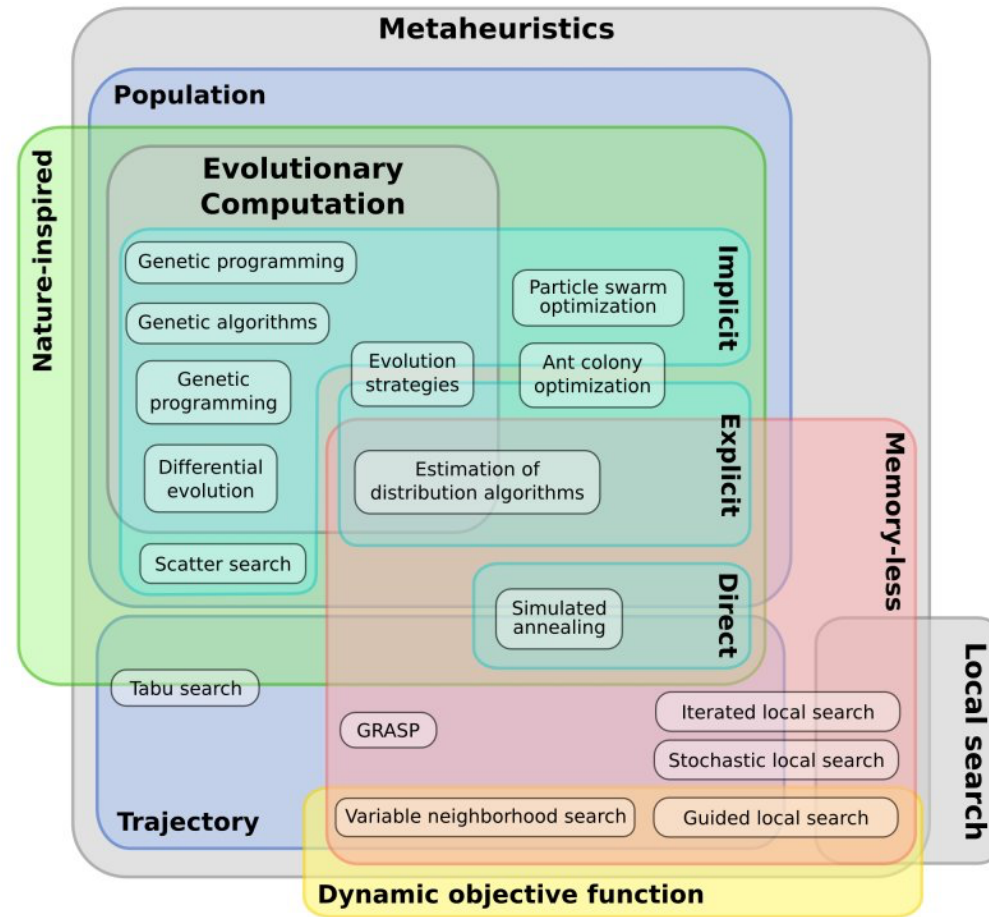


Tema 6. PSO (*Particle Swarm Optimization*)

Profesor: Sebastián Ventura



Índice

- **Introducción a las nubes de partículas**
- Tipos de algoritmos

Introducción a las nubes de partículas

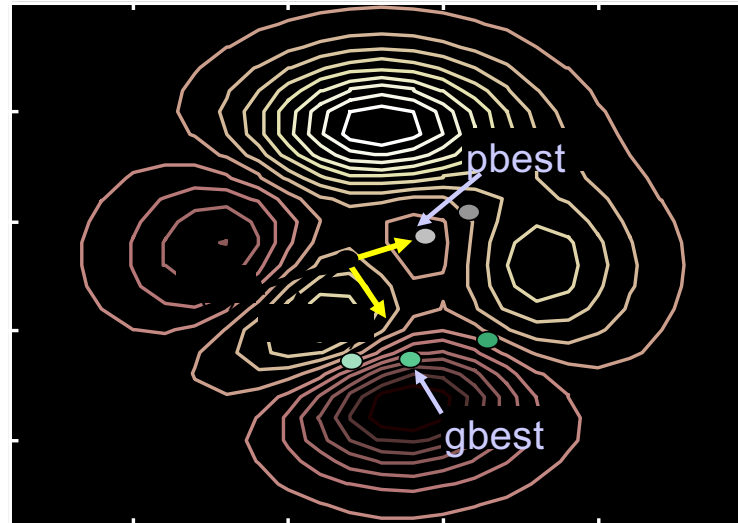
- La *Particle Swarm Optimization* (PSO) es una metaheurística poblacional inspirada en el comportamiento social del vuelo de las bandadas de aves y el movimiento de los bancos de peces
- La población se compone de varias partículas que se mueven por el espacio de búsqueda durante la ejecución del algoritmo
- Este movimiento de cada partícula p depende de:
 - Su mejor posición desde que comenzó el algoritmo (pBest)
 - La mejor posición de las partículas de su entorno (lBest) o de toda la nube (gBest) desde que comenzó el algoritmo
- En cada iteración, se cambia aleatoriamente la velocidad de p para acercarla a las posiciones pBest y lBest/gBest.

