CI2613: Algoritmos y Estructuras III

Blai Bonet

Universidad Simón Bolívar, Caracas, Venezuela

Enero-Marzo 2015

Caminos de costo mínimo en grafos

Algoritmo de Johnson

Caminos entre todos los pares de vértices

El algoritmo de Floyd-Warshall calcula las distancias y caminos más cortos entre todos los pares de vértices en tiempo $\Theta(V^3)$

En **grafos densos**, donde $E = \Omega(V^2)$, Floyd-Warshall es **asintóticamente** el mejor algoritmo conocido

En **grafos dispersos**, donde $E = o(V^2)$, existen mejores algoritmos:

- Si los costos son no-negativos, podemos correr Dijkstra V veces en tiempo $O(V^2 \log V + VE)$ para obtener un mejor algoritmo
- El algoritmo de Johnson obtiene el mismo tiempo de corrida $O(V^2 \log V + VE)$ pero sin restricción en los pesos. Si existe un ciclo de costo negativo, el algoritmo reporta su existencia y termina

Caminos entre todos los pares de vértices

El algoritmo de Johnson utiliza la técnica de **cambio de pesos:**

- Si todos las aristas tienen costos no negativos, se corre Dijkstra desde cada vértice s utilizando un heap de Fibonacci
- Si existen costos negativos pero no ciclos de costo negativo, se calculan nuevo pesos no negativos $\hat{w}$ y volvemos al caso anterior

  Los nuevos pesos $\hat{w}$ deben satisfacer:

1. Para todo par de vértices $u, v \in V$: p es un mejor camino de u a v con respecto a w si y sólo si p es un mejor camino de u a v con respecto a $\hat{w}$
2. Para todas las aristas $(u, v) \in E$: $\hat{w}(u, v) \geq 0$

Cambio de peso: Repesaje

El **repesaje** de $w : E \to \mathbb{R}$ lo hacemos con una función $h : V \to \mathbb{R}$ que mapea vértices en valores

El repesaje de w es $\hat{w} : E \to \mathbb{R}$ defnida por

$$\hat{w}(u,v) \;=\; w(u,v) + h(u) - h(v)$$

Lema

Sea $G = (V, E)$ un digrafo con pesos $w : E \to \mathbb{R}$. Considere el repesaje $\hat{w} : E \to \mathbb{R}$. Sea p un camino de u a v en G. Entonces, p es un camino más corto con respecto a w si y sólo si p es un camino más corto con respecto a $\hat{w}$.

También, G tiene un costo de ciclo negativo con respecto a w si y sólo si G tiene un ciclo de costo negativo con respecto a $\hat{w}$.

Claramente, el Lema garantiza la propiedad ❶

Cambio de peso: Repesaje

Prueba del Lema: Sea $p = (v_0, \ldots, v_k)$ con $v_0 = u$ y $v_k = v$

Primero mostramos $\hat{w}(p) = w(p) + h(u) - h(v)$:

$$\begin{aligned}
\hat{w}(p) &= \textstyle\sum_{i=1}^{k} \hat{w}(v_{i-1}, v_i) \\
&= \textstyle\sum_{i=1}^{k} w(v_{i-1}, v_i) + h(v_{i-1}) - h(v_i) \\
&= \textstyle\sum_{i=1}^{k} w(v_{i-1}, v_i) + \sum_{i=1}^{k} h(v_{i-1}) - h(v_i) \\
&= w(p) + h(v_0) - h(v_k) \\
&= w(p) + h(u) - h(v)
\end{aligned}$$

Como $h(u)$ y $h(v)$ no dependen de p, p es óptimo relativo a w si y sólo si p es óptimo relativo a $\hat{w}$

Si $c = (v_0, \ldots, v_k)$ es un ciclo con $v_0 = v_k$,

$$\hat{w}(c) \;=\; w(c) + h(v_0) - h(v_k) \;=\; w(c)$$

Por lo tanto, $w(c) < 0$ si y sólo si $\hat{w}(c) < 0$ □

Pesos no negativos

Propiedad ❷: $\hat{w}(u,v) \geq 0$ para toda arista $(u,v) \in E$

Dado digrafo $G = (V, E)$, **aumentamos** G con un **vértice fuente** s: $G' = (V', E')$ donde $V' = V \cup \{s\}$ y $E' = E \cup \{(s,u) : u \in V\}$

Los pesos también son extendidos:

$$w'(u,v) \;=\; \begin{cases} w(u,v) & \text{si } (u,v) \in E \\ 0 & \text{si } (u,v) = (s,v) \text{ con } v \in V \end{cases}$$

Observación: G tiene un ciclo de costo negativo si y sólo si G' tiene un ciclo de costo negativo **alcanzable** desde s

