

# Evolución Diferencial Y PSO: Historia, Conceptos y Explicación

Cosijopii García

November 27, 2024

- 1 Historia de la Evolución Diferencial
- 2 Explicación de la Evolución Diferencial
- 3 Pseudocódigo de DE
- 4 Explicación Visual
- 5 Beneficios y Limitaciones
- 6 Aplicaciones
- 7 Conclusión

- Propuesta por **Storn y Price** en 1995.
- Inspirada en la evolución biológica:
  - Selección natural.
  - Cruce.
  - Mutación.
- Diseñada para optimización global en problemas complejos.
- Aplicaciones iniciales en ingeniería y ciencia computacional.

# ¿Qué es la Evolución Diferencial?

- Es un algoritmo de optimización basado en principios evolutivos.
- Diseñado para funciones no lineales y multimodales.
- Se centra en una población de soluciones que evolucionan iterativamente.

# Pasos principales

- 1 **Mutación:** Generación de nuevos vectores a partir de diferencias entre individuos.
- 2 **Recombinación:** Mezcla de características entre padres y vectores mutantes.
- 3 **Selección:** Retención del individuo más apto en cada iteración.

## Pseudocódigo básico de Evolución Diferencial:

Inicializar poblacion P con N vectores aleatorios

Repetir

    Para cada vector xi en P:

        Seleccionar tres vectores aleatorios (a, b, c)

        Crear vector mutante  $v = a + F * (b - c)$

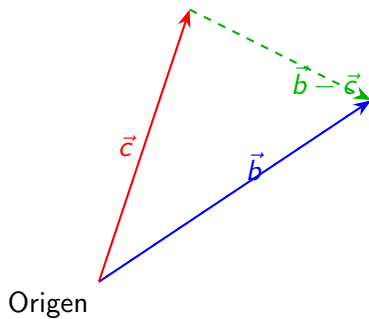
        Crear vector hijo u mezclando xi y v

        Si  $f(u) < f(xi)$ :

            Reemplazar xi con u

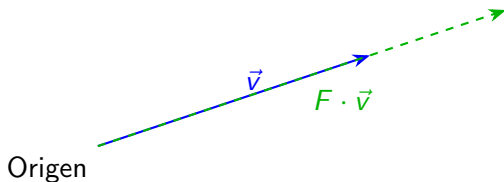
Hasta alcanzar criterio de parada

# Diferencia de vectores



- La diferencia entre dos vectores  $\vec{b} - \vec{c}$  apunta desde  $\vec{c}$  hacia  $\vec{b}$ .
- Este vector es fundamental para el operador de mutación en DE.

# Multiplicación por un escalar

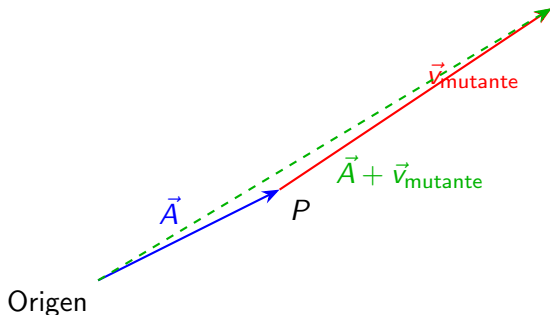


- Multiplicar un vector por un escalar  $F$  ajusta su magnitud.
- Esto controla la exploración del espacio de búsqueda.

$$\vec{v}_{\text{mutante}} = \vec{a} + F \cdot (\vec{b} - \vec{c})$$



# Suma de un vector $\vec{A}$ y el vector mutante $\vec{v}_{\text{mutante}}$



- La suma de vectores sigue el método "punta a cola".
- $\vec{A} + \vec{v}_{\text{mutante}}$  es el vector resultante que apunta desde el origen hasta la punta del vector mutante.

# Beneficios de la Evolución Diferencial

- Fácil de implementar.
- Eficiente para funciones multimodales y de alta dimensión.
- No requiere derivadas de las funciones objetivo.

- Puede ser lento en converger para problemas muy complejos.
- Requiere parametrización adecuada (tamaño de población, factor de escala).
- Puede quedarse atrapado en óptimos locales si no se ajusta correctamente.

