

# Algoritmos Evolutivos: Optimización del enjambre de partículas (PSO)

Guillermo Furlan - Sofía Escobar



Universidad del Valle de Guatemala  
Licenciatura en Matemáticas Aplicadas

Noviembre de 2023



## 1 Teoría

- Algoritmos Evolutivos
- PSO

## 2 Referencias

## 1 Teoría

- Algoritmos Evolutivos
- PSO

## 2 Referencias

# ¿Qué son los algoritmos evolutivos?

Un algoritmo evolutivo es un enfoque de optimización y búsqueda inspirado en la teoría de la evolución biológica. Los elementos clave de un algoritmo evolutivo incluyen:

- Población inicial
- Evaluación
- Selección
- Cruce, mutación o actualización
- Iteración

Algunos Métodos Evolutivos muy conocidos son:

- PSO
- Algoritmos genéticos
- Búsqueda Dispersa

Un algoritmo Particle Swarm Optimization (PSO) es un algoritmo evolutivo de optimización inspirado en el comportamiento social de la naturaleza y movimientos dinámicos con comunicaciones de insectos, pájaros y peces. El algoritmo utiliza un grupo de agentes, llamados partículas, que forman un enjambre y se mueven en un espacio de búsqueda en busca de la mejor solución.



- **Partículas en Vuelo:** El algoritmo inicia con un conjunto de partículas que representan soluciones potenciales; estas se desplazan en el espacio de búsqueda en busca de la solución óptima.
- **Ajuste de Movimiento de las Partículas:** Cada partícula en el espacio de búsqueda ajusta su "vuelo". Este ajuste se basa en la experiencia de vuelo de la propia partícula y en la de otras partículas en el enjambre.
- **Tres Parámetros de Cada Partícula:** Cada partícula tiene tres parámetros clave: posición, velocidad y la mejor posición previa, que se llama "mejor posición personal". La partícula con la mejor aptitud en todo el enjambre se conoce como "mejor posición global".
- **Conjunto de Partículas en Vuelo:** El conjunto de partículas en vuelo representa soluciones cambiantes en el espacio de búsqueda.

- **Ajuste Dinámico de la Velocidad:** Cada partícula ajusta dinámicamente su velocidad en función de su propia experiencia y la de sus compañeras.
- **Seguimiento de Mejores Soluciones:** Cada partícula realiza un seguimiento de dos tipos de mejores soluciones:
  - Su mejor solución personal (pbest).
  - La mejor solución encontrada por cualquier partícula en todo el enjambre (gbest).
- **Modificación de la Posición de las Partículas:** Las partículas modifican su posición en función de varios factores, como su posición actual, su velocidad actual y las distancias a sus mejores soluciones personales (pbest) y a la mejor solución global (gbest).

## Ventajas:

- 1 Insensible a la escala de las variables de diseño.
- 2 Implementación simple.
- 3 Libre de derivadas.
- 4 Muy pocos parámetros del algoritmo.
- 5 Algoritmo de búsqueda global muy eficiente.

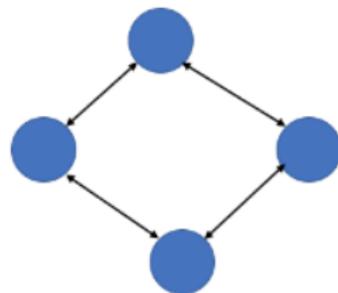
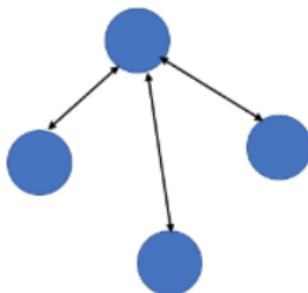
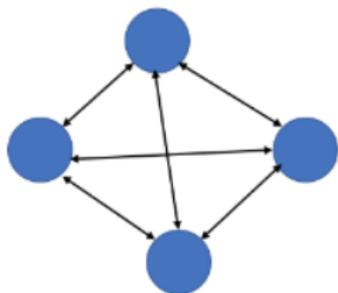
## Desventajas:

- 1 Convergencia lenta en la etapa de búsqueda refinada (debilidad en la capacidad de búsqueda local).
- 2 Aumentar la velocidad de búsqueda requiere mucha memoria.

