Shortest Paths

MSc Edson Ticona Zegarra

Taller avanzado 2025

Contenido

Caminos mínimos

- Dado un grafo con pesos, se busca minimizar la suma de los pesos de las aristas del camino entre s y t

Caminos mínimos

- Dado un grafo con pesos, se busca minimizar la suma de los pesos de las aristas del camino entre s y t
- Formalmente,

$$\min \sum_{e \in P} w(e)$$

Caminos mínimos

- Dado un grafo con pesos, se busca minimizar la suma de los pesos de las aristas del camino entre s y t
- Formalmente,

$$\min \sum_{e \in P} w(e)$$

- Tal que P es un camino entre s y t

Relajación

- La idea es verificar si existe una mejor opción para un camino mínimo, de ser así se utiliza tal opción

Relajación

- La idea es verificar si existe una mejor opción para un camino mínimo, de ser así se utiliza tal opción
- Sea $d(u)$ la distancia hasta el vértice u y sea $w(u, v)$ el peso de la arista que conecta u y v

Relajación

- La idea es verificar si existe una mejor opción para un camino mínimo, de ser así se utiliza tal opción
- Sea $d(u)$ la distancia hasta el vértice u y sea $w(u, v)$ el peso de la arista que conecta u y v
- Si $d(u) + w(u, v) < d(v)$ entonces el camino mínimo hasta v puede ser mejorado usando el camino mínimo hasta u junto con la arista que conecta u y v

Relajación

input : d vector de distancias, *parent* vector de caminos
if $d(u) + w(u, v) < d(v)$ **then**
    $d(v) \leftarrow d(u) + w(u, v)$
    $parent(v) \leftarrow u$
end

Algoritmo de Bellman-Ford

- Este algoritmo resuelve el caso de una fuente única al resto de los vértices del grafo (Single Source Shortest Path SSSP)

Algoritmo de Bellman-Ford

- Este algoritmo resuelve el caso de una fuente única al resto de los vértices del grafo (Single Source Shortest Path SSSP)
- Inicialmente el vector de distancia toma valores de infinito

Algoritmo de Bellman-Ford

- Este algoritmo resuelve el caso de una fuente única al resto de los vértices del grafo (Single Source Shortest Path SSSP)
- Inicialmente el vector de distancia toma valores de infinito
- La idea es relajar todas las $|E|$ aristas $|V| - 1$ veces

Algoritmo de Bellman-Ford

- Este algoritmo resuelve el caso de una fuente única al resto de los vértices del grafo (Single Source Shortest Path SSSP)
- Inicialmente el vector de distancia toma valores de infinito
- La idea es relajar todas las $|E|$ aristas $|V| - 1$ veces
- Pueden existir pesos negativos

Algoritmo de Bellman-Ford

```
input  : G grafo con pesos y s vértices inicial
output: true si no hay ciclos negativos
Init(d, s) ;    /* distancia 0 para s e ∞ al resto */
for v ∈ V do
    for e ∈ E do
        Relax(u, v, w)
    end
end
for e ∈ E do
    if d(v) > d(u) + w(u, v) then
        return false
    end
end
return true
```

Algoritmo de Bellman-Ford

- Complejidad: $O(EV)$

Algoritmo de Dijkstra

- Requiere pesos no negativos

Algoritmo de Dijkstra

- Requiere pesos no negativos
- Mejor complejidad que Berllman-Ford

Algoritmo de Dijkstra

- Requiere pesos no negativos
- Mejor complejidad que Berllman-Ford
- Se mantiene una cola de priodad en función a la distancia

Algoritmo de Dijkstra

- Requiere pesos no negativos
- Mejor complejidad que Berllman-Ford
- Se mantiene una cola de priodad en función a la distancia
- Se va relajando los vértices conforme se va avanzando en la cola de prioridades

Algoritmo de Dijkstra

```
input  : G grafo con pesos
output: d con todas las distancias
INIT(d, s) ;      /* distancia 0 para s e ∞ al resto */
Q.push(v ∈ V) ;       /* en funcion de la distancia */
while !Q.empty() do
    u ← Q.top()
    for v ∈ G(u) do
        RELAX(u, v, w)
    end
end
```

Algoritmo de Dijkstra

- Complejidad: $O(E + V \log V)$

Algoritm de Floyd-Warshall

All Pairs Shortest Paths

- Estos algoritmo encuentan las distancias mínimas entre todos los pares de vértices

All Pairs Shortest Paths

- Estos algoritmo encuentan las distancias mínimas entre todos los pares de vértices
- A diferencia de los anterior que encuentran la distancia mínima entre un vértice origen s y el resto de vértices

All Pairs Shortest Paths

- Estos algoritmo encuentan las distancias mínimas entre todos los pares de vértices
- A diferencia de los anterior que encuentran la distancia mínima entre un vértice origen s y el resto de vértices
- Este problema es una generalización y los algoritmos son más lentos

All Pairs Shortest Paths

- Estos algoritmo encuentan las distancias mínimas entre todos los pares de vértices
- A diferencia de los anterior que encuentran la distancia mínima entre un vértice origen s y el resto de vértices
- Este problema es una generalización y los algoritmos son más lentos
- La solución suele ser una matriz cuadrada π de $n \times n$, tal que π_{uv} denota la distancia mínima entre el vértice u y el vértice v

Algoritm de Floyd-Warshall

- El algoritmo de Floyd-Warshall considera pesos negativos pero no ciclos negativos

Algoritm de Floyd-Warshall

- El algoritmo de Floyd-Warshall considera pesos negativos pero no ciclos negativos
- Algoritmo de programación dinámica

Algoritmo de Bellman-Ford

input : G grafo con pesos y s vértices inicial
output: di matriz de distancias

```
for k ∈ |V| do
    for u ∈ |V| do
        for v ∈ |V| do
            d[u][v] = min(d[u][v], d[u][k] + d[k][v])
        end
    end
end
```

Algoritm de Floyd-Warshall

- Complejidad: $O(V^3)$

Comparación

Algoritmo	Caso	Observación	Complejidad
BFS	SSSP	sin pesos	$O(V + E)$
Bellman-Ford	SSSP	pesos negativos	$O(VE)$
Dijkstra	SSSP	pesos positivos	$O((V + E) \log V)$
Floyd-Warshall	APSP	Para grafos pequeños	$O(V^3)$