



TECNOLÓGICO NACIONAL DE MÉXICO



NOMBRE DE LA MATERIA: Tópicos de Inteligencia Artificial

TAREA DE VALIDACIÓN III

ALUMNOS: García Sánchez Sergio Jesús
Darío Corrales Palazuelos

CARRERA: Ingeniería en Sistemas Computacionales

NOMBRE DEL MAESTRO: Zuriel Dathan Mora Félix

1. Objetivo

El propósito de esta actividad es validar la comprensión y habilidades en el uso de algoritmos genéticos a través de la aplicación práctica en Python. Se implementó un algoritmo genético para resolver un problema de optimización de rutas (similar al problema del viajero o TSP). El ejercicio permite reforzar los fundamentos de la evolución artificial y la optimización heurística.

2. Descripción General del Algoritmo

El algoritmo genético (AG) desarrollado busca encontrar la ruta más corta que conecta un conjunto de municipios, emulando un proceso evolutivo inspirado en la selección natural de Darwin. El programa fue diseñado de manera modular para separar claramente cada una de las fases del proceso genético: creación de población, evaluación de aptitud, selección, cruce, mutación y generación sucesiva.

A diferencia de métodos deterministas, el AG explora múltiples soluciones simultáneamente, lo que lo hace robusto frente a óptimos locales. El código fue implementado completamente desde cero, utilizando bibliotecas básicas de Python como random, numpy, pandas y matplotlib.

3. Componentes Principales y Desglose del Funcionamiento

El algoritmo se compone de diversas etapas, cada una con una función específica dentro del proceso evolutivo. A continuación se detalla su funcionamiento interno:

3.1 Inicialización de la Población

El algoritmo comienza generando una población inicial de posibles soluciones (rutas). Cada individuo de la población representa un recorrido completo que pasa por todos los municipios una sola vez. Para ello, se emplea una permutación aleatoria de la lista de municipios utilizando la función random.sample(). Esta diversidad inicial es crucial para explorar diferentes regiones del espacio de búsqueda.

3.2 Función de Aptitud

La clase Aptitud calcula la calidad de cada individuo. Su función rutaApta() determina el valor de aptitud como el inverso de la distancia total recorrida. Una menor distancia implica una mayor aptitud. Internamente, la función distanciaRuta() recorre la ruta sumando las distancias euclidianas entre cada par consecutivo de municipios, cerrando el ciclo al conectar el último con el primero.

Esta evaluación cuantifica qué tan eficiente es cada solución dentro del conjunto poblacional.

3.3 Clasificación y Selección

Las rutas se clasifican de acuerdo con su aptitud usando la función `clasificacionRutas()`, que devuelve una lista ordenada de mejor a peor individuo. Posteriormente, la función `seleccionRutas()` aplica una combinación de elitismo y selección por ruleta. El elitismo preserva un porcentaje fijo de los mejores individuos sin alteración, garantizando que las mejores soluciones no se pierdan. La selección por ruleta introduce diversidad, permitiendo que individuos menos aptos aún tengan probabilidad de reproducirse, lo cual evita la convergencia prematura.

3.4 Crossover o Reproducción

En la etapa de cruce, se combinan dos rutas (padres) para generar una nueva (hijo). El método utilizado es un 'Ordered Crossover (OX)', donde se selecciona aleatoriamente un segmento del primer progenitor y se completan los espacios restantes con el orden de ciudades del segundo progenitor que no fueron incluidas. De esta manera, el hijo hereda parcialmente la estructura de ambos padres, manteniendo la validez de la ruta (sin duplicados).

3.5 Mutación

La función `mutacion()` introduce pequeñas variaciones aleatorias en la descendencia para mantener la diversidad genética. El método implementado es la 'mutación por intercambio' (swap mutation), donde dos municipios de la ruta intercambian posiciones con una pequeña probabilidad definida por `razonMutacion`. Esto permite escapar de óptimos locales y explorar nuevas configuraciones.

