

Grafos en red, aproximación a Ford-Fulkerson por algoritmo de contracción

José Alberto Benavides Vázquez

21 de mayo de 2018

Esta práctica se realizó a partir de un programa en desarrollo para flujo en redes alojado en <https://github.com/jbenavidesv87/FlujoRedes>. El código de esta práctica puede consultarse en <https://github.com/jbenavidesv87/FlujoRedes/tree/master/ejemplos/09Contraccion> [1].

En esta simulación se generó un grafo en forma con un total de 25^2 nodos distribuidos en forma de red de 25×25 . Cada nodo está conectado con sus vecinos inmediatos en distancia Manhattan igual a uno y con un peso $0.1 \geq p \geq 1$ elegido al azar mediante la función `random` de la librería `Math` de `Python` v. 3.0.

La disposición de grafos en forma de red se logró mediante la implementación del algoritmo descrito por [2]. Un ejemplo del grafo utilizado para esta práctica, con 10^2 nodos dispuestos de diez en diez, puede consultarse en la figura 1 (p. 2).

En esta práctica se ha implementado un algoritmo de contracción de nodos que permita aproximar el flujo máximo que existe entre un nodo inicial s y uno final t calculado por el algoritmo de Ford-Fulkerson. Primeramente, se eligen al azar dos nodos destinados a ser el nodo inicial s y el final t del grafo. A partir de estos nodos se calcula el flujo máximo mediante el algoritmo de Ford-Fulkerson. Posteriormente, se procede a realizar las contracciones. Las contracciones de nodos consisten en la elección al azar de un nodo n ; luego se toma uno de sus vecinos v de manera aleatoria. Si ese vecino no es s o t , v es contraído por n , lo que quiere decir que los vecinos de v pasan a ser vecinos de n y que v es eliminado. Cuando n ya cuenta entre sus vecindades con alguno de los vecinos de v , los pesos de las aristas que constituyen sus vecindades se suman. Esta operación se repite hasta que se tienen únicamente a s y a t conectados entre sí mediante un arco cuyo peso es la suma de todos los pesos elegidos al azar durante el algoritmo. Un ejemplo de este proceso puede consultarse en una animación gif alojada en <https://goo.gl/akzeSD> y una serie de cuatro imágenes en la figura 2 (p. 3).

A continuación se corrieron instancias del algoritmo hasta que se lograra aproximar el flujo un 5% por encima del flujo calculado por el algoritmo de Floyd-Warshall y se graficaron los resultados en la figura 3 (p. 4).

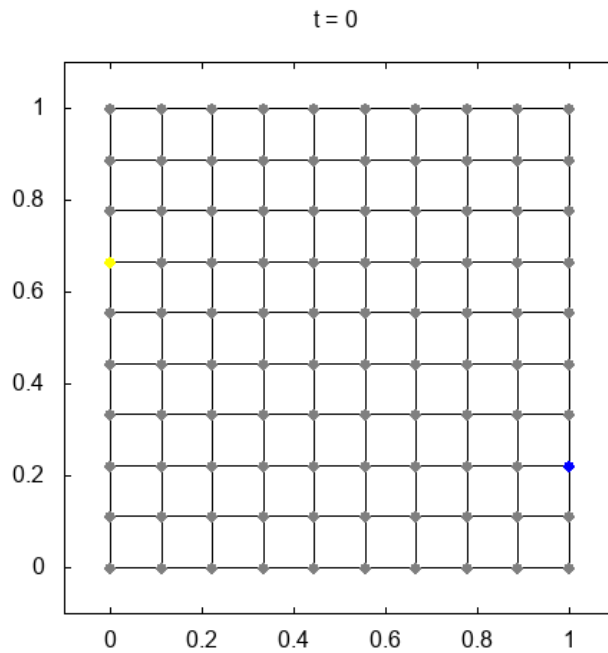


Figura 1: Ejemplo de grafo dispuesto en forma de red con 10^2 nodos colocados de diez en diez y conectados con vecinos a una distancia Manhattan de uno. En azul se muestra el nodo inicial s y en amarillo el nodo final t .

Referencias

- [1] José Alberto Benavides Vázquez. Grafos con python para flujo en redes.
<https://github.com/jbenavidesv87/FlujoRedes>.
- [2] José Alberto Benavides Vázquez. Grafos en red, distancia manhattan y percolación de vecindades.
<https://github.com/jbenavidesv87/FlujoRedes/blob/master/Reportes/p5/percolacion.pdf>.