Pesos no negativos

Considere $h : V \to \mathbb{R}$ dada por $h(u) = \delta'(s,u)$

Por la desigualdad triangular en G', para arista $(u,v) \in E$:

$$\delta'(s,v) \;\leq\; \delta'(s,u) + w'(u,v) \implies \delta'(s,u) - \delta'(s,v) \geq -w'(u,v)$$

Por lo tanto, para arista $(u,v) \in E$,

$$\begin{aligned}
\hat{w}(u,v) &= w(u,v) + h(u) - h(v) \\
&= w'(u,v) + \delta'(s,u) - \delta'(s,v) \\
&\geq w'(u,v) - w'(u,v) \\
&= 0
\end{aligned}$$

Algoritmo de Johnson

1. Dado un digrafo $G = (V, E)$, se construye el digrafo aumentado $G' = (V', E')$ con pesos $w' : E' \to \mathbb{R}$ **[Tiempo: $O(V + E)$]**

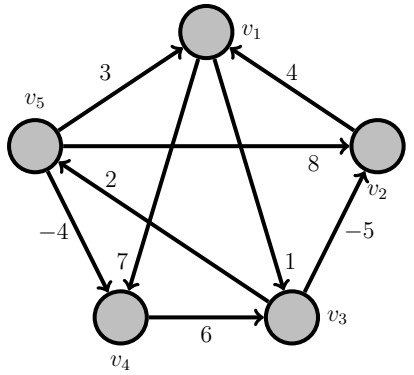
2. Correr Bellman-Ford para calcular las distancias $\delta'(s, u)$ en el digrafo G' para $u \in V$ **[Tiempo: $O(V'E') = O(VE + V^2)$]**

3. Calcular repesaje $\hat{w} : E \to \mathbb{R}$ relativo a $h(u) = \delta'(s, u)$ para $u \in V$ **[Tiempo: $O(E)$]**

4. Corremos el algoritmo de Dijkstra $|V|$ veces desde todos los vértices $s \in V$ **[Tiempo: $O(V^2 \log V + VE)$]**

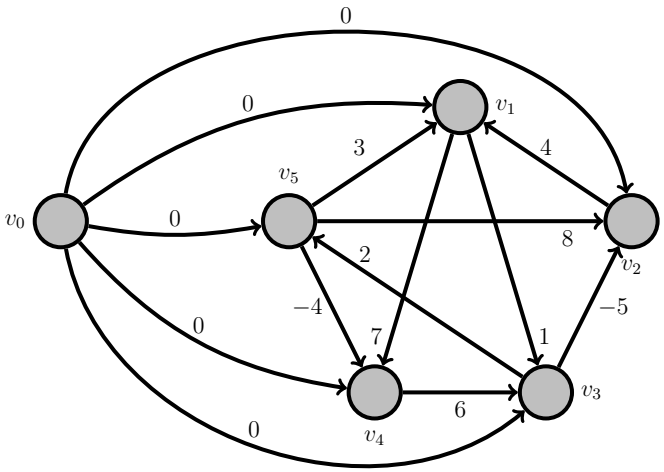
5. Distancias finales: $\delta(u, v) = \hat{\delta}(u, v) - h(u) + h(v)$ **[Tiempo: $O(V^2)$]**

