

Algoritmo de Bellman-Ford para Calcular la Ruta Más Corta en un Grafo

Juan Mendivelso Moreno, Assistant Professor

Universidad Nacional de Colombia
School of Science
Department of Mathematics

Contenido

- 1 Preliminares
 - Grafos
 - Rutas
- 2 Problema de la Ruta Más Corta
 - Problema
 - Aplicaciones
- 3 Solución al Problema de la Ruta Más Corta
 - Aspectos Importantes y Representación
 - Propiedades de Rutas Más Cortas y Relajación
 - Algoritmo de Bellman-Ford

Contenido

1 Preliminares

- Grafos
- Rutas

2 Problema de la Ruta Más Corta

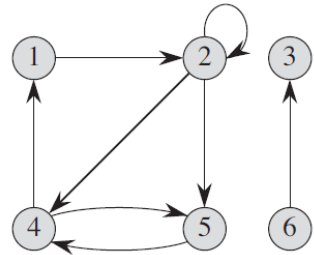
- Problema
- Aplicaciones

3 Solución al Problema de la Ruta Más Corta

- Aspectos Importantes y Representación
- Propiedades de Rutas Más Cortas y Relajación
- Algoritmo de Bellman-Ford

Grafo Dirigido

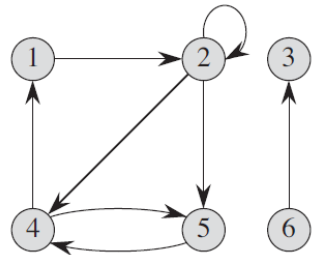
Un grafo dirigido G es un par de conjuntos (V, E) , donde V es un conjunto finito y E es una relación binaria sobre V . Los elementos de V se denominan *vértices* y los elementos de E se denominan *arcos*.



Grafo Dirigido

Example 1.

- $V = \{1, 2, 3, 4, 5, 6\}$
- $E = \{(1, 2), (2, 2), (2, 4), (2, 5), (4, 1), (4, 5), (5, 4), (6, 3)\}$

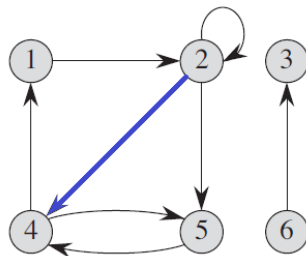


Adyacencia

Si el arco $(u, v) \in E$, se dice que el vértice v es **adyacente** al vértice u . Esto se denota como $u \rightarrow v$.

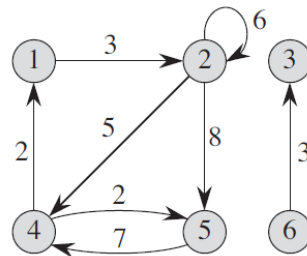
Example 2.

El vértice 4 es adyacente al vértice 2.



Grafo con Pesos

Un grafo dirigido $G = (V, E)$ tiene asociados **pesos** si existe una función de pesos $\omega : E \rightarrow \mathbb{R}$ que asocia un peso $\omega(u, v)$ a cada arco $(u, v) \in E$.



Contenido

1 Preliminares

- Grafos
- Rutas

2 Problema de la Ruta Más Corta

- Problema
- Aplicaciones

3 Solución al Problema de la Ruta Más Corta

- Aspectos Importantes y Representación
- Propiedades de Rutas Más Cortas y Relajación
- Algoritmo de Bellman-Ford

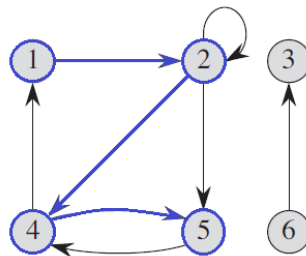
Ruta

Una **ruta** de tamaño k de un vértice u a un vértice u' en el grafo $G = (V, E)$ es una secuencia de vértices $\langle v_0, v_1, v_2, \dots, v_k \rangle$ tal que $u = v_0$ y $u' = v_k$ y $(v_{i-1}, v_i) \in E$ for $i \in \{1, 2, \dots, k\}$.

Tamaño de la ruta: Número de arcos.

Example 3.

La ruta resaltada es $\langle 1, 2, 4, 5 \rangle$.

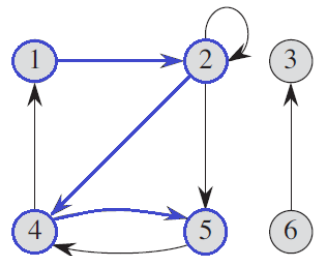


Vértices Alcanzables

Si existe una ruta p desde el vértice u hasta el vértice u' , se dice que u' es **alcanzable** desde u a través de p . Esto se denota como $u \xrightarrow{p} u'$.

