



SWARM-BASED ALGORITHMS





ALGORITMO OPTIMIZACIÓN POR ENJAMBRE DE PARTÍCULAS Particle Swarm Optimization (PSO)

INTRODUCCIÓN

- Autores: *Kennedy y Eberhart* (1995).
- Objetivo inicial: optimizar funciones no lineales.
- Inspiración: comportamiento de enjambres de insectos o pájaros.

INSPIRACIÓN NATURAL

- Intenta simular comportamientos sociales de animales.
- Animales como los insectos, pájaros y peces se mueven en grupo, siguiendo a ciertos individuos del grupo, adaptándose a la velocidad del grupo, pero teniendo en cuenta también su propio movimiento previo.



TÉCNICAS DE BÚSQUEDA DE LA SOLUCIÓN

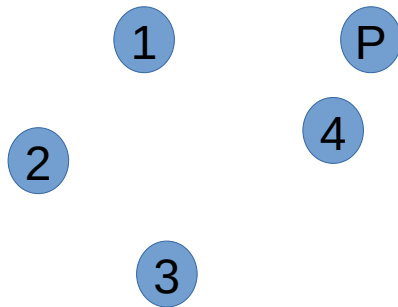
- Combina **exploración** y **explotación**.
 - *exploración* de nuevas soluciones.
 - *explotación* de la información ya obtenida.
- Aplica **búsqueda local** y **búsqueda global** para intentar establecer un equilibrio entre la exploración y la explotación.

EL PROBLEMA A RESOLVER

PROBLEMA DE OPTIMIZACIÓN (como se describe en la introducción).

LOS ELEMENTOS DE LA SOLUCIÓN

- Se considera un conjunto de P individuos denominados **partículas**. Cada partícula tiene asociado un estado que se modifica con el tiempo.



Variables asociadas a cada partícula:

- **posición**: representa su estado. Es una solución al problema.
- **velocidad**: determina la actualización de la posición.
- **mejor posición personal**: mejor solución encontrada por la partícula durante la ejecución del algoritmo.

partícula i : $\left\{ \begin{array}{l} \text{posición: } x_i(t) = (x_{i1}, \dots, x_{ir}) \\ \text{velocidad: } v_i(t) = (v_{i1}, \dots, v_{ir}) \\ \text{mejor posición personal: } b_i(t) \end{array} \right.$

LOS ELEMENTOS DE LA SOLUCIÓN

- La solución al problema viene dada por la mejor posición encontrada por el enjambre.

$g(t) \rightarrow$ mejor posición global

Las partículas se mueven en el espacio de solución del problema guiadas por su experiencia personal y por la experiencia de todo el enjambre

Ese movimiento hace variar su *posición*, *velocidad* y *mejor posición* personal.

ALGORITMO PSO

Inicializar la población de partículas

REPETIR

Evaluar el fitness de cada partícula

Actualizar la mejor solución personal de cada partícula

Actualizar la mejor solución global

Actualizar la velocidad y posición de cada partícula

HASTA (*condición de parada*)

ALGORITMO PSO

A- Inicializar la población de partículas

Inicializar posición y velocidad de todas ellas.

Posición inicial:
valores aleatorios del
espacio de búsqueda

Velocidad inicial:
valores aleatorios en
 $[v_{\min}, v_{\max}]$

ALGORITMO PSO

B- Evaluar el fitness de cada partícula

Para calcularlo se aplica la función objetivo del problema a la posición de la partícula:

$$\text{fitness}_i \approx f(x_i)$$

(se puede tomar directamente este resultado o usarlo para calcular un valor final).

El fitness asociado a la posición de la partícula determina la calidad de la solución que dicha posición representa.

ALGORITMO PSO

C- Actualizar la mejor solución personal de cada partícula

Si la nueva posición de una partícula es mejor que la mejor personal almacenada para ella hasta el momento, actualizar esta última con el valor de la primera.

Dar valor a $b_i(t+1)$ para $i=1,\dots,P$.

ALGORITMO PSO

D- Actualizar la mejor solución global

Si se ha encontrado una solución que mejora a la considerada mejor hasta el momento, guardarla como nueva mejor solución global.

Dar valor a $g(t+1)$

ALGORITMO PSO

E- Actualizar velocidad y posición

En primer lugar se actualiza la velocidad y después se actualiza la posición.

ALGORITMO PSO

E-1- Actualizar velocidad

Se calcula la nueva velocidad de cada partícula i:

$$v_i(t+1) = w v_i(t) + f_1 e_1 [b_i(t) - x_i(t)] + f_2 e_2 [g(t) - x_i(t)] \quad (1)$$

Parámetros:

e_1, e_2 : valores aleatorios en $[0, 1]$

w : inercia

f_1 : parámetro cognitivo

f_2 : parámetro social

La actualización incluye 3 componentes ponderadas por los parámetros:

ALGORITMO PSO

E-1- Actualizar velocidad

1º- velocidad anterior:

$$v_i(t+1) = \mathbf{w} \mathbf{v}_i(t) + f_1 e_1 [b_i(t) - x_i(t)] + f_2 e_2 [g(t) - x_i(t)] \quad (1)$$

Parámetros:

w: inercia

$v_i(t)$: velocidad previa de la partícula

seleccionarla con cuidado:

$\left\{ \begin{array}{l} \text{grande} \Rightarrow \text{exploración global} \\ \text{pequeña} \Rightarrow \text{exploración local} \end{array} \right.$

Es buena idea que decrezca con las iteraciones.