# PSO vs Algoritmo Genético

La principal diferencia es que los Algoritmos Genéticos, en cada iteración, "randomizan" la población con el fin de buscar la mejor solución, mientras que en PSO, la misma población va cubriendo el espacio o moviéndose para encontrar el óptimo.

# Topología



# Actualización de la Velocidad

## Diversificar

Busca la solución con las potenciales mejores soluciones

$$V_i^{t+1} = W \cdot V_i^t + c_1 U_1^t (P_{b_1}^t - P_i^t) + c_2 U_2^t (g_b^t - P_i^t)$$

## Inercia

Hace a la partícula moverse en la misma dirección y velocidad

## Influencia Personal

Mejora la posición basada en las mejores anteriores

## Influencia Social

Hace a la partícula moverse en función de la mejor posición de la vecindad

# Actualización de la posición

La actualización de la posición en el algoritmo de Optimización por Enjambre de Partículas (PSO) se realiza mediante la siguiente fórmula general:

$$\text{posicion}[i, j] = \text{posicion}[i, j] + \text{velocidad}[i, j]$$

Donde:

- $i$  representa el índice de la partícula.
- $j$  representa el índice de la dimensión (en el espacio de búsqueda) de la partícula.

Esta fórmula indica que la nueva posición de una partícula en la dimensión  $j$  se obtiene sumando la velocidad actual de la partícula en esa dimensión a su posición actual en esa dimensión.

## 1 Mejor Posición Local de Cada Partícula:

- Se compara el valor de aptitud de la posición actual de una partícula con el valor de aptitud de su mejor posición histórica (mejor posición local).

$$\text{best\_pos}[i, j] = \begin{cases} \text{pos}[i, j], & \text{si } \text{fit}(\text{pos}[i, j]) < \text{fit}(\text{best\_pos}[i, j]) \\ \text{best\_pos}[i, j], & \text{en caso contrario} \end{cases}$$

## 2 Mejor Posición Global del Enjambre:

- Se compara el valor de aptitud de la mejor posición local de cada partícula con el valor de aptitud de la mejor posición global del enjambre.

$$\text{swarm\_best\_pos}[j] = \text{best\_pos}[\text{best índice global}, j]$$

donde índice de la partícula con la mejor posición global es el índice de la partícula cuya mejor posición local tiene el valor de aptitud más bajo en esa iteración.

- 1 Inicializar un enjambre de partículas con posiciones y velocidades aleatorias en el espacio de soluciones.
- 2 Para cada partícula, evaluar la aptitud de la partícula.
- 3 Calcular la velocidad de la partícula.
- 4 Actualizar la posición de la partícula.
- 5 Comparar la aptitud de la partícula con su mejor aptitud personal (pBest). Si la aptitud actual es mejor, entonces establecer la aptitud actual como la nueva pBest.
- 6 Elegir la partícula con la mejor aptitud de todas las partículas como el mejor global (gBest).
- 7 Si se cumple el criterio de parada, entonces terminar y devolver gBest.
- 8 Si no, volver al paso 2.

- 1 **Número máximo de iteraciones:** El algoritmo se detiene después de un número predefinido de iteraciones.
- 2 **Sin cambios en gBest:** Si el mejor global (gBest) no cambia (o el cambio es muy pequeño) durante un cierto número de iteraciones, entonces el algoritmo se detiene.
- 3 **Tiempo computacional:** El algoritmo se detiene después de un cierto tiempo de ejecución.