Example 4.

El vértice 5 es alcanzable desde el vértice 1.

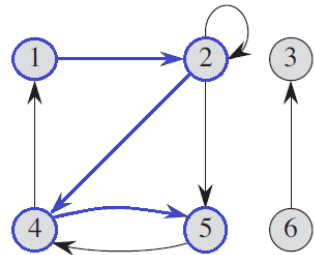


Ruta Simple

Una ruta es **simple** si todos los vértices que contiene son distintos.

Example 5.

La ruta resaltada resaltado es simple.

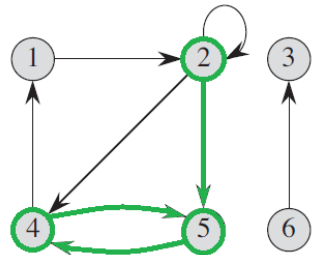


Ruta Simple

Una ruta es **simple** si todos los vértices que contiene son distintos.

Example 6.

La ruta resaltada no es simple.

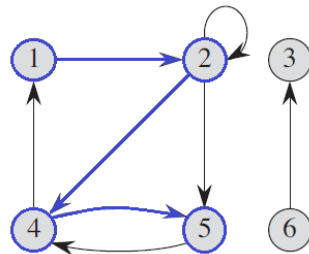


Sub-Ruta

Dada la ruta $p = \langle v_0, v_1, \dots, v_k \rangle$, la subsecuencia $\langle v_i, v_{i+1}, \dots, v_j \rangle$ es una **sub-ruta** de p donde $0 \leq i \leq j \leq k$.

Example 7.

Un ejemplo de subruta de la ruta resaltada es $\langle 2, 4, 5 \rangle$.

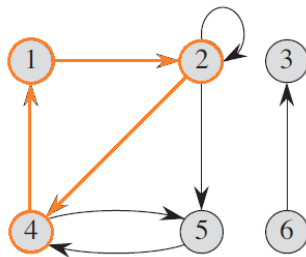


Ciclo

- En un grafo dirigido, la ruta $p = \langle v_0, v_1, \dots, v_k \rangle$ es un **ciclo** si $v_0 = v_k$ y al menos contiene un arco.
- Un ciclo es **simple** si v_1, v_2, \dots, v_k son distintos.

Example 8.

La ruta resaltada es un ciclo simple.



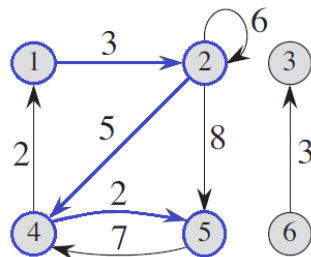
Peso de una Ruta

Dado un grafo dirigido con pesos $G = (V, E)$, cuya función de pesos es $\omega : E \rightarrow \mathbb{R}$, el **peso de la ruta** $p = \langle v_0, v_1, \dots, v_k \rangle$ se define como:

$$\omega(p) = \sum_{i=1}^k \omega(v_{i-1}, v_i) \quad (1)$$

Example 9.

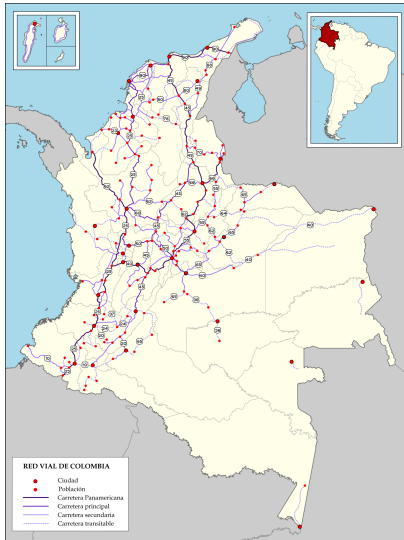
El peso de la ruta resaltada es 10.



Contenido

- 1 Preliminares
 - Grafos
 - Rutas
- 2 Problema de la Ruta Más Corta
 - Problema
 - Aplicaciones
- 3 Solución al Problema de la Ruta Más Corta
 - Aspectos Importantes y Representación
 - Propiedades de Rutas Más Cortas y Relajación
 - Algoritmo de Bellman-Ford

Importancia



¿Cuál es la ruta más corta entre Villavicencio y Cali?

Ruta Más Corta

El **peso de la ruta más corta** desde u hasta v se define como:

$$\delta(u, v) = \begin{cases} \text{mín}\{\omega(p) : u \rightsquigarrow_p v\} & \text{si } v \text{ es alcanzable desde } u \\ \infty & \text{en caso contrario.} \end{cases} \quad (2)$$

La **ruta más corta** desde el vértice u hasta el vértice v se define como cualquier ruta p con peso $\omega(p) = \delta(u, v)$.