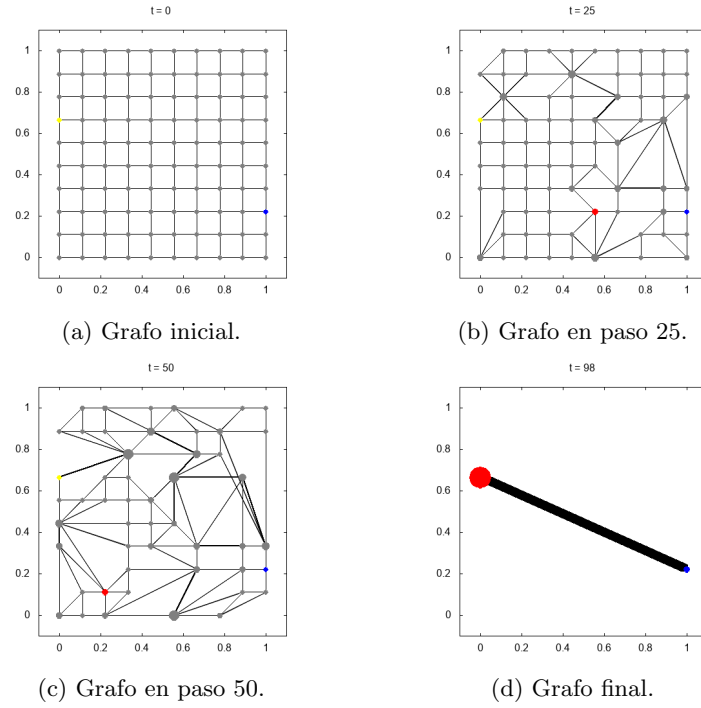


Figura 2: Muestra del avance de contracción en un grafo de 10^2 nodos dispuestos en forma de red. El grosor de los arcos indica el peso del nodo, el nodo en azul es el nodo inicial s , el amarillo es el final t y el rojo representa el nodo que en el paso mostrado fue contraído a su vecino.

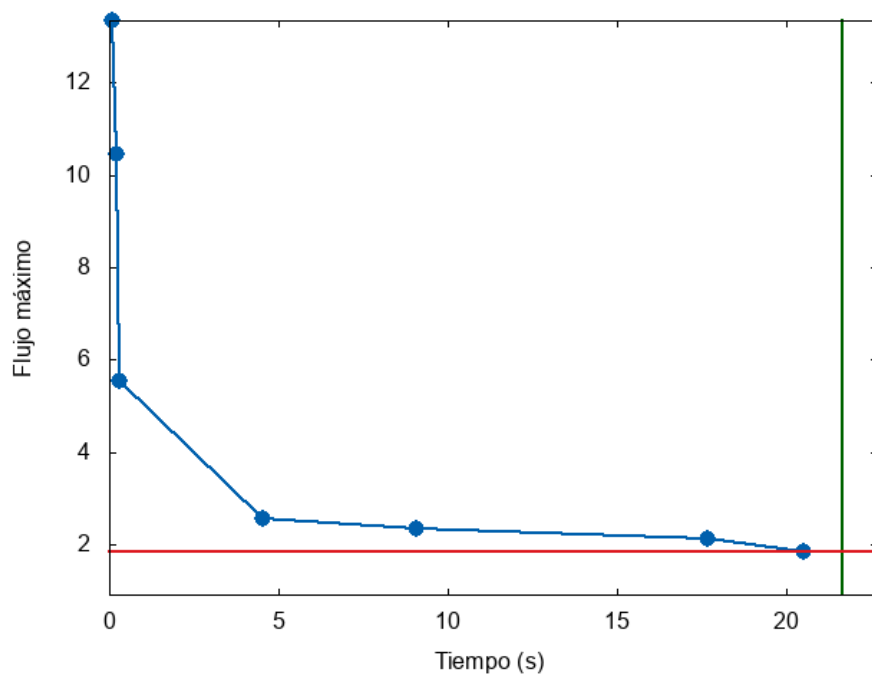


Figura 3: Gráfica del flujo máximo calculado por el tiempo en segundos de corrida del algoritmo diseñado para esta práctica (en azul). La línea horizontal roja representa el flujo máximo calculado por Floyd-Warshall y la línea vertical verde el tiempo que este último algoritmo tomó en ejecutarse.