# Ejemplos: Función De Jong

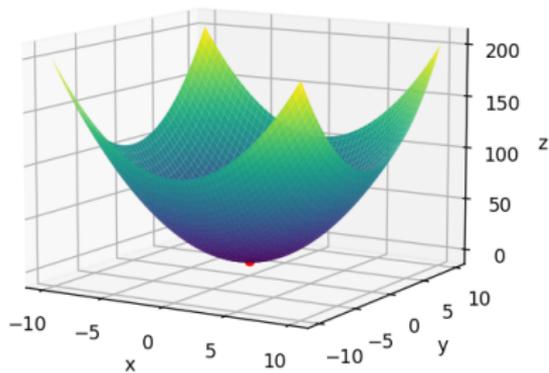


Figura: Superficie de la función

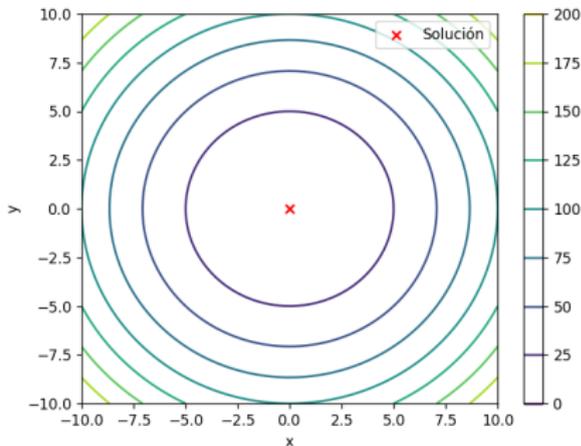


Figura: Contorno de la función

# Ejemplos: Función De Jong

Partículas	Iteraciones	Solución	Ajuste
10	100	[ 5.63E-10, -2.98E-12]	1.36E-21
30	100	[1.12E-13, 3.33E-13]	1.23E-15
50	100	[-4.68E-13, -3.59E-12]	1.32E-23
100	100	[3.16E-12, 4.82E-13]	1.22E-25
100	200	[1.01E-24, 7.49E-24]	7.55E-49
100	500	[-4.74E-59, -4.47E-59]	4.25E-117

Cuadro: Resultados de las simulaciones.