3.6 Nueva Generación

La función `nuevaGeneracion()` integra todas las etapas anteriores para producir una nueva población a partir de la actual. Primero evalúa la aptitud, luego selecciona los padres, aplica el cruce y finalmente introduce mutaciones. Cada generación evoluciona hacia mejores soluciones, lo que se refleja en la disminución progresiva de la distancia promedio.

3.7 Ciclo Evolutivo Completo

La función principal `algoritmoGenetico()` controla el proceso evolutivo a lo largo de un número determinado de generaciones. Durante la ejecución, imprime la distancia inicial, la mejor distancia actual y la mejor global cada cierto número de generaciones. Al finalizar, devuelve la ruta más corta encontrada, junto con una

visualización generada por matplotlib que muestra el recorrido optimizado entre los municipios.

4. Resultados

Al ejecutar el algoritmo con una población de 100 individuos, 20 seleccionados por elitismo, una tasa de mutación de 1% y 500 generaciones, se observó una convergencia estable hacia una distancia final de 24.7624 unidades. Esto indica que el algoritmo encontró una configuración de ruta significativamente optimizada con respecto a las iniciales.

Mejor ruta encontrada:

$[(37.3891, -5.9845), (43.3623, -5.845), (41.6488, -0.8891), (41.3784, 2.1925), (39.4699, -0.3763), (40.4168, -3.7038)]$

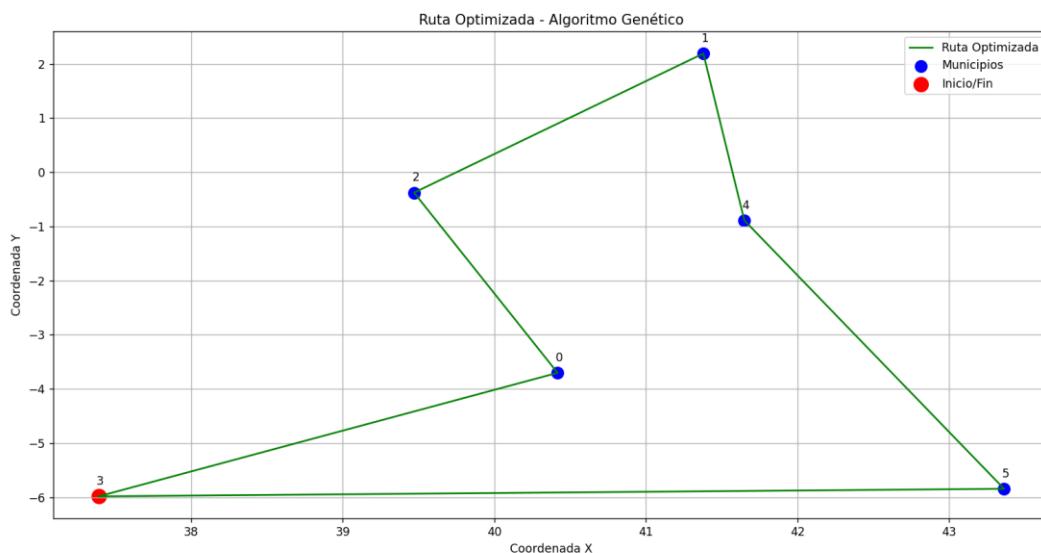


Figura 1. Ruta optimizada generada por el algoritmo genético.

5. Conclusiones

El algoritmo genético demostró ser una herramienta eficaz para la resolución de problemas de optimización combinatoria. Su naturaleza evolutiva permite aproximarse a soluciones cercanas al óptimo global mediante una búsqueda no determinista. La modularización del código, así como la documentación detallada, facilitaron la comprensión de cada proceso interno. Además, la visualización final ayuda a validar de forma intuitiva la calidad de la solución encontrada.

El conocimiento adquirido en esta práctica refuerza la comprensión de los mecanismos evolutivos artificiales, su relación con los algoritmos de optimización y su aplicabilidad en problemas reales, como la logística, el diseño de rutas y la planificación eficiente.