TRABAJO FINAL - TÉCNICAS DE OPTIMIZACIÓN - TSP MEDIANTE PSO

Juan José Díaz Zuluaga - Oscar Andrés Gutiérrez Rivadeneira - Santiago Vargas Higuera

Universidad de Antioquia

***Index Terms*—** TSP, PSO, Metaheurísticas, Optimización, Gurobi

1. INTRODUCCIÓN

El problema del agente viajero (TSP) es uno de los problemas más estudiados dentro del campo de la optimización, ya que tiene infinidad de aplicaciones en situaciones de la vida real, como planes de rutas de distribución, aplicaciones de envíos e incluso diseño de circuitos impresos.

En el siguiente trabajo se aborda este problema mediante la metaheurística de Enjambre de Partículas (PSO por sus siglas en inglés), este modelo pertenece a un grupo de técnicas inspiradas por la naturaleza que en este caso se caracteriza por intentar imitar el comportamiento colectivo de sistemas biológicos como cardúmenes de peces o bandadas de aves. El uso del PSO permite explotar el espacio de soluciones considerando cada elemento del Enjambre como una solución factible para el sistema. Esta solución se desplaza basada en su posición y velocidad respecto a los otros componentes del grupo.

2. DEFINICIÓN DEL PROBLEMA

El Problema del Agente Viajero consiste en encontrar la ruta más corta que permite recorrer a un viajero un conjunto de ciudades partiendo en una de ellas y retornando a esta misma. Este problema fue postulado inicialmente en el siglo XIX por los matemáticos William Rowan Hamilton y Thomas Kirkman.

El TSP es ampliamente conocido debido a que es del tipo "NP-hard", lo que significa una alta complejidad computacional para instancias de gran tamaño.

2.1. Formulación Matemática

Dado un grafo G = (V, E) con distancias d_e, se busca encontrar el camino de menor costo que recorre todos los vértices del grafo y retorna al nodo de origen.

$$\text{F.O} \rightarrow \text{MIN} \sum_{e \in E} d_e x_e$$

Sujeto a:

$$\sum_{e \in \delta(i)} x_e = 2 \qquad \forall i \in V$$

$$\sum_{e \in \delta(U)} x_e \geq 2 \qquad \forall \phi \subset U \subset V$$

$$x_e \in \{0, 1\} \qquad \forall e \in E$$

Donde:

- $x_e = 1$ si e está en el tour
- $\delta(i)$ son los enlaces adyacentes a la ciudad
- V son todos los vertices del Grafo
- E son los enlaces del Grafo
- U es un conjunto de nodos cualquiera
- $\delta(U)$ son los enlaces que unen a U y V/U

3. MODELO DE OPTIMIZACIÓN

La metaheurística seleccionada para la solución del problema fue la optimización por Enjambre de Partículas (PSO). Este funciona inspirándose en el comportamiento colectivo de sistemas naturales, como bandadas de aves o bancos de peces, para encontrar soluciones óptimas a problemas complejos. En este contexto, las partículas representan posibles soluciones y se mueven a través del espacio de búsqueda influenciadas por dos factores principales:

- **Experiencia propia**: Cada partícula recuerda su mejor solución local encontrada (*pbest*).
- **Cooperación global**: Las partículas tienden a acercarse a la mejor solución global (*gbest*) hallada por el enjambre.

En cada iteración, las partículas ajustan sus posiciones según su historial y el del grupo, explorando diferentes soluciones mientras convergen hacia las más prometedoras. Para adaptar este modelo al Problema del Viajero (TSP), las posiciones de las partículas son rutas posibles entre ciudades, y las

actualizaciones de movimiento se realizan mediante operaciones de intercambio (*swap*). Estas operaciones modifican las rutas considerando probabilidades de influencia por *pbest* y *gbest*, permitiendo equilibrar la exploración y explotación. El objetivo final es minimizar el costo total de la ruta, generalmente la distancia recorrida.

4. DESCRIPCIÓN DE LA SOLUCIÓN

El enfoque de la solución implementada combina rutas generadas para minimizar la distancia total recorrida por un vendedor que debe visitar un conjunto de ciudades exactamente una vez.

