Lógica, algoritmia y computación

Problema algoritmo genético

Giordano Castilla García y Fernando Melero Muñoz

2024 / 2025

Universidad Camilo José Cela

Indice

1 Introducción

Esta práctica consiste en crear un algoritmo genético para resolver el Problema del viajero (TSP, Travelling Salesman Problem). El TSP es un problema clásico de optimización combinatoria ampliamente estudiado en matemáticas, informática y ciencias de la operación, y tiene una gran importancia debido a sus aplicaciones prácticas en campos como la logística, la planificación de rutas, y diseño de circuitos, entre otros, sin embargo, su solución aumenta de complejidad a medida que aumenta el número de “ciudades”.

En este caso, haremos uso del algoritmo genético para encontrar la ruta más corta de entre todas las rutas posibles de una determinada ubicación geográfica. En el archivo .tsp tenemos un listado con todas las coordenadas de las que tenemos que calcular la distancia entre estas y de entre todas ellas obtener la ruta más corta.

2 ¿Qué es un Algoritmo Genético (AG)?

Los algoritmos genéticos son técnicas de optimización que simulan la evolución natural para resolver problemas.

En este caso, se basa en generar soluciones iniciales, evaluarlas con una función “de aptitud (fitness)” y mejorarlas mediante mutaciones y cruzamientos. El proceso es iterativo y se enfoca en seleccionar las mejores soluciones de cada generación.

3 ¿Qué es el Problema del Agente Viajero (TSP)?

Es un problema clásico de optimización que consiste en encontrar el recorrido más corto para visitar un conjunto de ciudades y volver al punto de origen sin repetir ciudades.

En este caso, el problema será resuelto usando un algoritmo genético.

4 Método de resolución

1. Leer las coordenadas de las ciudades:

Crear una función para leer un archivo .tsp y almacenar las coordenadas en una estructura adecuada ({id, x, y}).

2. Generar población inicial:

Crear rutas iniciales aleatorias (permutaciones de las ciudades/padres) para formar la población inicial.

3. Bucle del algoritmo genético:

Iterar los siguientes pasos hasta cumplir el criterio de parada:

Definir los siguientes parámetros:

- Tamaño de la población.
- Criterio de parada (número máximo de generaciones).

i. Calcular distancias y aptitud:

Implementar una función que calcule las distancias entre dos ciudades usando la fórmula euclidiana.

Dada una ruta completa, calcule su distancia total y el valor de aptitud.

ii. Selección por ruleta:

Seleccionar padres basados en la aptitud.

iii. Cruzamiento (por ciclos):

Implementar el método de cruzamiento por ciclos para generar hijos válidos a partir de dos rutas padres.

iv. Mutación (por inversión):

Reemplazar la población actual con los nuevos individuos generados.

v. Crear la nueva población:

Calcular la distancia total de cada ruta y su aptitud.

vi. Verificar el criterio de parada:

Continuar si no se cumple el criterio; detener si se cumple.

4. Resultado final:

Mostrar la mejor ruta encontrada y su distancia.

5 Resultados

El comportamiento y los resultados del algoritmo genético pueden variar dependiendo de los valores para los parámetros clave, como el tamaño de la población, la probabilidad de mutación y el número de generaciones.

Debido a esta dependencia, conviene ejecutar el algoritmo varias veces y observar los resultados. Entre los parámetros a tener en cuenta están:

- **Tamaño de la población:** Poblaciones más grandes tienden a explorar mejor el espacio de búsqueda, pero aumentan el tiempo de ejecución.
- **Probabilidad de mutación:** Un valor bajo puede limitar la diversidad genética, mientras que un valor alto puede generar demasiada aleatoriedad.
- **Número de generaciones:** Aumentar las generaciones permite refinar las soluciones, pero con un costo computacional adicional.

El análisis iterativo y el ajuste manual de estos parámetros son esenciales para maximizar la efectividad del algoritmo y adaptarlo a problemas específicos.

Bibliografía

NumPy tutorial. (s/f). W3schools.com. Recuperado el 16 de diciembre de 2024, de https://www.w3schools.com/python/numpy/default.asp

Wikipedia contributors. (2024a, septiembre 25). Computational complexity theory. Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Computational_complexity_theory&oldid=1247765879

Wikipedia contributors. (2024b, diciembre 5). Travelling salesman problem. Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Travelling_salesman_problem&oldid=1261381073