Variantes

Problema de la Ruta Más Corta - Pareja Única

- **Entrada:** Grafo $G = (V, E)$ con función de pesos $\omega : E \rightarrow \mathbb{R}$, vértices $u, v \in V$.
- **Salida:** Ruta más corta entre u y v .

Problema de la Ruta Más Corta - Fuente Única

- **Entrada:** Grafo $G = (V, E)$ con función de pesos $\omega : E \rightarrow \mathbb{R}$, vértice fuente $s \in V$.
- **Salida:** Ruta más corta desde s hasta cada uno de los vértices $v \in V$.

Variantes

Problema de la Ruta Más Corta - Destino Único

- **Entrada:** Grafo $G = (V, E)$ con función de pesos $\omega : E \rightarrow \mathbb{R}$, vértice destino $t \in V$.
- **Salida:** Ruta más corta desde cada vértice $v \in V$ hasta t .

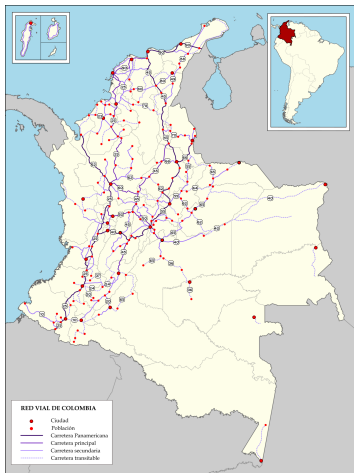
Problema de la Ruta Más Corta - Todas las Parejas

- **Entrada:** Grafo $G = (V, E)$ con función de pesos $\omega : E \rightarrow \mathbb{R}$.
- **Salida:** Ruta más corta desde u hasta v para todo $u, v \in V$.

Contenido

- 1 Preliminares
 - Grafos
 - Rutas
- 2 Problema de la Ruta Más Corta
 - Problema
 - Aplicaciones
- 3 Solución al Problema de la Ruta Más Corta
 - Aspectos Importantes y Representación
 - Propiedades de Rutas Más Cortas y Relajación
 - Algoritmo de Bellman-Ford

Aplicaciones



¿Cuál es la ruta más corta entre Villavicencio y Cali?

Calcular la ruta más corta en un grafo $G = (V, E)$ con pesos donde:

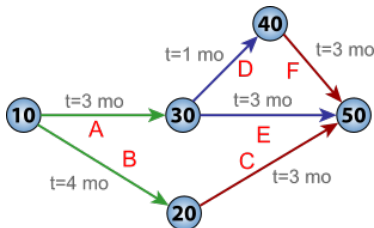
- V es el conjunto de intersecciones de vías.
- E es el conjunto de segmentos de vía entre intersecciones.
- La función de peso $\omega : E \rightarrow \mathbb{R}$ asigna a cada segmento de vía (arco) su longitud como peso.

Aplicaciones

El cálculo de la ruta más corta permite solucionar problemas diversos asociados a grafos cuyos pesos representan **tiempo**, **costo**, **penalizaciones** u otras cantidades que se puedan acumular a lo largo de una ruta y que se desean optimizar.

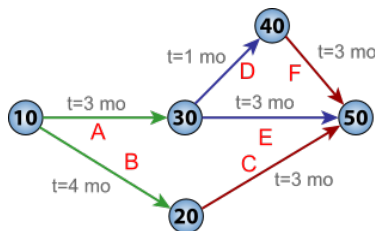
Aplicaciones

- Las Técnicas de Revisión y Evaluación de Proyectos (PERT) facilitan la administración y gestión de proyectos.
- La planeación de actividades se representa a través de grafos donde:
 - Los arcos representan actividades y sus pesos el tiempo que estas requieren.
 - Los vértices establecen precedencia de actividades.



Aplicaciones

- Una ruta es una secuencia de actividades que deben realizarse en determinado orden.
- La **ruta crítica** es la ruta que tarda la mayor cantidad de tiempo en realizarse. Es una cota inferior del tiempo requerido para realizar todas las actividades.
- Se puede encontrar la ruta crítica calculando la ruta más corta, para lo cual se vuelven negativos los tiempos.



Contenido

- 1 Preliminares
 - Grafos
 - Rutas
- 2 Problema de la Ruta Más Corta
 - Problema
 - Aplicaciones
- 3 Solución al Problema de la Ruta Más Corta
 - Aspectos Importantes y Representación
 - Propiedades de Rutas Más Cortas y Relajación
 - Algoritmo de Bellman-Ford

Generalidades

- El algoritmo de Bellman-Ford soluciona el problema de la Ruta Más Corta - Fuente Única.
- Los pesos del grafo pueden ser negativos.
- Si el grafo contiene ciclos de peso negativo, el algoritmo lo reporta retornando falso.
- En caso contrario, se retorna verdadero y se calcula la ruta más corta y su peso $\delta(s, v)$ para todo $v \in V$.

