



Implementación y Evaluación del Algoritmo de Dijkstra.

Docente: M. en C. Erika Sánchez-Femat

Autor: Roberto Benitez Zamora

Fecha: 1 de Diciembre del 2023

1 Introducción

El algoritmo de Dijkstra, desarrollado por el científico holandés Edsger W. Dijkstra en 1956, es una técnica de búsqueda de caminos mínimos en grafos ponderados no dirigidos o dirigidos. Este algoritmo encuentra el camino más corto desde un nodo fuente (o nodo origen) a todos los demás nodos en un grafo, siempre y cuando los pesos de las aristas no sean negativos.

El algoritmo mantiene una lista de nodos visitados y no visitados, calculando continuamente las distancias mínimas desde el nodo origen a los demás nodos. A través de iteraciones, va expandiendo el conjunto de nodos visitados y actualizando las distancias mínimas, asegurándose de tomar siempre la ruta más corta disponible hasta cada nodo.

2 Desarrollo

Clase Grafo:

- Constructor `__init__`: Inicializa el grafo como un diccionario vacío.
- Método `agregar_vertice`: Agrega un vértice al grafo si no existe previamente.
- Método `agregar_arista`: Agrega una arista entre dos vértices con un peso asociado.
- Método `dijkstra`:
 - Inicializa un diccionario `distancias` con todas las distancias desde el nodo inicial como infinito, excepto el nodo de inicio que tiene distancia 0.
 - Crea una cola de prioridad (`cola_prioridad`) utilizando una lista para almacenar tuplas de la forma (`distancia`, `nodo`).
 - Mientras la cola de prioridad no esté vacía:
 - * Extrae el nodo con la menor distancia actual (`nodo_actual`) de la cola de prioridad.
 - * Para cada vecino del `nodo_actual`, actualiza la distancia si la nueva distancia desde el inicio es menor a la almacenada. Actualiza la cola de prioridad con las nuevas distancias.

- * Devuelve un diccionario con los caminos mínimos desde el nodo de inicio hasta cada nodo del grafo.

Función `imprimir_caminos_minimos_con_tiempo`:

- Toma el tiempo inicial.
- Ejecuta el algoritmo de Dijkstra para encontrar los caminos mínimos desde el vértice de inicio.
- Toma el tiempo final.
- Imprime los caminos mínimos y el tiempo de ejecución.

El algoritmo en sí utiliza la estructura de cola de prioridad para mantener un seguimiento eficiente de los nodos con las distancias mínimas conocidas hasta el momento.

3 Resultados

3.1 Casos de Ejecución del Algoritmo

4 Conclusiones

El algoritmo de Dijkstra es una herramienta fundamental en el campo de la teoría de grafos y la optimización de rutas. Su eficiencia para encontrar los caminos mínimos en grafos con pesos no negativos lo hace ampliamente utilizado en aplicaciones como redes de transporte, telecomunicaciones y sistemas de navegación.

Este algoritmo, aunque poderoso, tiene una complejidad temporal de $O(V^2)$ para implementaciones simples y puede mejorarse a $O((V + E) \log V)$ utilizando una cola de prioridad, donde V es el número de vértices y E es el número de aristas en el grafo.

5 Referencias

Artículos Académicos

1. Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1), 269-271.
2. Fredman, M. L., & Tarjan, R. E. (1984). Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM (JACM)*, 31(4), 538-554.

Recursos en Línea

1. GeeksforGeeks: *Dijkstra's shortest path algorithm*. <https://www.geeksforgeeks.org/dijkstras-shortest-path-algorithm-greedy-algo-7/>
2. Khan Academy: *Dijkstra's algorithm*. <https://www.khanacademy.org/computing/computer-science/algorithms/graph-representation/a/dijkstras-algorithm>
3. Wikipedia: *Dijkstra's algorithm*. <https://en.wikipedia.org/wiki/Dijkstra>