Introducción a las nubes de partículas

- Características atribuidas a PSO:
 - Asume un intercambio de información (interacciones sociales) entre los agentes de búsqueda
 - Implementación sencilla y con pocos parámetros
 - Convergencia rápida hacia buenas soluciones
- Cada partícula:
 - Puede interactuar con un numero de vecinos
 - Conoce el valor (fitness) y posición de cada uno de sus vecinos
 - Implementa los principios de comparar e imitar

Introducción a las nubes de partículas

- Cada partícula aprende ajustando su posición y velocidad:
 - Parcialmente atraído a su mejor posición (pBest) hasta el momento
 - Parcialmente atraído a la mejor posición (gBest) de la vecindad



S. Ventura - J.M. Luna

Introducción a las nubes de partículas

- Movimiento de las partículas:
 - Se hace simplemente añadiendo el vector velocidad V_i al vector posición X_i para obtener un nuevo vector posición: $X_i = X_i + V_i$
 - Una vez calculada la nueva posición de la partícula, se evalúa ésta. Si el nuevo fitness es mejor que el que la partícula tenía hasta ahora, $pBestFitness$, entonces:
 - $pBest_i = X_i$
 - $pBestFitness = X_iFitness$
 - 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:

$$\mathbf{v}_i^t \leftarrow \omega^{t-1} \cdot \mathbf{v}_i^{t-1} + \varphi_1 \cdot rand_1 \cdot (\mathbf{mejorpos}_i - \mathbf{x}_i^{t-1}) + \varphi_2 \cdot rand_2 \cdot (\mathbf{mejorpos} - \mathbf{x}_i^{t-1})$$

Introducción a las nubes de partículas

- Vector velocidad:

$$\mathbf{v}_i^t \leftarrow \omega^{t-1} \cdot \mathbf{v}_i^{t-1} + \underbrace{\varphi_1 \cdot rand_1}_{\text{componente cognitivo}} \cdot (\text{mejorpos}_i - \mathbf{x}_i^{t-1}) + \underbrace{\varphi_2 \cdot rand_2}_{\text{componente social}} \cdot (\text{mejorpos} - \mathbf{x}_i^{t-1})$$

componente cognitivo

componente social

\mathbf{x}_i^t es el vector posición de la partícula i en la iteración t

\mathbf{v}_i^t es el vector velocidad de la partícula i en la iteración t

ω^t es el vector de inercia en la iteración t

φ_1 φ_2 son los pesos que controlan los componentes cognitivo y social

$rand_1$ y $rand_2$ son números aleatorios entre 0 y 1

mejorpos_i es la mejor posición encontrada por la partícula i hasta el momento

mejorpos representa la posición de la partícula con la mejor solución

Introducción a las nubes de partículas

- **Componente cognitivo:** indica la decisión que tomará la partícula y depende de su propia experiencia. Representa la distancia entre la posición actual y la mejor conocida por esa partícula
- **Componente social:** apunta la decisión que tomará la partícula en base a la influencia del resto de partículas que componen la nube. Representa la distancia entre la posición actual y la mejor posición encontrada por vecindario

Índice

- Introducción a las nubes de partículas
- **Tipos de algoritmos**

Tipos de algoritmos

- Según la importancia de los pesos cognitivo y social:
 - **Completo**, si $\varphi_1 \varphi_2 > 0$. Tanto el componente cognitivo como el social intervienen en el movimiento
 - **Cognitivo**, si $\varphi_1 > 0$ y $\varphi_2 = 0$. Sólo interviene el componente cognitivo en el movimiento
 - **Social**, si $\varphi_1 = 0$ y $\varphi_2 > 0$. Sólo interviene el componente social en el movimiento
 - **Social exclusivo**, si $\varphi_1 = 0$, $\varphi_2 > 0$, y *mejorpos* no es igual que la posición de la partícula.repre. La posición de la partícula en sí no puede ser la mejor de su entorno
- Según el tipo de vecindario utilizado (cantidad y posición de las partículas que intervienen en el cálculo de la distancia en la componente social):
 - **Local**. Calcula la distancia entre la posición actual de la partícula y la posición de la mejor partícula perteneciente al entorno local de aquella. El entorno local consiste en las partículas inmediatamente cercanas
 - **Global**. La distancia se obtiene entre la posición actual de la partícula y la posición de la mejor partícula considerando la nube completa

Tipos de algoritmos

PSO Clásico

```
t=0
Nube ← Inicializar Nube de Partículas
Mientras no se alcance la condición de parada hacer
    t=t+1
    Para i = 1 hasta tamaño (Nube) hacer
        Evaluar cada partícula  $x_i$  de la Nube
        Si fitness_ $x_i$  es mejor que fitness _mejorposi entonces
            mejorposi ←  $x_i$ 
            fitness _mejorposi ← fitness _x
        Fin Si
        Si fitness _mejorposi es mejor que fitness _mejorpos entonces
            mejorpos ← mejorposi;
            fitness _mejorpos ← fitness _mejorposi
        Fin Si
    Fin Para
    Para i = 1 hasta tamaño (Nube) hacer
        Calcular la velocidad  $v_i$  de  $x_i$ , en base a los valores  $x_i$ , mejorposi y mejorpos
        Calcular la nueva posición de  $x_i$ , de su valor actual y  $v_i$ 
    Fin Para
Fin Mientras
Salida: Devuelve la mejor solución encontrada.
```