Aspectos Importantes y Representación

El algoritmo BELLMAN-FORD tiene en cuenta los siguientes aspectos:

- Sub-estructura óptima de la ruta más corta.
- Arcos con pesos negativos.
- Ciclos.
- Representación de la ruta más corta.
- Estimación de la ruta más corta.
- Relajación.

Sub-estructura óptima de la ruta más corta

Lemma 10 (Sub-rutas de la ruta más corta son rutas más cortas).

Sea $G = (V, E)$ un grafo dirigido con función de pesos $\omega : E \rightarrow \mathbb{R}$ y $p = \langle v_0, v_1, \dots, v_k \rangle$ una ruta más corta desde el vértice v_0 hasta el vértice v_k , donde $v_i \in V$ para $i \in \{0, 1, \dots, k\}$. Además, sea $p_{ij} = \langle v_i, v_{i+1}, \dots, v_j \rangle$, donde $0 \leq i \leq j \leq k$, una sub-ruta de p desde el vértice v_i hasta el vértice v_j . Entonces, p_{ij} es una ruta más corta desde v_i hasta v_j .

Sub-estructura óptima de la ruta más corta

Demostración.

Si se descompone p en $v_0 \xrightarrow{p_{0i}} v_i \xrightarrow{p_{ij}} v_j \xrightarrow{p_{jk}} v_k$, entonces

$$\omega(p) = \omega(p_{0i}) + \omega(p_{ij}) + \omega(p_{jk}).$$

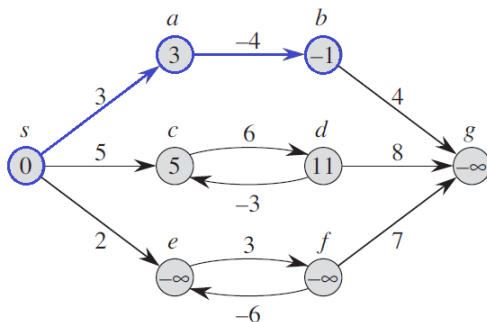
Supóngase que hay una ruta p'_{ij} tal que $\omega(p'_{ij}) < \omega(p_{ij})$. Entonces, la ruta p' conformada por $v_0 \xrightarrow{p_{0i}} v_i \xrightarrow{p'_{ij}} v_j \xrightarrow{p_{jk}} v_k$ tiene peso:

$$\omega(p') = \omega(p_{0i}) + \omega(p'_{ij}) + \omega(p_{jk}) < \omega(p).$$

Esto contradice el supuesto que p es una ruta más corta. □

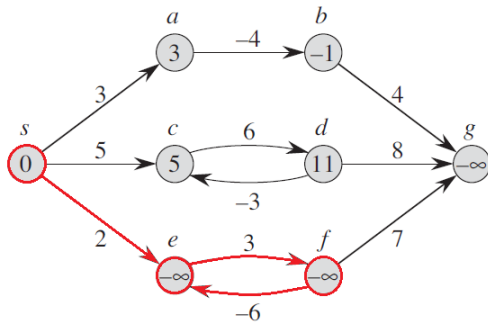
Arcos con Pesos Negativos

Un grafo dirigido $G = (V, E)$, con función de pesos $\omega : E \rightarrow \mathbb{R}$, puede tener pesos negativos y aún así tener rutas más cortas definidas. Incluso dichas rutas pueden tener peso negativo.



Arcos con Pesos Negativos

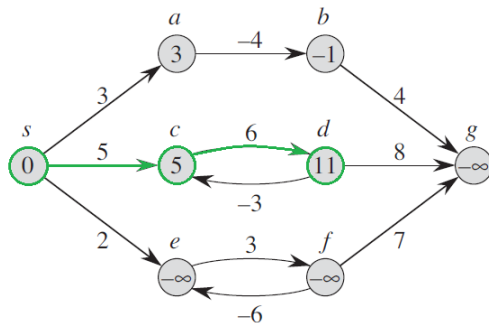
Sin embargo, si el grafo contiene ciclos con peso negativo, no existe ruta más corta hasta los vértices que sean alcanzables desde algún vértice de dicho ciclo. La distancia hasta dichos vértices es $\delta(s, v) = -\infty$.



Ciclos

Una ruta más corta no tiene ciclos c porque:

- Si $\omega(c) < 0$, la ruta más corta no está definida.
- Si $\omega(c) > 0$, remover c de la ruta produce una ruta con menor peso.