# Ejemplos: Función de Rastrigin

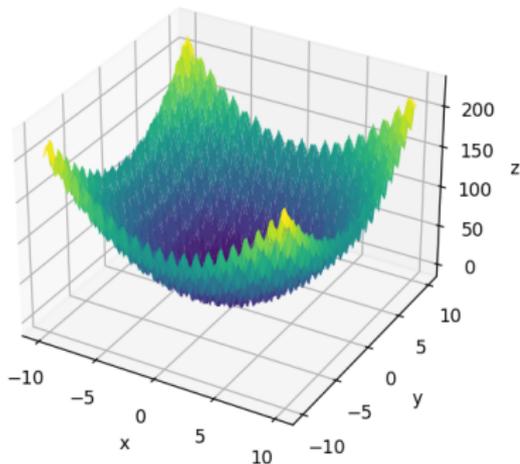


Figura: Superficie de la función

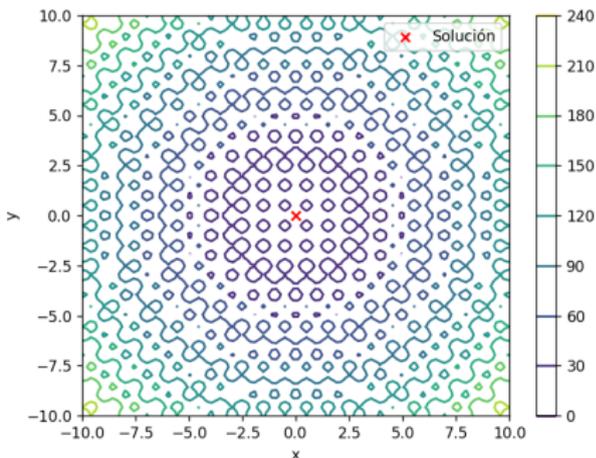


Figura: Contorno de la función

# Ejemplos: Función de Rastrigin

Partículas	Iteraciones	Solución	Ajuste
10	50	[-9.95E-01, -1.14E-06]	9.95E-01
30	50	[0.0002409, 0.003535]	1.40E-05
50	50	[8.41E-06, 7.23E-07]	6.15E-10
100	50	[3.14E-06, 5.33E-06]	1.38E-10
30	200	[-3.05E-09, 9.87E-10]	5.00E-09
30	1000	[ 3.006E-09, -6.19E-10]	1.00E-21

Cuadro: Resultados de las simulaciones.

# Ejemplos: Función de Rosenbrock

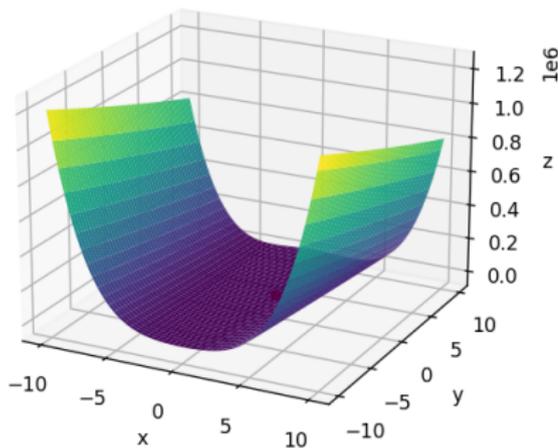


Figura: Superficie de la función

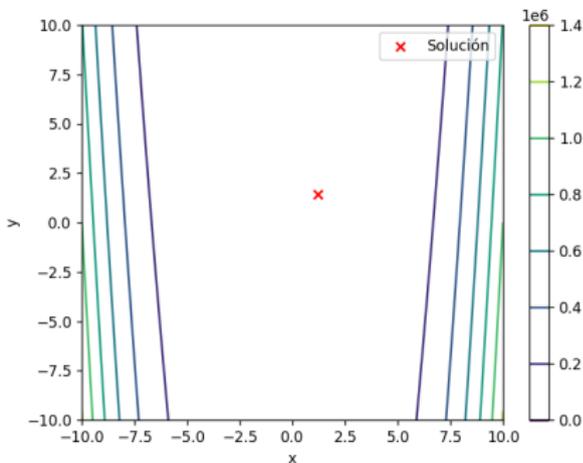


Figura: Contorno de la función

# Ejemplos: Función de Rosenbrock

Partículas	Iteraciones	Solución	Ajuste
10	100	[0.29135647, 0.07755668]	5.08E-01
30	100	[1.03520197, 1.06086752]	1.79E-03
50	100	[0.99999704, 0.99999513]	1.74E-15
100	100	[1.00000001, 1.00000002]	2.38E-15
30	200	[1.00001903, 1.00003655]	5.90E-10
30	500	[1. 1.]	0.00E+00

Cuadro: Resultados de las simulaciones.

Código del algoritmo -  
<https://github.com/sofiaesc07/Proyecto2-MN2.git>

- 1 Teoría
  - Algoritmos Evolutivos
  - PSO

- 2 Referencias



S. Sanyal

*An Introduction to Particle Swarm Optimization (PSO algorithm).*

Analytics Vidhya, 9 de agosto de 2023.

<https://www.analyticsvidhya.com/blog/2021/10/an-introduction-to-particle-swarm-optimization-algorithm/#h-neighborhood-topologies>



A. Michaloglou, N. L. Tsitsas

*Particle Swarm Optimization Algorithms with Applications to Wave Scattering Problems.*

En IntechOpen eBooks, 2022.

<https://doi.org/10.5772/intechopen.97217>