ALGORITMO PSO

E-1- Actualizar velocidad

2º- La distancia a la mejor posición personal:

$$v_i(t+1) = w v_i(t) + \mathbf{f_1 e_1 [b_i(t) - x_i(t)]} + f_2 e_2 [g(t) - x_i(t)] \quad (1)$$

Parámetros:

e_1 : valor aleatorio en $[0, 1]$

f_1 : parámetro cognitivo

$b_i(t)$: mejor posición personal de la partícula

$x_i(t)$: posición de la partícula

determina la importancia que se da a la experiencia propia.

ALGORITMO PSO

E-1- Actualizar velocidad

3º- la distancia a la mejor solución global.

$$v_i(t+1) = w v_i(t) + f_1 e_1 [b_i(t) - x_i(t)] + \mathbf{f_2 e_2 [g(t) - x_i(t)]} \quad (1)$$

Parámetros:

e_2 : valor aleatorio en $[0, 1]$

f_2 : parámetro social

$g(t)$: mejor posición global

$x_i(t)$: posición de la partícula

determina la importancia que se da
a la experiencia del enjambre.

ALGORITMO PSO

E-1- Actualizar velocidad

El valor calculado se debe ajustar al intervalo válido:
 $[v_{\min}, v_{\max}]$

$$\begin{cases} \text{Si } v_i(t+1) > v_{\max} & \Rightarrow v_i(t+1) = v_{\max} \\ \text{Si } v_i(t+1) < v_{\min} & \Rightarrow v_i(t+1) = v_{\min} \end{cases}$$

Alternativa:

$$\begin{cases} \text{Si } v_i(t+1) > [1 - (t/T)^h] v_{\max} & \Rightarrow v_i(t+1) = [1 - (t/T)^h] v_{\max} \\ \text{Si } v_i(t+1) < -[1 - (t/T)^h] v_{\min} & \Rightarrow v_i(t+1) = -[1 - (t/T)^h] v_{\min} \end{cases}$$

Parámetros:

t: iteración actual **T**: número total de iteraciones
h: cte. positiva

ALGORITMO PSO

E-2- Actualizar posición

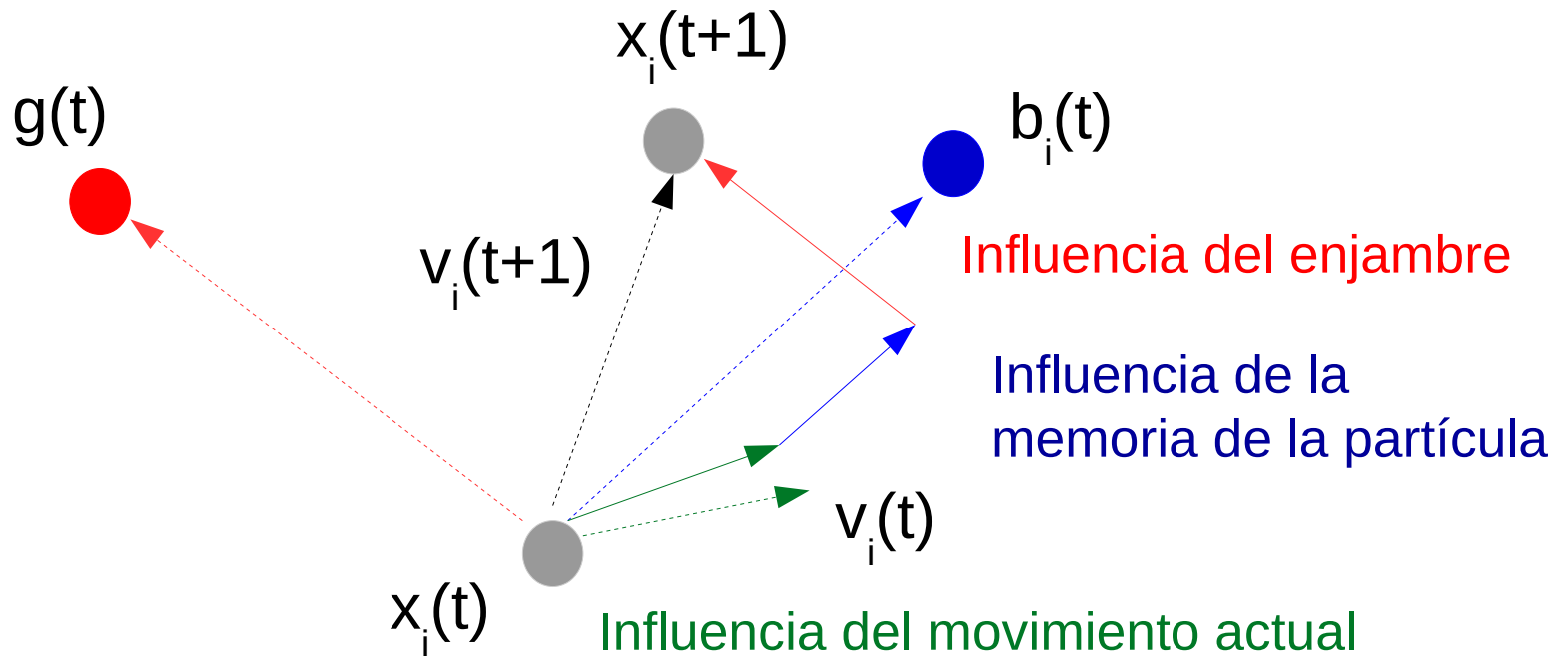
Se calcula la nueva posición de cada partícula i:

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (2)$$

La nueva posición depende de la actual y de la nueva velocidad que acabamos de calcular.

ALGORITMO PSO

E- Actualizar velocidad y posición



Referencias

- Eberhart, R., Kennedy, J. (1995). *A new optimizer using particle swarm theory*. In Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on (pp. 39-43). IEEE.