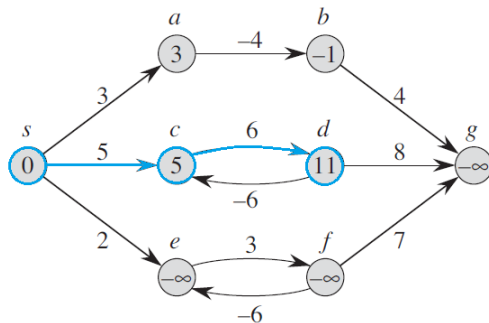


Ciclos

Una ruta más corta no tiene ciclos c porque:

- Si $\omega(c) = 0$, existe una ruta con el mismo peso que no contenga c .

Entonces la ruta más corta tiene máximo $|V|$ vértices y $|V| - 1$ arcos.



Representación de Rutas Más Cortas

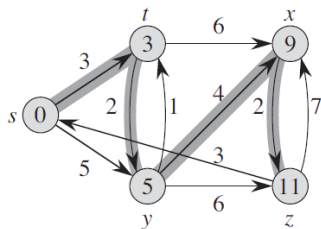
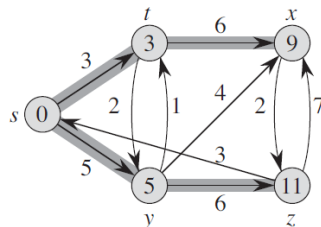
- Dado el grafo $G = (V, E)$, para cada vértice $v \in V$ se mantiene un **predecesor** $v.\pi$.
- Si $v.\pi = NIL$, v no es alcanzable desde s .
- Si $v.\pi \neq NIL$, la ruta más corta entre s y v , es la cadena de predecesores desde v hacia atrás hasta llegar a s .

Representación de Rutas Más Cortas

$\text{PRINT-PATH}(G, s, v)$

1. **if** $v = s$ **then**
 2. **print** s
 3. **else if** $v.\pi = \text{NIL}$ **then**
 4. **print** "no path"
 5. **else**
 6. $\text{PRINT-PATH}(G, s, v.\pi)$
 7. **print** v
-

Complejidad: lineal.



Estimación de Ruta Más Corta

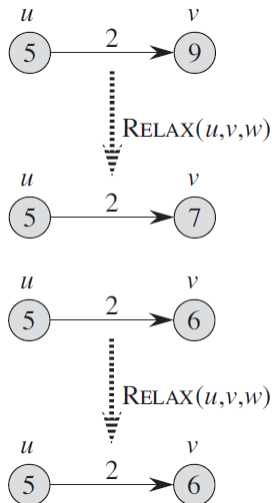
Dado el grafo $G = (V, E)$, para cada uno de los vértices $v \in V$ se hace una estimación de ruta más corta, la cual se denota como $v.d$. Estas se inicializan como se ilustra a continuación.

INITIALIZE-SINGLE-SOURCE(G, s)

1. **for each vertex** $v \in G.V$ **do**
 2. $v.d = \infty$
 3. $v.\pi = \text{NIL}$
 4. $s.d \leftarrow 0$
-

Complejidad: $\Theta(|V|)$.

Relajación



Relajar el arco (u, v) consiste en evaluar si se puede mejorar la ruta más corta que se tiene hasta v considerando la ruta a través de u .

$\text{RELAX}(u, v, \omega)$

1. **if** $v.d > u.d + \omega(u, v)$ **then**
 2. $v.d = u.d + \omega(u, v)$
 3. $v.\pi = u$
-

Complejidad: $O(1)$.

Contenido

- 1 Preliminares
 - Grafos
 - Rutas
- 2 Problema de la Ruta Más Corta
 - Problema
 - Aplicaciones
- 3 Solución al Problema de la Ruta Más Corta
 - Aspectos Importantes y Representación
 - Propiedades de Rutas Más Cortas y Relajación
 - Algoritmo de Bellman-Ford

Propiedades de Rutas más Cortas y Relajación

Para probar que el algoritmo de Bellman-Ford es correcto, se requiere considerar las siguientes propiedades de las rutas más cortas y la relajación:

- Desigualdad triangular.
- Propiedad de la cota superior.
- Propiedad de la inexistencia de rutas.
- Propiedad de la convergencia.
- Propiedad de la relajación de rutas.

Desigualdad Triangular

Lemma 11 (Desigualdad Triangular).

Sea $G = (V, E)$ un grafo dirigido con función de pesos $\omega : E \rightarrow \mathbb{R}$ y s el vértice fuente. Entonces, para todos los arcos $(u, v) \in E$, se tiene:

$$\delta(s, v) \leq \delta(s, u) + \omega(u, v)$$

Demostración.