Estructura

Clases Principales

1. **City**:
   - Representa una ciudad con coordenadas (x, y).
   - Permite calcular la distancia euclidiana entre ciudades.
2. **Particle**:
   - Representa una partícula con las siguientes características:
     - **route**: La ruta actual de la partícula.
     - **pbest**: La mejor solución local encontrada.
     - **velocity**: Secuencia de operaciones de intercambio (*swap*) entre ciudades.
   - Actualiza costos y mejora su ruta local mediante los operadores.
3. **PSO**:
   - Contiene las partículas y ciudades.
   - Calcula las mejores rutas locales (*pbest*) y globales (*gbest*) basándose en las probabilidades de intercambio.

Procedimiento del Algoritmo

1. **Inicialización**:
   - Generar rutas iniciales.
   - Crear partículas a partir de estas rutas.
2. **Iteraciones**:
   - Actualizar la mejor solución global (*gbest*).
   - Aplicar operadores de intercambio (*swap*) para ajustar rutas, considerando las probabilidades de los mejores locales y globales (*pbest* y *gbest*).
   - Actualizar las rutas y costos de cada partícula.
3. **Criterio de parada**: Detenerse después de un número máximo de iteraciones o cuando se alcance un umbral en la distancia.

Visualización

1. **Convergencia**: Se grafica la distancia global mínima alcanzada en función de las iteraciones.
2. **Mapas**: Se generan rutas optimizadas y mapas interactivos utilizando la librería *folium*.

5. ANÁLISIS DE RESULTADOS

Se implementó el algoritmo PSO (Optimización por Enjambre de Partículas) para resolver el problema del viajante (TSP) con diferentes cantidades de ciudades: 10, 15, 20, 30, 50, 60, 100* y 150*. Durante los experimentos, se mantuvieron constantes los parámetros de número máximo de iteraciones (2500) y tamaño de población inicial (1000). Las probabilidades asociadas a los coeficientes de aprendizaje, c_1 y c_2, se fijaron inicialmente en valores bajos (0.08 y 0.05, respectivamente) y se incrementaron de forma gradual con cada iteración.

Al analizar el impacto de los parámetros de probabilidad, se observó que un incremento en el valor inicial de c_1 fomenta que las partículas exploren en mayor medida el entorno cercano a sus mejores posiciones individuales (p_{best}), por lo tanto tiende a encontrar mínimos locales en menor cantidad de iteraciones, pero con el tiempo deja de encontrar mejores soluciones y permanece en la misma solución, reduciendo la posibilidad de diversificar. Este fenómeno se evidencia aún más cuando se incrementan el número de ciudades. Por otra parte, al aumentar el valor inicial de c_2 el enjambre encuentra un mínimo global en menos iteraciones, pero impide que cada partícula siga explorando mejores soluciones en su recorrido.

Numero de Ciudades	Best PSO	Optima	Error [%]
10	104.81	104.81	0
15	109.41	109.4	0.01
20	126.89	122.78	3.35
30	154.45	141,00	9.54
50	201.1	173.13	16.16
60	213.82	176.88	20.88
100	264.79	-	-
150	295.48	-	-

Table 1: Comparación de mejor solución PSO vs Solución óptima

Al observar los resultados, en primer lugar se identifica que al aumentar la cantidad de ciudades, será más difícil para

el algoritmo encontrar la solución óptima ya que aumenta su complejidad.

Número de Ciudades	Costo Rutas Aleatorias	Costo Ruta ciudades más cercanas	Optima
10	104.81	104.81	104.81
15	109.41	109.50	109.4
20	126.89	127.05	122.78
30	154.45	165.48	141.00
50	308.02	201.10	173.13
60	414.61	213.82	176.88
100	784.78	264.79	-
150	540.54	295.48	-

Table 2: Comparación del Costo final con rutas iniciales aleatorias vs ruta inicial de ciudades más cercanas

En la generación de la población inicial, es posible optar por dos enfoques: crear todas las rutas de forma completamente aleatoria o incluir una ruta inicial generada mediante una heurística que priorice las ciudades más cercanas entre sí. Este último enfoque busca forzar la identificación de un mínimo local, acelerando el proceso de búsqueda de soluciones óptimas.

