

Proyecto 3: Vuelta al mundo en 20 ciudades:

SPA-MST-TSP-SA

Miguel Ochoa Hernández, Cesar Fernando Laguna Ambriz, Eduardo Armando Villarreal García
Maestría: Ciencias Computacionales
Universidad Autónoma de Guadalajara

Resumen. En el siguiente documento se pretende exponer el caso para resolución del problema del recorrido de 20 ciudades para encontrar la ruta más corta (SPA - Shortest Path Algorithm), la conexión mínima entre todas las ciudades (MST- Minimum Spanning Tree), y resolver el problema del agente viajero (TSP - Travel Salesman Problem).

Se describen los conceptos de SPA, MST y TSP en el desarrollo se muestran los algoritmos en desarrollo y el análisis de los grafos como resultados obtenidos

I. INTRODUCCIÓN

En la teoría de grafos, el problema del camino más corto es el problema que consiste en encontrar un camino entre dos vértices (o nodos) de tal manera que la suma de los pesos de las aristas que lo constituyen es mínima. Un ejemplo de esto es encontrar el camino más rápido para ir de una ciudad a otra en un mapa. En este caso, los vértices representarían las ciudades y las aristas las carreteras que las unen, cuya ponderación viene dada por el tiempo que se emplea en atravesarlas.

A. Los algoritmos más importantes para resolver este problema son:

- Algoritmo de Dijkstra, resuelve el problema de los caminos más cortos desde un único vértice origen hasta todos los otros vértices del grafo.
- Algoritmo de Bellman - Ford, resuelve el problema de los caminos más cortos desde un origen si la ponderación de las aristas es negativa.
- Algoritmo de Búsqueda A*, resuelve el problema de los caminos más cortos entre un par de vértices usando la heurística para intentar agilizar la búsqueda.
- Algoritmo de Floyd - Warshall, resuelve el problema de los caminos más cortos entre todos los vértices.
- Algoritmo de Johnson, resuelve el problema de los caminos más cortos entre todos los vértices y puede ser más rápido que el de Floyd-Warshall en grafos de baja densidad.
- Algoritmo de Viterbi, resuelve el problema del camino estocástico más corto con un peso probabilístico adicional en cada vértice.

II. PROPUESTA DE SOLUCIÓN

Para obtener la ruta más corta se eligió el algoritmo de Dijkstra, también llamado algoritmo de caminos mínimos (Shortest Path Algorithm). Es un algoritmo para la determinación del camino más corto dado un vértice origen al resto de los vértices en un grafo con pesos en cada arista. Su nombre se refiere a Edsger Dijkstra, quien lo describió por primera vez en 1959. La idea subyacente en este algoritmo consiste en ir explorando todos los caminos más cortos que parten del vértice origen y que llevan a todos los demás vértices; cuando se obtiene el camino más corto desde el vértice origen, al resto de vértices que componen el grafo, el algoritmo se detiene. El algoritmo es una especialización de la búsqueda de costo uniforme, y como tal, no funciona en grafos con aristas de costo negativo (al elegir siempre el nodo con distancia menor, pueden quedar excluidos de la búsqueda nodos que en próximas iteraciones bajaría el costo general del camino al pasar por una arista con costo negativo).

Orden de complejidad del algoritmo:

$$O(|V|^2 + |A|) = O(|V|^2).$$

Para calcular el MST se implementó el algoritmo de Kruskal. El algoritmo de Kruskal es un algoritmo de árbol de expansión mínima que encuentra una ventaja del menor peso posible que conecta dos árboles cualesquiera en el bosque. Es un algoritmo codicioso en la teoría de grafos, ya que encuentra un árbol de expansión mínimo para

un gráfico ponderado conectado que agrega incrementos en los costos de arcos en cada paso.

Pasos para determinar MST usando Algoritmo de Kruskal:

- Clasifique todos los bordes en orden decreciente de su peso.
- Elija el borde más pequeño. Compruebe si forma un ciclo con el árbol de expansión formado hasta el momento. Si el ciclo no está formado, incluya este borde. De lo contrario, descartarlo.
- 3. Repita el paso no. 2 hasta que haya $(V-1)$ en el árbol de expansión.