Supóngase que p es una ruta más corta desde s hasta v . Entonces $\delta(s, v) = \omega(p)$ no es mayor que ninguna otra ruta hasta v , incluyendo aquella que tome una ruta más corta hasta u y luego el arco (u, v) . □

Propiedad de la Cota Superior

Lemma 12 (Propiedad de la cota superior).

Sea $G = (V, E)$ un grafo dirigido con función de pesos $\omega : E \rightarrow \mathbb{R}$ y s el vértice fuente. Además, sea G inicializado con $\text{INITIALIZE-SINGLE-SOURCE}(G, s)$. Entonces, $v.d \geq \delta(s, v)$ para todo $v \in V$ y esta invariante se mantiene para cualquier secuencia de relajaciones de los arcos de G . Además, una vez $v.d$ alcanza su cota inferior $\delta(s, v)$, nunca cambia.

Propiedad de la Cota Superior

Demostración.

Se prueba la hipótesis inductiva $v.d \geq \delta(s, v)$ para cada $v \in V$. Para el caso base $v.d \geq \delta(s, v)$ para $v \in V - \{s\}$ ya que se asigna $v.d = \infty$. Para s , se asigna $s.d = 0 \geq \delta(s, s)$. Para el caso inductivo, considérese la relajación de (u, v) . Antes de dicha relajación la hipótesis se cumple para todos los vértices. Después de esta, la única estimación que puede cambiar es $v.d$. Si cambia,

$$\begin{aligned} v.d &= u.d + \omega(u, v) \\ &\geq \delta(s, u) + \omega(u, v) && \text{por la hipótesis inductiva} \\ &\geq \delta(s, v) && \text{por la desigualdad triangular.} \end{aligned}$$



Propiedad de la Inexistencia de Rutas

Corollary 13 (Propiedad de la inexistencia de rutas).

Supóngase que en el grafo dirigido $G = (V, E)$, con función de pesos $\omega : E \rightarrow \mathbb{R}$, no existe una ruta que conecte el vértice fuente s con un vértice dado $v \in V$. Entonces, después de que el grafo es inicializado con $\text{INITIALIZE-SINGLE-SOURCE}(G, s)$, se tiene que $v.d = \delta(s, v) = \infty$, y esta igualdad se mantiene invariante para cualquier secuencia de relajaciones de los arcos de G .

Demostración.

Por la Propiedad de la Cota Superior, se tiene que $\delta(s, v) = \infty \leq v.d$, y por la inicialización se establece $v.d = \infty = \delta(s, v)$. □

Propiedad de la Convergencia

Lemma 14.

Sea $G = (V, E)$ un grafo dirigido con función de pesos $\omega : E \rightarrow \mathbb{R}$ y $(u, v) \in E$. Entonces, inmediatamente se relaja el arco (u, v) , invocando $\text{RELAX}(u, v, \omega)$, se tiene:

$$v.d \leq u.d + \omega(u, v)$$

Demostración.

Si antes de $v.d > u.d + \omega(u, v)$ antes de la relajación, después de esta $v.d = u.d + \omega(u, v)$. Si por el contrario, $v.d \leq u.d + \omega(u, v)$ antes de la relajación, esta desigualdad se mantiene ya que ninguna estimación cambia. □

Propiedad de la Convergencia

Lemma 15 (Propiedad de la convergencia).

Sea $G = (V, E)$ un grafo dirigido con función de pesos $\omega : E \rightarrow \mathbb{R}$ y s el vértice fuente. Además, sea $s \rightsquigarrow u \rightarrow v$ una ruta más corta en G donde $u, v \in V$. Supóngase que G es inicializado con $\text{INITIALIZE-SINGLE-SOURCE}(G, s)$ y que luego una secuencia de relajaciones que incluye $\text{RELAX}(u, v, \omega)$ se ejecuta sobre los arcos de G . Si $u.d = \delta(s, u)$ en algún tiempo anterior a la invocación, entonces $v.d = \delta(s, v)$ en todo momento posterior a la invocación.

Propiedad de la Convergencia

Demostración.

Si $u.d = \delta(s, u)$ en algún momento previo a la relajación de (u, v) , esta estimación se mantiene. Después de la relajación, se tiene que:

$$\begin{aligned} v.d &\leq u.d + \omega(u, v) && \text{por el Lema 14} \\ &= \delta(s, u) + \omega(u, v) \\ &= \delta(s, v) && \text{por el Lema 10.} \end{aligned}$$

Por la Propiedad de la Cota Superior, se tiene que $v.d \geq \delta(s, v)$.
Entonces, $v.d = \delta(s, v)$. □

Propiedad de la Relajación de Rutas

Lemma 16 (Propiedad de la relajación de rutas).