# Aplicaciones de la Evolución Diferencial

- Ajuste de parámetros en modelos de aprendizaje automático.
- Optimización en diseño de circuitos eléctricos.
- Resolución de problemas de planificación y logística.
- Aplicaciones en robótica y simulaciones físicas.

- La Evolución Diferencial es una herramienta versátil y efectiva para optimización global.
- Su simplicidad permite una amplia gama de aplicaciones en problemas complejos.
- Ajustar los parámetros es crucial para garantizar su éxito.

**¿Preguntas?**

# ¿Qué es PSO?

- Algoritmo metaheurístico inspirado en el comportamiento colectivo de bandadas de aves y bancos de peces.
- Introducido por Kennedy y Eberhart en 1995.
- Busca soluciones óptimas explorando el espacio de búsqueda mediante un grupo de partículas (individuos).
- Cada partícula tiene una posición, una velocidad y una memoria de la mejor posición visitada.

# Funcionamiento de PSO

- 1 Inicializar una población de partículas con posiciones y velocidades aleatorias.
- 2 Evaluar la función objetivo para cada partícula.
- 3 Actualizar las posiciones y velocidades:

$$v_i^{t+1} = \omega v_i^t + c_1 r_1 (pBest_i^t - x_i^t) + c_2 r_2 (gBest^t - x_i^t)$$

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

- 4 Actualizar las mejores soluciones individuales ( $p_i$ ) y globales ( $g$ ).
- 5 Repetir hasta cumplir un criterio de parada (número de iteraciones o precisión deseada).

- $\omega$  (inercia): Controla la influencia de la velocidad previa.
- $c_1, c_2$  (coeficientes de aprendizaje): Pesos para la atracción hacia las mejores soluciones individuales y globales.
- $r_1, r_2$  (valores aleatorios): Generados en cada iteración para diversificar el movimiento.
- Tamaño de la población: Número de partículas.
- Número de iteraciones: Límite máximo de pasos de optimización.



# Ejemplo de Actualización

- Supongamos una partícula  $i$  en un espacio de búsqueda 2D:

$$x_i = [2, 3], \quad v_i = [0.5, -0.2]$$

- Mejores posiciones conocidas:

$$p_i = [1.8, 2.5], \quad g = [1.5, 2.0]$$

- Parámetros:

$$\omega = 0.7, \quad c_1 = 1.5, \quad c_2 = 2.0, \quad r_1 = 0.4, \quad r_2 = 0.7$$

- Nuevas velocidades y posiciones:

$$v_i^{t+1} = 0.7[0.5, -0.2] + 1.5(0.4)[1.8-2, 2.5-3] + 2.0(0.7)[1.5-2, 2.0-3]$$

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

# Características de PSO

- No requiere derivadas de la función objetivo.
- Convergencia rápida en problemas de baja dimensión.
- Fácil de implementar y ajustar.
- Buena exploración y explotación del espacio de búsqueda.

- Tendencia a quedarse atrapado en óptimos locales, especialmente en problemas de alta dimensión.
- Sensibilidad a la configuración de parámetros ( $\omega$ ,  $c_1$ ,  $c_2$ ).
- Desempeño decrece en problemas con restricciones complicadas o espacios de búsqueda discontinuos.

- **PSO con Velocidad Adaptativa:** Ajusta dinámicamente  $\omega$ ,  $c_1$ ,  $c_2$ .
- **Multi-Swarm PSO:** Utiliza múltiples enjambres para explorar diferentes regiones del espacio.
- **PSO Híbrido:** Combina PSO con otros algoritmos como GA o SA.

# Ejemplo: Optimización de la función Rastrigin

- Función objetivo:

$$f(x) = 10n + \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i))$$

- Dominio:  $x_i \in [-5.12, 5.12]$ .
- Partículas comienzan en posiciones aleatorias y convergen hacia el mínimo global en  $(0, 0, \dots, 0)$ .

- PSO es un algoritmo eficiente y versátil para problemas de optimización continua y discreta.
- Configurar adecuadamente los parámetros es clave para obtener buenos resultados.
- Las variantes de PSO permiten adaptarlo a problemas más complejos.