

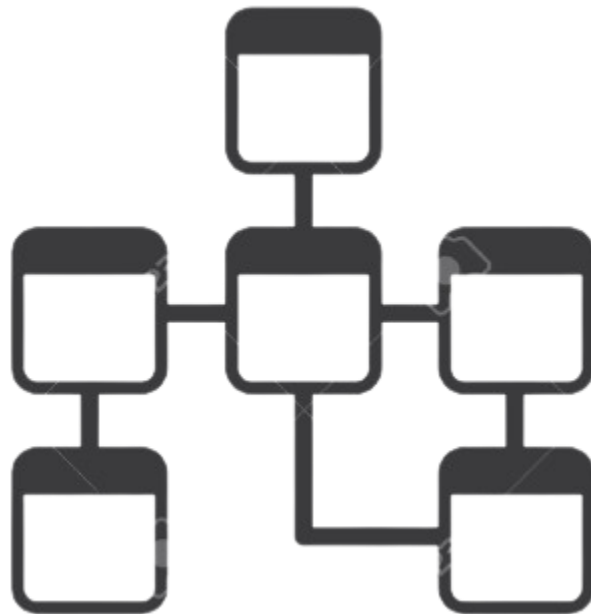


TECNOLÓGICO
NACIONAL DE MÉXICO



Instituto Tecnológico de Culiacán

Ingeniería en Tecnologías de la Información y Comunicaciones



Materia: Tópicos de IA

Evolución Diferencial

Maestro: Zuriel Dathan Mora Félix

Alumno: Daniel Alejandro De Los Santos Cuevas

¿Qué es?

Es un **algoritmo de optimización heurística** utilizado principalmente para resolver problemas de optimización **no lineales, no diferenciables y multimodales** (es decir, con múltiples óptimos locales). Es un tipo de **algoritmo evolutivo**, como los algoritmos genéticos, pero con un enfoque diferente en la forma de generar nuevas soluciones.

Componentes

La evolución diferencial trabaja con una **población de soluciones candidatas**, llamadas **vectores**, que se actualizan en cada generación mediante operaciones inspiradas en la evolución natural:

1. Inicialización:

- Se genera una población inicial aleatoria de soluciones.

2. Mutación:

- Se seleccionan aleatoriamente tres vectores distintos de la población, por ejemplo, A,B,C, y se crea un nuevo vector mutado:

$$V = A + F \cdot (B - C)$$

- Aquí, F es un **factor de escala**, normalmente entre 0 y 2, que controla la magnitud del cambio.

3. Cruzamiento (recombinación):

- Se mezcla el vector mutado V con el vector objetivo (el vector actual que está siendo evaluado en la población) para generar un nuevo **vector de prueba U**.
- Esta mezcla se hace controladamente usando una **tasa de cruce CR**.

4. Selección:

- Si el nuevo vector de prueba U tiene mejor rendimiento (menor función objetivo, si es minimización) que el vector actual, se reemplaza.

5. Repetición:

- El proceso se repite durante un número determinado de generaciones o hasta que se cumpla un criterio de parada.

Características

1. Basado en población:

- Trabaja con múltiples soluciones simultáneamente, lo que permite una mejor exploración del espacio de búsqueda.

2. Heurístico:

- No requiere derivadas ni conocimiento interno del problema.
- Utiliza números aleatorios, lo cual le da una naturaleza probabilística.

3. Simple de implementar:

- Tiene pocos parámetros y su lógica es directa.

4. Buena capacidad de exploración y explotación:

- Puede evitar caer fácilmente en óptimos locales, gracias a la mutación y recombinación.

5. Versatilidad:

- Se adapta a problemas continuos y puede modificarse para problemas discretos o combinatorios.

6. Parámetros clave:

- **F (factor de escala):** controla la amplitud de la mutación.
- **CR (tasa de cruce):** controla la cantidad de mezcla entre padres e hijos.

- **NP (tamaño de la población):** balancea exploración y tiempo de cómputo.