Sea $G = (V, E)$ un grafo dirigido con función de pesos $\omega : E \rightarrow \mathbb{R}$ y s el vértice fuente. Considérese una ruta más corta $p = \langle v_0, v_1, \dots, v_k \rangle$ desde $s = v_0$ hasta v_k . Si G es inicializado con $\text{INITIALIZE-SINGLE-SOURCE}(G, s)$ y luego una secuencia de relajaciones que incluye, en orden, la relajación de los arcos $(v_0, v_1), (v_1, v_2), \dots, (v_{k-1}, v_k)$, entonces $v_k = \delta(s, v_k)$ después de estas relajaciones y en todo tiempo posterior. Esta propiedad se cumple incluso si se agregan relajaciones adicionales intermezcladas con las de los arcos de p .

Propiedad de la Relajación de Rutas

Demostración.

Se muestra por inducción que después que el arco (v_{i-1}, v_i) se relaja, se tiene que $v_i.d = \delta(s, v_i)$.

Para la base, $i = 0$, antes de relajar los arcos de p , se inicializó $v_0.d = s.d = 0 = \delta(s, s)$. Este valor no cambia por la Propiedad de la Cota Superior.

Para el caso inductivo, se asume que $v_{i-1}.d = \delta(s, v_{i-1})$. Después de relajar el arco (v_{i-1}, v_i) , se tiene $v_i.d = \delta(s, v_i)$ por la Propiedad de Convergencia. □

Contenido

1 Preliminares

- Grafos
- Rutas

2 Problema de la Ruta Más Corta

- Problema
- Aplicaciones

3 Solución al Problema de la Ruta Más Corta

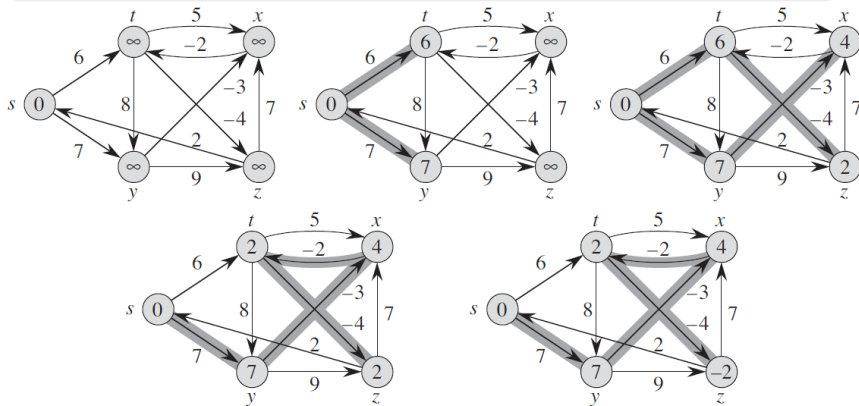
- Aspectos Importantes y Representación
- Propiedades de Rutas Más Cortas y Relajación
- Algoritmo de Bellman-Ford

Algoritmo de Bellman-Ford

BELLMAN-FORD(G, ω, s)

1. INITIALIZE-SINGLE-SOURCE(G, s)
 2. **for** $i \leftarrow 1$ **to** $|G.V| - 1$ **do**
 3. **for each edge** $(u, v) \in G.E$ **do**
 4. RELAX(u, v, ω)
 5. **for each edge** $(u, v) \in G.E$ **do**
 6. **if** $v.d > u.d + \omega(u, v)$ **do**
 7. **return false**
 8. **return true**
-

Ejemplo

$$(t, x), (t, y), (t, z), (x, t), (y, x), (y, z), (z, x), (z, s), (s, t), (s, y)$$


Complejidad Computacional en Tiempo

BELLMAN-FORD(G, ω, s)

1.	INITIALIZE-SINGLE-SOURCE(G, s)	$\Theta(V)$
2.	for $i \leftarrow 1$ to $ G.V - 1$ do	$ V - 1$ iteraciones
3.	for each edge $(u, v) \in G.E$ do	$ E $ arcos
4.	RELAX(u, v, ω)	$O(1)$
5.	for each edge $(u, v) \in G.E$ do	$ E $ arcos
6.	if $v.d > u.d + \omega(u, v)$ do	$O(1)$
7.	return false	$O(1)$
8.	return true	$O(1)$

Complejidad: $O(|V| \times |E|)$.

Correctitud

Con base en estas propiedades, se puede probar la correctitud del algoritmo de BELLMAN-FORD a través del siguiente lema y teorema.

Lemma 17.