Puede mostrarse que el algoritmo de Kruskal se ejecuta en el tiempo $O(E \log E)$, o equivalentemente, el tiempo $O(E \log V)$, donde E es el número de aristas en el gráfico y V es el número de vértices, todos con estructuras de datos simples.

El algoritmo del vecino más próximo fue, en las ciencias de la computación, uno de los primeros algoritmos utilizados para determinar una solución para el problema del viajante.

Este método genera rápidamente un camino corto, pero generalmente no el ideal. Abajo está la aplicación del algoritmo del vecino más próximo al problema del viajante.

Estos son los pasos del algoritmo:

- Elección de un vértice arbitrario respecto al vértice actual.
- Descubra la arista de menor peso que ya esté conectada al vértice actual y a un vértice no visitado V .
- Convierta el vértice actual en V .
- Marque V como visitado.
- 5. Si todos los vértices del dominio estuvieran visitados, cierre el algoritmo.
- 6. Vaya al paso 2.

La secuencia de los vértices visitados es la salida del

Algoritmo. El algoritmo del vecino más próximo es fácil de implementar y ejecutar rápidamente, pero algunas veces puede perder rutas más cortas, que son fácilmente notadas con la visión humana, debido a su naturaleza más "ávida".

Como norma general, si los últimos pasos del recorrido son comparables en longitud al de los primeros pasos, el recorrido es razonable; si estos son mucho mayores, entonces es probable que

existan caminos mucho mejores. Tiempo de complejidad:

$O(N^2)$.

Otro algoritmo que se implementó para resolver TSP fue el

Algoritmo Simulated annealing (SA).

Como ya se mencionó anteriormente, la descripción del algoritmo es la siguiente:

- Primero, debemos establecer la temperatura inicial y crear una solución inicial aleatoria.
- Luego comenzamos a repetir hasta que se cumpla nuestra condición de detención. Por lo general, o bien el sistema se ha enfriado lo suficiente o se ha encontrado una solución lo suficientemente buena.
- Desde aquí seleccionamos un vecino haciendo un pequeño cambio en nuestra solución actual.
- Luego decidimos si pasar a la solución vecina.
- Finalmente, disminuimos la temperatura y continuamos.

III. RESULTADOS

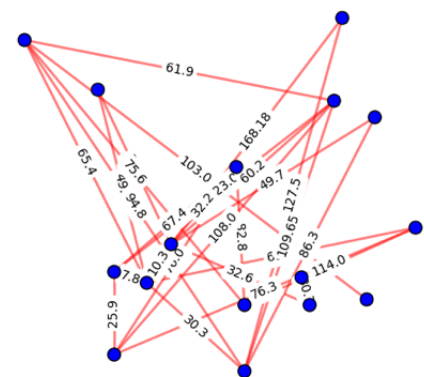


Figura 1. Ruta más corta Tepoztlán - Xochiltepec

En esta sección se muestran algunos datos sobre la ejecución del programa.

Para la implementación de los algoritmos SPA y MST se ingresaron 20 ciudades con una distancia definida entre ellas.

En la figura 1 se muestra el resultado de la ruta más corta obtenida para realizar un viaje desde Guadalajara hasta Puerto Vallarta.

La ruta obtenida fue Tepoztlán - Xochiltepec con una distancia de 114.00 km.

