Práctica 6

Jose David Martinez Ruiz

21 de mayo de 2018

1. Introducción

En esta última practica lo que se desea es crear una heurística para poder encontrar el flujo máximo entre dos nodos. Se espera encontrar una aproximación lo más cercana posible al valor optimo encontrado con el algoritmo de Ford-Fulkerson a través de esa heurística. Se espera que los valores obtenidos por esta heurística sean muy cercanos al óptimo.

2. Desarrollo

Se decidió utilizar la heurística en el tipo de grafo cuadrado que se vio en la practica anterior. En lo que consiste esta heurística es en comprimir los dos nodos que forma alguna arista al azar entre las aristas que conforman el grafo. Se elimina la arista escogida al azar y se "fusionan" los dos nodos que conformaban esa arista, es decir, ahora no hay dos nodos, sino uno que se encuentra en el punto medio de ese par de nodos. Algo más que se hace es que todos los nodos que se conectaban con los nodos que se "fusionaron" ahora están conectados con el nuevo nodo.

Al realizar este proceso es posible que varias aristas vayan de un nodo en particular a otro, esto sucede cuando se fusionan dos nodos, donde cada uno de ellos van a otro nodo. Así se forman dos aristas de un nodo a otro. Debido a que la arista que se escoge al azar, si se escoge una arista repetida podría causar problemas por cómo funciona nuestra heurística, así que, para eso, cuando se llega a escoger alguna arista repetida, se eliminan todas las aristas idénticas a la escogida al igual que sus correspondientes pesos.

Debido a que nos interesa saber cuál es el flujo máximo entre un par de nodos en particular, no se debe permitir que esos dos nodos se fusionen, además, para poder llevar un control de esto, ya que cada vez que se fusionan dos nodos se crea uno en medio de donde estaban los dos, cada vez que alguno de los dos nodos de los cuales nos interesa saber el flujo máximo que puede pasar entre ellos, se actualiza el nodo involucrado por el nodo resultante de la fusión.

Este proceso se hará hasta que solamente queden dos nodos, el par de nodos que nos interesan. Hay varias aristas repetidas que conectan a estos dos nodos, el flujo máximo que se obtiene con este método es igual a la suma de los pesos de las aristas que quedaron después de realizar todo el proceso.

El algoritmo de Ford-Fulkerson nos da el flujo máximo entre un par de nodos, esto lo obtiene al verificar todos los caminos posibles entre esos nodos y a partir de las capacidades de las aristas que conforman esos caminos, se obtiene el flujo máximo. La heurística que estamos implementado da una aproximación a lo obtenido con el algoritmo de Ford-Fulkerson, debido a que se escogen aristas al azar, es posible que se elimine una arista que forme parte de uno de los caminos que nos permite pasar más flujo entre ese par de nodos, por lo que, al finalizar el proceso, esto dará como resultado una cota inferior al valor encontrado con el algoritmo de Ford-Fulkerson.

Debido a que el azar afecta los resultados que se pueden obtener de esta heurística, se hicieron se hicieron pruebas con diferente tamaño de réplicas de todo el proceso y de los resultados obtenidos para cada tamaño de réplica, se escogió el más grande para que sea nuestra cota inferior al flujo máximo. Se hicieron las pruebas en un grafo cuadrado con k igual a 5 y l igual 3. En la

figura 1 se muestran los resultados obtenidos.

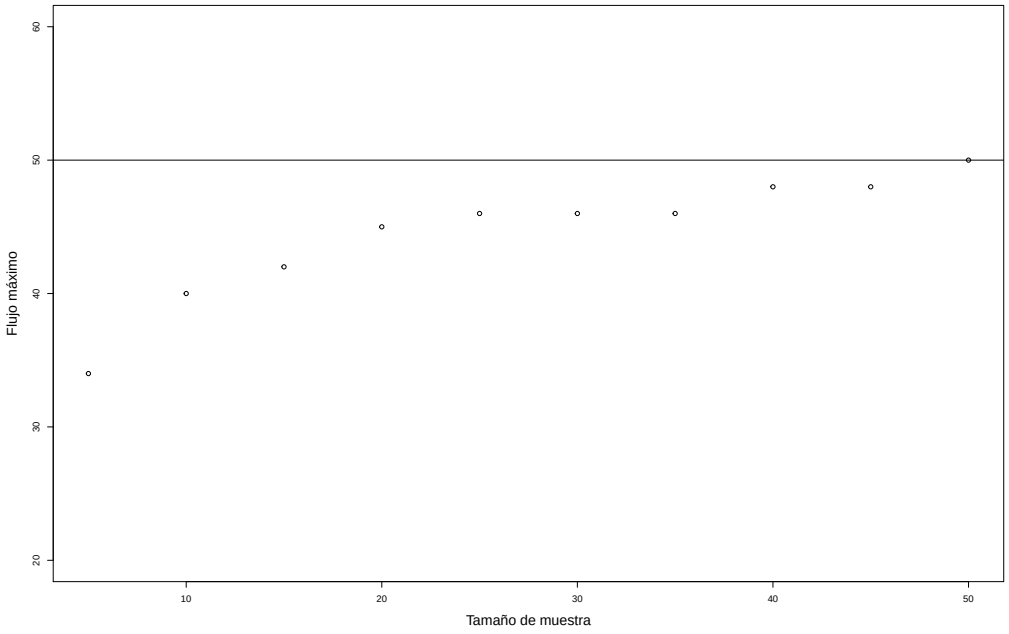

Figura 1: Flujo vs repeticiones

3. Conclusiones

Esta heurística sacrifica precisión a cambio de rapidez. El algoritmo de Ford-Fulkerson debido a que tiene que encontrar todos los caminos posibles de un par de nodos toma mucho tiempo cuando se trata de grafos más grandes. La heurística que se implementó lo hace de una forma más rápida debido a que solo tiene que juntar nodos, aunque tiene el riesgo de eliminar aristas clave para encontrar el flujo máximo óptimo. Esta heurística requiere de muchas repeticiones para poder encontrar un resultado más cercano al óptimo, aunque también debido al azar es posible encontrar un valor bastante bueno sin requerir demasiadas repeticiones.

Referencias

David Martinez. Tarea 5 Optimzación de flujo en redes. Abril 2018. https://github.com/DavidM0413/Flujo-en-Redes/blob/master/practica4/t5.py