Instituto Tecnológico de Culiacán

Ingeniería en Sistemas Computacionales

Materia:

Inteligencia Artificial.

Tarea:

Algoritmo de Dijkstra.

Alumno:

Flores Saldaña Martin Alejandro

Docente:

Jose Mario Rios Felix

Grupo:

18:00-19:00

4 de septiembre de 2025

Índice

¿Qué es el algoritmo de Dijkstra?

El **algoritmo de Dijkstra**, también llamado **algoritmo de caminos mínimos** es un algoritmo para la determinación del camino más corto, dado un vértice origen, hacia el resto de los vértices en un grafo que tiene pesos en cada arista. Su nombre alude a Edsger Dijkstra, científico de la computación de los Países Bajos que lo concibió en 1956 y lo publicó 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 hasta el resto de los vértices que componen el grafo, el algoritmo se detiene. Se trata de una especialización de la búsqueda de costo uniforme y, como tal, no funciona en grafos con aristas de coste negativo (al elegir siempre el nodo con distancia menor, pueden quedar excluidos de la búsqueda nodos que en próximas iteraciones bajarían el costo general del camino al pasar por una arista con costo negativo)

¿Cómo funciona?

Teniendo un grafo dirigido ponderado de N nodos no aislados, sea x el nodo inicial. Un vector D de tamaño N guardará al final del algoritmo las distancias desde x hasta el resto de los nodos.

1. Inicializar todas las distancias en D con un valor infinito relativo, ya que son desconocidas al principio, exceptuando la de x que se debe colocar en 0 debido a que la distancia de x a x sería 0.
2. Sea $a = x$ (Se toma a como nodo actual).
3. Se recorren todos los nodos adyacentes de a, excepto los nodos marcados. Se les llamará nodos no marcados v_i.
4. Para el nodo actual, se calcula la distancia tentativa desde dicho nodo hasta sus vecinos con la siguiente fórmula: $dt(v_i) = D_a + d(a, v_i)$. Es decir, la distancia tentativa del nodo 'v_i' es la distancia que actualmente tiene el nodo en el vector D más la distancia desde dicho nodo 'a' (el actual) hasta el nodo v_i. Si la distancia tentativa es menor que la distancia almacenada en el vector, entonces

se actualiza el vector con esta distancia tentativa. Es decir, si $dt(v_i) < D_{v_i} \rightarrow D_{v_i} = dt(v_i)$

5. Se marca como completo el nodo a.
6. Se toma como próximo nodo actual el de menor valor en D (puede hacerse almacenando los valores en una cola de prioridad) y se regresa al paso 3, mientras existan nodos no marcados.

PSEUDOCÓDIGO

```
DIJKSTRA (Grafo G, nodo_fuente s)
    para u ∈ V[G] hacer
        distancia[u] = INFINITO
        padre[u] = NULL
        visto[u] = false
    distancia[s] = 0
    adicionar (cola, (s, distancia[s]))
    mientras que cola no es vacía hacer
        u = extraer_mínimo(cola)
        visto[u] = true
        para todos v ∈ adyacencia[u] hacer
            si ¬ visto[v]
                si distancia[v] > distancia[u] + peso (u, v) hacer
                    distancia[v] = distancia[u] + peso (u, v)
                    padre[v] = u
                    adicionar(cola,(v, distancia[v]))
```

¿Cuál es su complejidad algorítmica?

O(|V|²+|A|) = O(|V|²), sin utilizar cola de prioridad, :O((|A|+|V|) log |V|) = O(|A| log |V|) utilizando cola de prioridad (por ejemplo, un montículo binario o un árbol binario balanceado). Por otro lado, si se utiliza un montículo de Fibonacci, sería O(|V| log |V|+|A|)

Bibliografía

1. https://es.wikipedia.org/wiki/Algoritmo_de_Dijkstra