

# Representación mediante grafos de un problema de asignación de horarios

José Alberto Benavides Vázquez

8 de abril de 2018

## 1. Introducción

Los algoritmos de Floyd-Warshall y Ford-Fulkerson se utilizan en teoría de grafos para realizar mediciones de flujos a grafos ponderados. **Floyd-Warshall** calcula el camino más corto entre todos los pares de vértices, mientras que **Ford-Fulkerson** se utiliza para encontrar el máximo flujo que puede correr entre un nodo de inicio y uno de fin dados. Los nodos de inicio y fin de este segundo algoritmo se suelen denotar por  $s$  y  $t$  respectivamente.

Este par de algoritmos se implementaron en un programa capaz de crear nodos, conectarlos y asignarles pesos y direcciones a las aristas así generadas. Este programa se halla alojado en <https://github.com/jbenavidesv87/FlujoRedes>. Se mostrará a continuación la explicación de dichos algoritmos además de ejecutarlos con un ejemplo para mayor claridad.

## 2. Teoría y ejemplos

Tanto el algoritmo de Floyd-Warshall como el de Ford-Fulkerson toman un grafo previamente definido con al menos dos nodos conectados entre sí por una arista ponderada. Para demostrar el uso y funcionalidad de estos algoritmos, se construyó a manera de ejemplo un grafo  $G$  no dirigido con cuatro nodos  $a, b, c, d$ , con aristas ponderadas como los mostrados en la figura 1 (p. 3).

Tras correr el algoritmo de Floyd-Warshall se obtuvieron, para cada par de nodos, el camino más corto entre todos los nodos del grafo. La distancia de estos caminos se calcula a partir de la suma de los pesos de las aristas que conectan a los nodos entre sí. En cada iteración, se exploran todos los caminos que van desde un nodo inicial hasta uno de destino. Se dice que un camino es mejor que otro cuando se encuentra un camino cuya distancia es menor a una existente (si no hay una distancia previamente almacenada, entonces se toma la primera distancia registrada como la mejor). Al final, el algoritmo regresa una lista de las mejores distancias entre todos los nodos. Para este ejemplo construido, la lista se puede consultar en el cuadro 1 (p. 2).

Tabla 1: Camino más corto entre todos los pares de nodos, denotados por cada iteración como nodo inicial  $s$  y nodo final  $t$ , para el grafo  $G$  construido a manera de ejemplo inicial.

Nodo inicial $s$	Nodo final $t$	Camino más corto
a	a	0
a	c	7
a	b	5
b	b	0
b	a	5
b	d	3
b	c	2
c	c	0
c	a	7
c	d	4
c	b	2
d	d	0
d	c	4
d	b	3
a	d	8
d	a	8

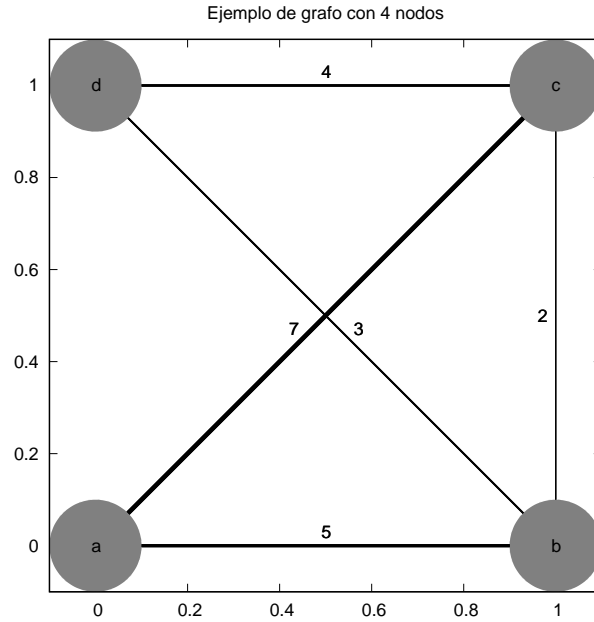


Figura 1: Grafo  $G$  ponderado con 4 nodos  $a, b, c, d$  que se usará como ejemplo para probar el funcionamiento de los algoritmos de Floyd-Warshall y Ford-Fulkerson.

