

TSP

Resumen

En este estudio, se emplean algoritmos bioinspirados, como el algoritmo genético y la Optimización por Colonia de Hormigas (OCH), para abordar el Problema del Agente Viajero. Se compara el desempeño de estos algoritmos con la solución óptima obtenida a través de un modelo matemático resuelto en GAMS. Los resultados muestran que el algoritmo genético se acerca a la solución óptima del modelo matemático, aunque mejorar su precisión requiere aumentar el tamaño de la población o el número de generaciones, lo que conlleva un costo computacional adicional. En cuanto al tiempo de ejecución, la Optimización por Colonia de Hormigas resulta ligeramente más rápida que el algoritmo genético en diferentes iteraciones. Sin embargo, el algoritmo genético proporciona resultados más cercanos a la solución óptima en ambas instancias de los algoritmos.

Introducción

El **Problema del Agente Viajero** (TSP) trata de encontrar la mejor ruta para visitar varios lugares y regresar al punto de inicio. El objetivo es encontrar la forma más económica de hacerlo, ya sea minimizando la distancia o los costos [1].



Un **algoritmo genético** es un método de búsqueda y optimización que se inspira en la teoría de la evolución biológica propuesta por Charles Darwin. A través de procesos de selección, cruces y mutaciones, los individuos más aptos son identificados y reproducidos, lo que da lugar a una nueva generación de soluciones adaptadas y mejoradas [2].



La **Optimización por Colonia de Hormigas** (OCH) [3] se basa en la forma en que las hormigas encuentran rutas óptimas para el transporte de alimentos, y se ha convertido en una herramienta en la búsqueda de soluciones efectivas desde logística y planificación de rutas hasta diseño de redes y otros problemas de optimización.



Descripción del problema

En el contexto de este problema de optimización de rutas, se dispone de una lista de 100 direcciones, de las cuales se toman n direcciones que deben visitarse una sola vez, comenzando y terminando en un punto fijo. El objetivo principal es determinar la ruta más corta que cumple con estas restricciones. Para abordar esta problemática, se emplearán algoritmos bioinspirados, específicamente el algoritmo genético y el enfoque basado en Optimización de Colonias de Hormigas (OCH) y comparándolos con la solución óptima del modelo matmático resuelto en GAMS.

Parámetros de algoritmos bioinspirados

Algoritmo Genético

- Tipo de cromosoma: **tiene valores enteros entre 0 y n, donde n es el número de clientes menos 1**
- Longitud: **n, número de puntos**
- Criterio de inicialización: **Aleatoria.**
- Infactibilidad: **No**
- Criterio de paro: **100 generaciones**
- Tamaño de población: **1000**
- Función fitness: **Función objetivo, minimizar distancia**
- Criterio de selección: **Por torneo**
- Probabilidad de cruce: **0.7**
- Puntos de cruce: **2**
- Lugar de cruce: **Aleatorio**
- Probabilidad de mutación: **0.02**
- Criterio de reemplazo: **Reemplazo de la generación.**

Colonia de Hormigas

- Número de ciudades = 100 **Cambiar si se quiere 40 ciudades o 100 ciudades**
- Número de hormigas = 100 **Menor o igual al número de ciudades**
- Número de iteraciones = 100
- alpha = 1.0 **Influencia de la pheromona**
- beta = 2.0 **Influencia de heurístico**
- ro = 0.1 **Evaporación de feromonas**

Resultados

	100 Nodos	
	Hormigas	
Iteraciones:	100	1000
Distancia (km):	241.91	192.88
Tiempo (min):	0:01:51	0:18:35
	Genético	
	100	1000
Distancia (km):	236.16	179.05
Tiempo (min):	0:02:06	0:21:10

40 Nodos			
Nodos	Gams	Hormigas	Genetico
1	62.22	75.71	74.71
2	71.66	78.58	84.53
3	58.56	67.8	66.11
4	67.95	85.75	77.68
5	58.37	68.96	70.24
6	70.5	91.4	79.33
7	76.26	86.54	92.42
8	77.42	90.76	86.65
9	65.36	77.85	74.12
10	59.47	73.35	70.07
Costo (Distancia recorrida en km)			

Conclusiones y observaciones

Los resultados obtenidos en el algoritmo genético se aproximan a lo obtenido en el modelo matemático de GAMS. Podemos mejorar la precisión del algoritmo aumentando el tamaño de la población o el el número de generaciones, pero eso tiene un costo computacional que por el momento no contamos, por lo que lo adaptamos a nuestros recursos. En cuanto a tiempo de ejecución, el OCH fue un tanto más rápido que el genético, teniendo una diferencia de tiempo de 15 segundos en 100 iteraciones y de casi 3 minutos en 1000 iteraciones. No obstante, se logró obtener un mejor resultado con el genético teniendo diferencias más cortas en ambas instancias de los algoritmos. La mejora en los resultados ya dependerá tanto del ajuste de los parámetros de cada algoritmo como la complejidad del problema, en este caso, el tamaño ruta del TSP.