Los resultados muestran que, para instancias con menos de 50 ciudades, las rutas generadas aleatoriamente tienden a ofrecer un rendimiento ligeramente superior. Sin embargo, cuando se trabaja con una mayor cantidad de ciudades, incluir una ruta inicial basada en la proximidad entre ciudades mejora significativamente los resultados. Este comportamiento sugiere que, en problemas más grandes, las soluciones heurísticas iniciales contribuyen a guiar el algoritmo hacia regiones más prometedoras del espacio de búsqueda.

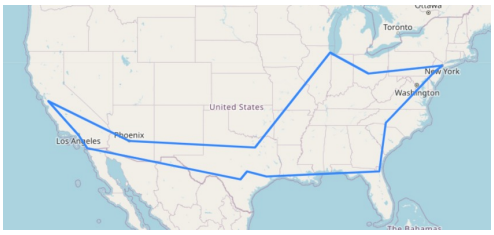

Fig. 1: Solución con Gurobi para 15 ciudades.

Al analizar las diferencias entre el óptimo determinado mediante Gurobi y la solución generada por PSO para la mayoría de los casos, se observa una notable similitud en el trazado de las rutas. Sin embargo, el recorrido obtenido con PSO presenta algunas desviaciones respecto al óptimo, que pueden atribuirse a la naturaleza heurística del algoritmo.

La solución generada por Gurobi, al ser basada en métodos exactos, sigue un trayecto más directo y eficiente. Por otro lado, el PSO ofrece una alternativa más flexible y eficiente para escenarios donde la cantidad de ciudades es mucho mayor, a pesar de su tendencia a converger hacia soluciones ligeramente menos óptimas.

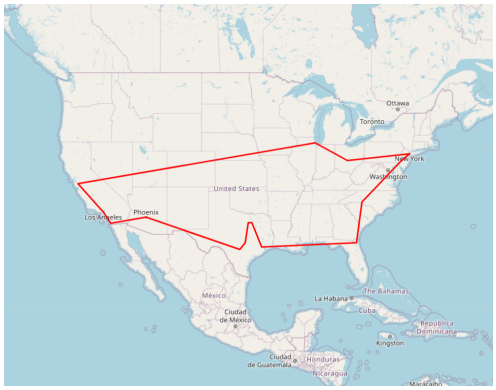

Fig. 2: Solución con PSO para 15 ciudades.

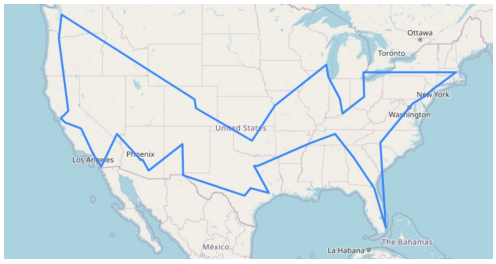

Fig. 3: Solución con Gurobi para 40 ciudades.

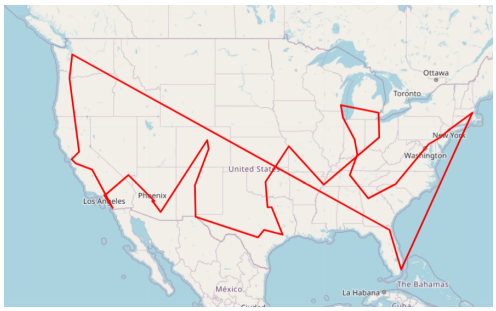

Fig. 4: Solución con PSO para 40 ciudades.

Se puede apreciar que la solución de PSO es más dispersa en ciertos tramos del recorrido, mientras que Gurobi optimiza la distancia total al priorizar un trayecto más ordenado.

6. BIBLIOGRAFÍA

[1] Elizabeth F. G. Goldbarg, Marco C. Goldbarg and Givanaldo R. de Souza (2008). Particle Swarm Optimization Algorithm for the Traveling Salesman Problem, Traveling Salesman Problem, Federico Greco

[2] Noshy. R (2020). Traveling Salesman Optimization github.com/rameziophobia/Travelling_Salesman_Optimization

[3] Castro de Souza M. (2015). TSP_PSO
https://github.com/marcoscastro/tsp_pso