El máximo flujo entre dos nodos se calcula con el algoritmo de Ford-Fulkerson. Este algoritmo consiste en recorrer todos los caminos que van desde un nodo inicial  $s$  hasta uno final  $t$ , tomando como limitante el peso mínimo entre los pesos de los arcos que constituyen cada camino. Ese peso limitante indica el máximo flujo que podría soportar dicho camino. Cada iteración, se actualiza el grafo en el sentido en que se resta al peso de cada arco del camino elegido, el peso del camino mínimo. Cada iteración posterior, elegirá caminos en los que todos sus arcos posean capacidad disponible y se repite el proceso. Cuando no haya más caminos con arcos que tengan capacidades disponibles, se suman los pesos actualizados del grafo resultante, camino por camino, y se obtiene de su suma el máximo flujo del nodo  $s$  al  $t$ . Del ejemplo graficado en la figura 1 (p. 3) se calculó el flujo máximo entre los nodos  $a$  y  $b$ , obteniendo por resultado 10; y entre los nodos  $a$  y  $d$ , 7.

Ahora, para mostrar el funcionamiento en un grafo dirigido, estos mismos algoritmos se correrá con el mismo grafo, pero dirigido, tal como se muestra en la figura 2 (p. 4). A este nuevo grafo se le llamará  $H$ .

Tras correr el algoritmo de Floyd-Warshall sobre el grafo  $H$  se obtienen los caminos más cortos mostrados en el cuadro 2 (5).

Por su parte, el algoritmo de Ford-Fulkerson en el grafo  $H$ , da por resultado 5 del nodo  $a$  al  $b$ ; y 7 del nodo  $a$  al  $d$ .

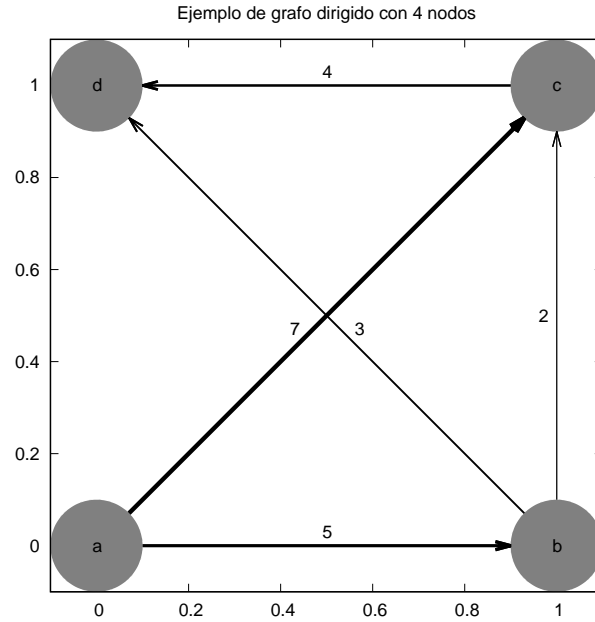


Figura 2: Grafo  $H$  ponderado y dirigido con 4 nodos  $a, b, c, d$  que se usará como ejemplo para probar el funcionamiento de los algoritmos de Floyd-Warshall y Ford-Fulkerson.

Este par de ejemplos están contenidos en un mismo código que se puede encontrar comentado en <https://github.com/jbenavidesv87/FlujoRedes/blob/master/ejemplos/05FloydWarshall/main.py>.

En la siguiente sección, se estudiará el tiempo de ejecución que toma correr cada algoritmo con diferentes tamaños de grafos.

### 3. Tiempos de ejecución

Para medir los tiempos de ejecución se realizó un programa que generaba nueve grafos dirigidos con una cantidad de nodos igual a las siguientes potencias de 2: 2, 4, 8, 16, 32, 64, 128 y 256. A cada uno de estos grafos se les efectuó la siguiente serie de acciones:

1. Se crearon  $n$  nodos correspondientes a la potencia del grafo en turno.
2. A cada nodo se le asignó una posición al azar en un espacio unitario bidimensional.
3. Se calculó la cantidad total de aristas que se podrían establecer entre todas las parejas de nodos, como  $t = n(n - 1)$ .

Tabla 2: Camino más corto entre todos los pares de nodos, denotados por cada iteración como nodo inicial  $s$  y nodo final  $t$ , para el grafo  $H$  construido a manera de ejemplo para nodos dirigidos.

Nodo inicial $s$	Nodo final $t$	Camino más corto
a	a	0
a	c	7
a	b	5
b	b	0
b	c	2
b	d	3
c	c	0
c	d	4
d	d	0
a	d	8

4. Se crearon un total de  $\text{techo}(0.1t)$  aristas entre parejas de nodos elegidos al azar, cuidando que no se repitieran vecinos.
5. Se calculó el tiempo de ejecución del algoritmo de Floyd-Warshall.
6. Se eligió al azar una pareja de nodos distintos que sirvieron de base para correr el algoritmo de Ford-Fulkerson y se midió el tiempo que tomó arrojar un resultado.

