	$-5 \leq x_2 \leq 5$	-5	5
	$x_1, x_2 \in \mathbb{R}$		

# PSO: Ejemplo 01

- Tamaño de la población 6, cantidad de iteraciones 100, y:

$\omega$  : número aleatorio entre 0 y 1, (para cada iteración)

$\varphi_1, \varphi_2$  : 2.0

$rand_1, rand_2$  : números aleatorios entre 0 y 1,

# PSO: Ejemplo 01

\*\*\* Cúmulo de Partículas Inicial \*\*\*

```
1) x1= 0.3306116025717065; x2= -0.817303713515285; v1= 0.7582307407207431; v2= -0.9063717159683999
2) x1= 2.0481296845177566; x2= 1.046966045289314; v1= 0.2557331890526595; v2= 0.4044704951831317
3) x1= 0.3820900819661954; x2= 0.027260923093686884; v1= -0.9778569615395492; v2= 0.7268889929594033
4) x1= -2.7749147764355877; x2= 0.8949178069834653; v1= -0.11838948842945274; v2= -0.979946986388041
5) x1= -4.830591048339905; x2= 1.3088814065615848; v1= 0.6086045386929595; v2= -0.33292753953603027
6) x1= 3.5313002328092704; x2= 3.301309292568069; v1= 0.1306646985585973; v2= 0.18326798159659408
```

\*\*\*\* Fitness

```
1) 0.7772893918809071
2) 5.290973104591551
3) 0.14673598866485385
4) 8.501029897736663
5) 25.047780412744252
6) 23.368724379435093
```



# PSO: Ejemplo 01

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

$$x_i^{k+1} = x_i^k + v_i^{k+1}$$

\*\*\*\*\*Iteracion 1:

Mejor: x1: 0.3820900819661954; x2: 0.027260923093686884; fitness: 0.14673598866485385

Cúmulo de Partículas Siguiente

- 1) x1= 0.20793603267789784; x2= 0.2818912852080646; v1= -0.9451311272017631; v2= 0.9518835150270968
- 2) x1= -1.2000963539238447; x2= -0.7589928670917205; v1= -1.6736619728342277; v2= -1.090189838913415
- 3) x1= -0.1375050394048366; x2= 0.41350143614926926; v1= 0.4582618401685171; v2= -0.34064847990382097
- 4) x1= 0.5801072800042713; x2= 0.2556907937992867; v1= -2.3121737681284165; v2= 1.9308039766524274
- 5) x1= 2.5910809732073505; x2= -0.5882956756635536; v1= -2.90676109577521; v2= 0.8255247578610438
- 6) x1= 0.02708095810244915; x2= -0.2564169834794632; v1= -1.8890976766401728; v2= -1.9260030363099667

\*\*\*\* Fitness

- 1) 0.12270009036207823
- 2) 2.016301430997416
- 3) 0.18989107355923387
- 4) 0.4019022383476633
- 5) 7.059792411721588
- 6) 0.0664830477084539

# PSO: Ejemplo 01

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

\*\*\*\*\*Iteracion 2:

Mejor: x1: 0.02708095810244915; x2: -0.2564169834794632; fitness: 0.0664830477084539

Cúmulo de Partículas Siguiente

```
1) x1= -1.0651338133822787; x2= 0.1823770347705671; v1= -1.2730698460601766; v2= -0.09951425043749751
2) x1= 1.094188249315002; x2= 0.348308748217798; v1= 2.2942846032388466; v2= 1.1073016153095185
3) x1= 0.7522405029418796; x2= -0.5347155122671323; v1= 0.8897455423467162; v2= -0.9482169484164016
4) x1= -2.060264269289781; x2= 1.8011424289941156; v1= -2.640371549294052; v2= 1.5454516351948289
5) x1= -1.9368664772214328; x2= 0.43808638942129563; v1= -4.527947450428783; v2= 1.0263820650848492
6) x1= -1.8242347650049202; x2= -2.143899959063231; v1= -1.8513157231073694; v2= -1.8874829755837674
```

\*\*\*\* Fitness

```
1) 1.1677714232219796
2) 1.3185669090240784
3) 0.8517864533253536
4) 7.488802908834978
5) 3.9433714351803504
6) 7.92413951232388
```

# PSO: Ejemplo 01

\*\*\*\*\*Iteracion 99:

Mejor: x1: 5.92768349562405E-6; x2: -5.929596455615998E-6; fitness: 7.029754575074756E-11

Cúmulo de Partículas Siguiente

- 1) x1= 6.164595757283193E-6; x2= -6.028070746356717E-6; v1= -1.820778605324262E-8; v2= 7.104955916311254E-9
- 2) x1= 5.921696446685173E-6; x2= -5.92725277422093E-6; v1= -5.98704893887659E-9; v2= 2.3436813950682817E-9
- 3) x1= -7.972492142709628E-4; x2= 4.2896806726283956E-4; v1= -0.0013869372488892946; v2= 7.594544450405514E-4
- 4) x1= 6.030602569311466E-6; x2= -5.969798692979776E-6; v1= -9.632272205992289E-8; v2= 3.763655494036004E-8
- 5) x1= 5.969600799102485E-6; x2= -5.945994903131636E-6; v1= -2.3996547335328467E-7; v2= 9.388257930723375E-8
- 6) x1= 6.163152646676611E-6; x2= -6.025390599066216E-6; v1= -5.382168348502125E-9; v2= 1.932144585732729E-9

\*\*\*\* Fitness

- 1) 7.433987777379556E-11
- 2) 7.019881425619351E-11
- 3) 8.196199123868837E-7
- 4) 7.20066637836891E-11
- 5) 7.099098908871242E-11
- 6) 7.428978241755245E-11

Mejor después de 99 Iteraciones

x1: 5.92768349562405E-6; x2: -5.929596455615998E-6; fitness: 7.029754575074756E-11

# Práctica (0 a 20)

- Aplicar el algoritmo PSO para optimizar la función del ejemplo.
- Mostrar los valores vistos en el ejemplo.
- Utilizar los parámetros del ejemplo.

# GRACIAS

Dr. Edward Hinojosa Cárdenas  
[ehinojosa@unsa.edu.pe](mailto:ehinojosa@unsa.edu.pe)