Sea $G = (V, E)$ un grafo dirigido con función de pesos $\omega : E \rightarrow \mathbb{R}$ y s el vértice fuente. Supóngase que G no contiene ciclos de peso negativo que sean alcanzables desde s . Entonces, después de $|V| - 1$ iteraciones del ciclo de las líneas 2 – 4 de BELLMAN-FORD, se tiene que $v.d = \delta(s, v)$ para todos los vértices $v \in V$ que sean alcanzables desde s .

Correctitud

Demostración.

Considérese el vértice v que sea alcanzable desde s y que $p = \langle v_0, v_1, \dots, v_k \rangle$ sea una ruta más corta a este. Dado que las rutas más cortas no contienen ciclos, p contiene máximo $|V|$ vértices y $|V| - 1$ arcos., i.e. $k \leq |V| - 1$. Cada una de las iteraciones del ciclo de las líneas 2 – 4 del algoritmo BELLMAN-FORD relaja todos los arcos. Entonces, dentro de las relajaciones realizadas en la iteración i -ésima, para $i \in \{1, 2, \dots, |V| - 1\}$, se relaja el arco (v_{i-1}, v_i) . Por tanto, por la Propiedad de la Relajación de rutas, se tiene que $v.d = v_k.d = \delta(s, v_k) = \delta(s, v)$. □

Correctitud

Corollary 18.

Sea $G = (V, E)$ un grafo dirigido con función de pesos $\omega : E \rightarrow \mathbb{R}$ y s el vértice fuente. Supóngase que G no contiene ciclos de peso negativo que sean alcanzables desde s . Entonces, para cada vértice $v \in V$, hay una ruta desde s hasta v si y solo si BELLMAN-FORD termina con $v.d < \infty$ al ser ejecutado sobre G .

Correctitud

Theorem 19 (Correctitud del Algoritmo BELLMAN-FORD).

Sea el algoritmo BELLMAN-FORD ejecutado sobre el grafo dirigido $G = (V, E)$ con función de pesos $\omega : E \rightarrow \mathbb{R}$ y vértice fuente s . Si G no contiene ciclos de peso negativo que sean alcanzables desde s , entonces:

- *El algoritmo retorna verdadero.*
- *Se tiene $v.d = \delta(s, v)$ para todos los vértices $v \in V$.*

Si G contiene algún ciclo de peso negativo alcanzable desde s , el algoritmo retorna falso.

Correctitud

Demostración.

Supóngase que G no contiene ningún ciclo de peso negativo. Primero se muestra que $v.d = \delta(v, d)$ para todos los vértices $v \in V$. Si v es alcanzable desde s , esto se cumple por el Lema 17. Si v no es alcanzable, se cumple por la Propiedad de Inexistencia de Rutas. El algoritmo retorna **verdadero** porque, al finalizar la línea 4, para todos los arcos (u, v) se cumple que:

$$\begin{aligned} v.d &= \delta(s, v) \\ &\leq \delta(s, u) + \omega(u, v) && \text{por Desigualdad Triangular} \\ &= u.d + \omega(u, v). \end{aligned}$$



Correctitud

Demostración.

Supóngase que G contiene un ciclo de peso negativo alcanzable desde s . Sea $c = \langle v_0, v_1, \dots, v_k \rangle$ dicho ciclo; $v_0 = v_k$ y $\omega(c) < 0$. Para obtener una contradicción, asumir que BELLMAN-FORD retorna **verdadero**. Se cumple que $v_i.d \leq v_{i-1}.d + \omega(v_{i-1}, v_i)$ para $i \in \{1, 2, \dots, k\}$. Sumando estas desigualdades se obtiene:

$$\sum_{i=1}^k v_i.d \leq \sum_{i=1}^k v_{i-1}.d + \sum_{i=1}^k \omega(v_{i-1}, v_i)$$

Pero como $v_0 = v_k$, $\sum_{i=1}^k v_i.d = \sum_{i=1}^k v_{i-1}.d$, se tiene $0 \leq \omega(c)$ lo cual contradice que el ciclo tiene peso negativo. \square

Conclusiones

- El problema de la ruta más corta tiene aplicaciones importantes en diferentes campos.
- Si hay ciclos de peso negativo alcanzables desde la fuente, la ruta más corta no está definida.
- El algoritmo **BELLMAN-FORD** permite reportar si un grafo con pesos tiene ciclos de pesos negativos o, en caso contrario, encontrar la ruta más corta desde una fuente única hasta todos sus vértices en $O(|V| \times |E|)$ tiempo.

Bibliografía

- T. Cormen, C. Leiserson, R. Rivest, C. Stein. Introduction to Algorithms, Third Edition. The MIT Press. 2009.
- K. Ruohonen. Graph Theory. 2008.

Gracias

¿Preguntas?