Esta serie de acciones ejecutadas por cada uno de los nueve grafos se repitió veinte veces. Se obtuvieron como resultado grafos como el que se muestra en la figura 3 (6). Para grafos con tamaños menores a 64 nodos, el tiempo de ejecución del algoritmo de Floyd-Warshall es menor a un segundo, mientras que para grafos de 256 nodos, se alcanzan tiempos que rondan los once segundos. Los tiempos de corrida del algoritmo de Ford-Fulkerson tienen tiempos menores a un segundo en grafos de 32, mientras que para grafos de 128 nodos o superiores, los tiempos podrían sobrepasar el minuto, habiéndose registrado tiempos de hasta 45 minutos para grafos con 256 nodos. Suponemos que estos tiempos tan dilatados en comparación con el resto responden a que se eligen al azar grafos muy distantes y con una gran variedad de caminos entre sí. Investigar este comportamiento resultaría interesante pero se escapa de los alcances de este reporte, por lo que se propone como trabajo futuro.

Los tiempos registrados para el algoritmo de Floyd-Warshall han sido representados en escala logarítmica y por cantidad de nodos en diagramas de caja y bigotes en la figura 4 (6), mientras que para el algoritmo de Ford-Warshall puede consultarse la figura 5 (7).

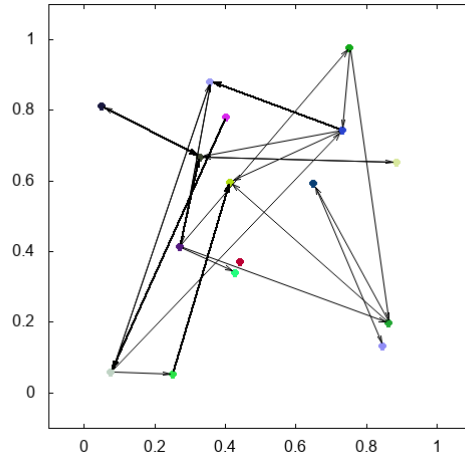


Figura 3: Ejemplo de grafo de 16 nodos con 24 arcos, equivalentes a un 10 % del total de arcos posibles, es decir  $16 \cdot 15 = 240$  arcos dirigidos posibles.

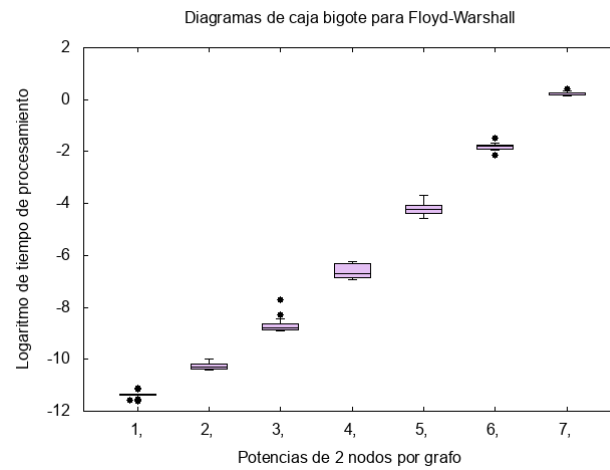


Figura 4: Diagrama de caja y bigotes de los tiempos en segundos en escala logarítmica de ejecución del algoritmo de Floyd-Warshall de los grupos de grafos generados al azar.

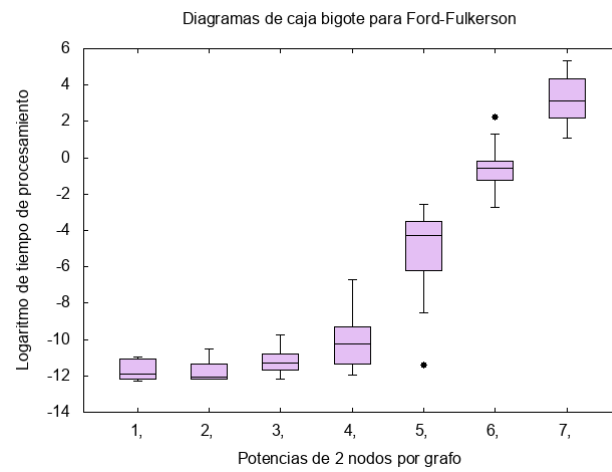


Figura 5: Diagrama de caja y bigotes de los tiempos en segundos en escala logarítmica de ejecución del algoritmo de Ford-Fulkerson de los grupos de grafos generados al azar.