Algoritmo de Johnson: Ejemplo



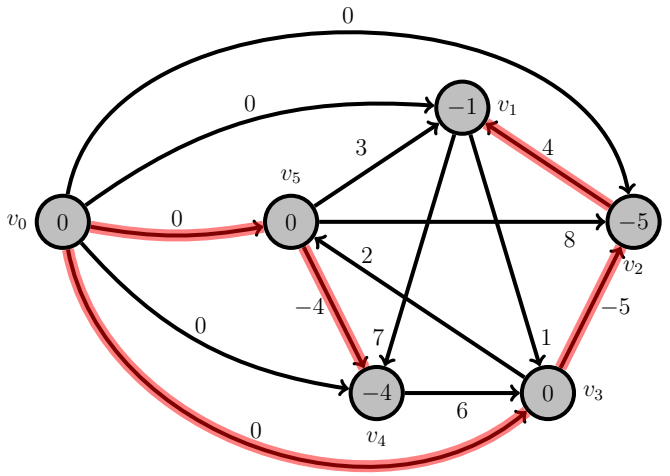
Digrafo $G = (V, E)$

Algoritmo de Johnson: Ejemplo



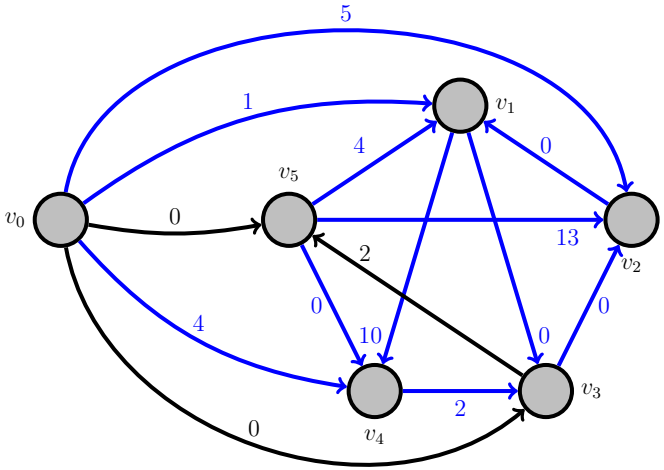
Digrafo aumentado $G' = (V', E')$

Algoritmo de Johnson: Ejemplo



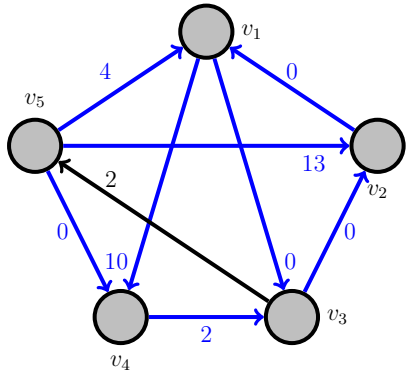
Después de correr Bellman-Ford desde v_0

Algoritmo de Johnson: Ejemplo



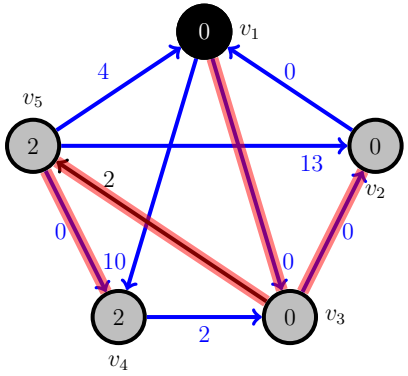
Repesaje de aristas

Algoritmo de Johnson: Ejemplo



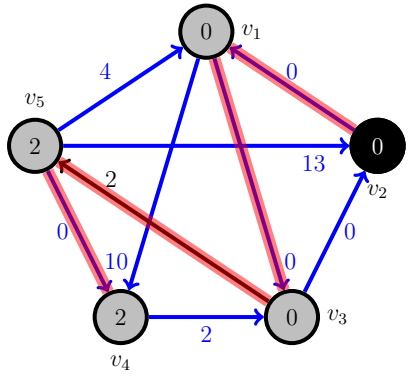
Repesaje de aristas sobre grafo original

Algoritmo de Johnson: Ejemplo



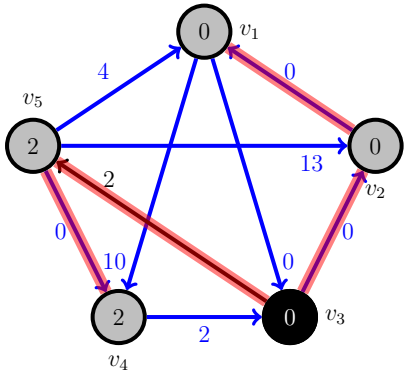
Después de correr Dijkstra desde el vértice v_1

Algoritmo de Johnson: Ejemplo



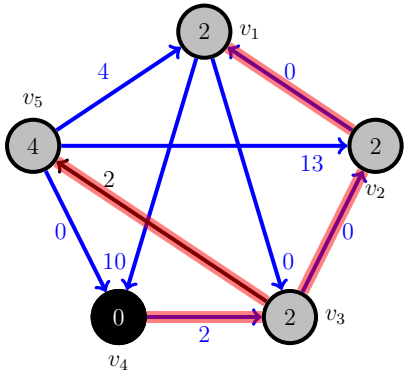
Después de correr Dijkstra desde el vértice v_2

Algoritmo de Johnson: Ejemplo



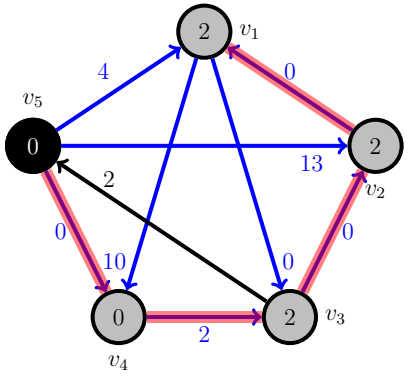
Después de correr Dijkstra desde el vértice v_3

Algoritmo de Johnson: Ejemplo



Después de correr Dijkstra desde el vértice v_4

Algoritmo de Johnson: Ejemplo



Después de correr Dijkstra desde el vértice v_5