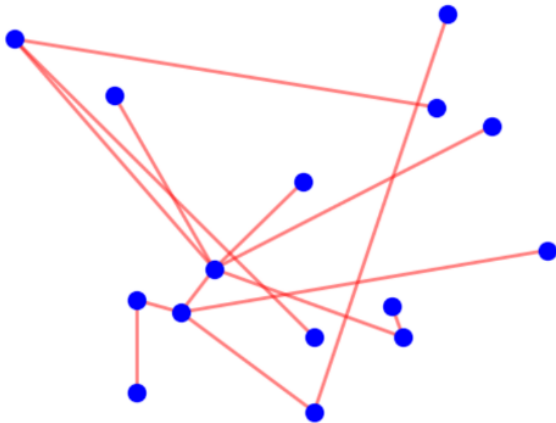


Figura 2. MST obtenido de las 20 ciudades

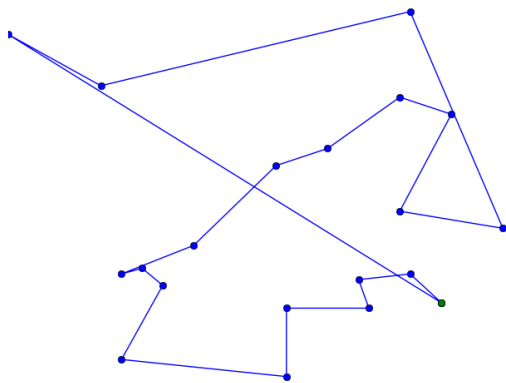
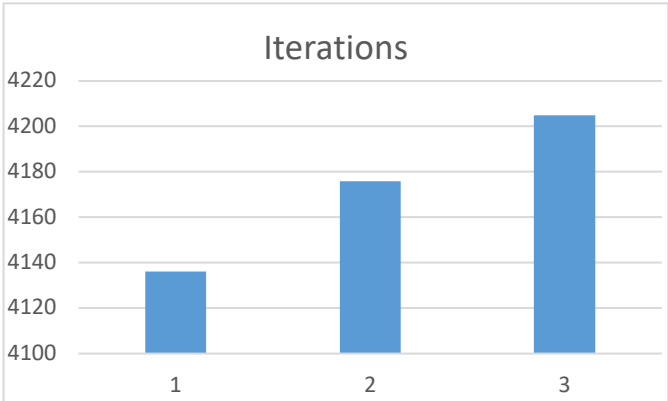


Figura 3. TSP obtenido partiendo de la ciudad de Cuernavaca visitando todas las ciudades y volviendo al punto de origen.

	20 ciudades	100 Ciudades	150 Ciudades	200 Ciudades
SPA	114	0	0	0
MST	800	0	0	0
SA	328	25051	30051	35912

Tabla 1: Tabla comparativa de los tiempos que se procesaron (las cantidades esta en KM.)

	100 ciudades	150 ciudades	200 ciudades
Iterations	4136	4176	4205



Grafica 1: Grafica comparativa de los tiempos que se procesaron (las cantidades e sta en KM.)

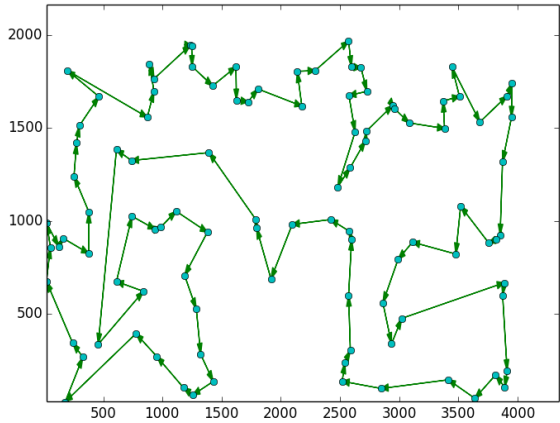


Figura 3: Solución TSP con 100 ciudades con el algoritmo Simulated Annealing

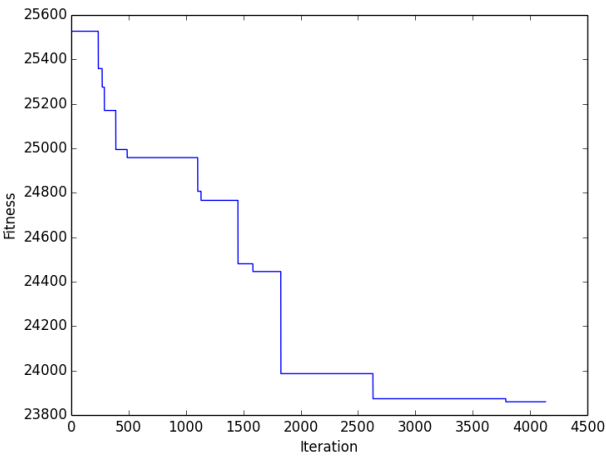


Figura 4: Solución TSP con 100 ciudades con el algoritmo Simulated Annealing

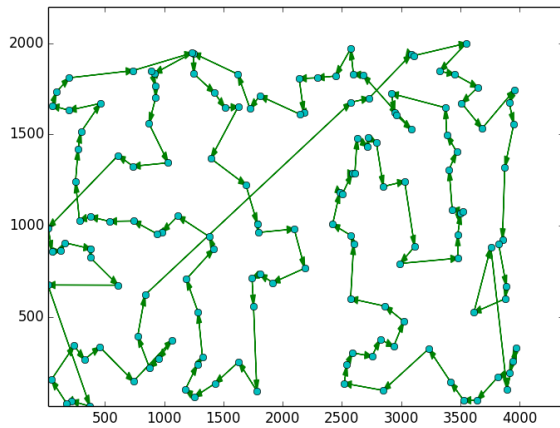


Figura 5: Solución TSP con 150 ciudades con el algoritmo Simulated Annealing

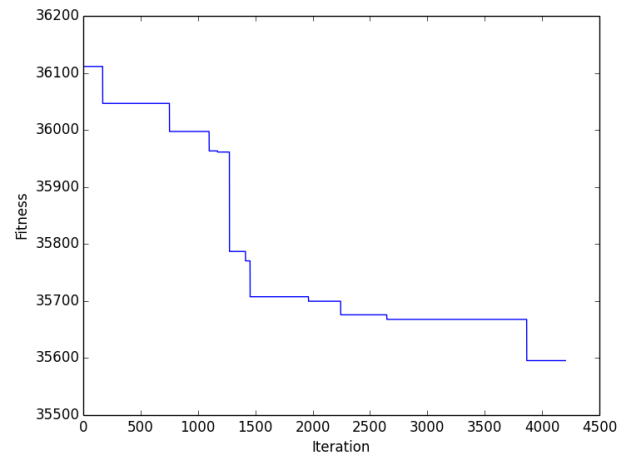


Figura 8: Solución TSP con 200 ciudades con el algoritmo Simulated Annealing

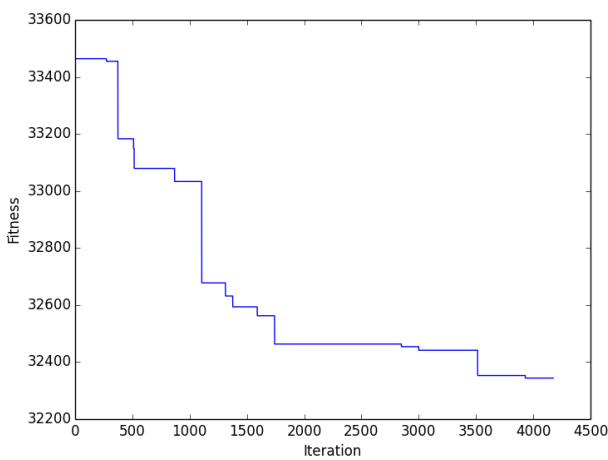


Figura 6: Solución TSP con 150 ciudades con el algoritmo Simulated Annealing

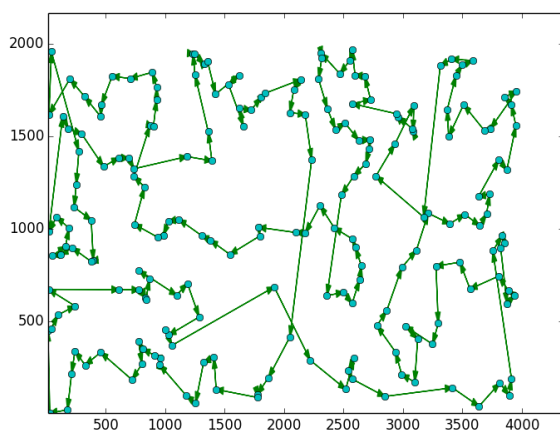


Figura 7: Solución TSP con 200 ciudades con el algoritmo Simulated Annealing

IV. CONCLUSIONES

A medida que nos damos cuenta de que se crean nodos y se implementan estos algoritmos nos damos cuenta del grado de su complejidad.

REFERENCIAS

Applegate, D., Bixby, R., Chvatal V., Cook, W. "On the solution of the Traveling Salesman Problem". Documenta Mathematica-Extra Volume ICM III. 1998. 645-656.

Brest, J., Zerovnik, J. 'A heuristic for the Asymmetric Traveling Salesman Problem'. The 6th Metaheuristics International Conference. 2005. 145-150.