

Tarea 6

Optimización de flujo en redes

Beatriz Alejandra García Ramos

21 de mayo de 2018

1. Introducción

En esta práctica se tiene un análisis sobre el flujo máximo que se obtiene de un grafo desde su nodo inicial hasta el nodo final realizando en lugar del algoritmo de Ford-Fulkerson el algoritmo de corte mínimo. En la práctica anterior [1] únicamente se realizó un análisis en el cual se tomaba en cuenta la percolación de nodos y aristas. Ahora se tiene en cuenta la unión de nodos y sus aristas correspondientes, donde se hará un proceso hasta quedar dos nodos finales que darán como resultado el flujo máximo del grafo.

La cantidad de nodos que se tomó en cuenta para la realización el grafo es de cien nodos.

2. Flujo máximo - Corte mínimo

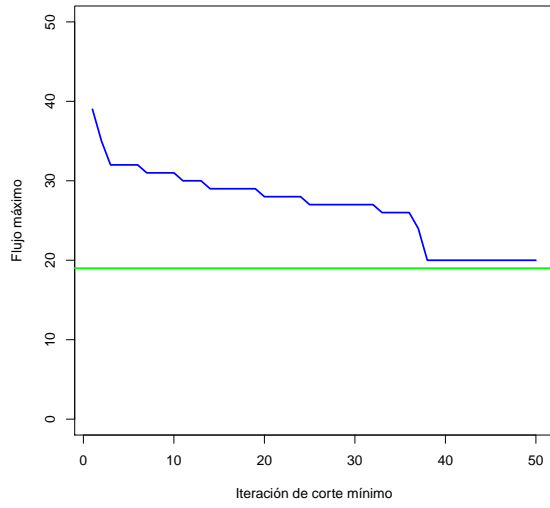
Para poder realizar el algoritmo del corte mínimo se deben simular uniones de nodos. Cuando un nodo se une con otro se olvidan esos dos nodos, se crea un nodo intermedio y la arista correspondiente entre ellos ya no se toma en cuenta; se tiene además que los vecinos correspondientes a esos dos nodos ahora son vecinos del nodo creado por la unión de ellos.

Para que se pueda lograr la unión se toma en cuenta elecciones al azar de nodos, una vez que ambos nodos se tienen se crea el nodo que está a una distancia media entre ellos dos, se elimina la arista correspondiente entre los nodos elegidos del diccionario de aristas, se unen los nodos vecinos de los dos nodos elegidos con el nuevo nodo y se crean aristas entre ellos, las aristas anteriores, aquellas que iban de los nodos elegidos a sus vecinos, se eliminan del diccionario de aristas y ahora existen nuevas aristas con el nuevo nodo. Al terminar de crear aristas se eliminan los nodos elegidos.

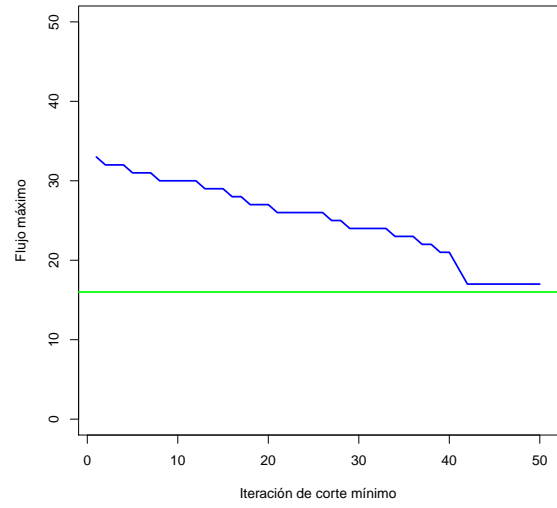
Al inicio del proceso se realiza la creación del grafo y se calcula el flujo máximo por medio del algoritmo de Ford-Fulkerson. Durante el proceso, cuando se realiza el corte mínimo, se tiene el resultado del flujo y se guarda en un archivo .csv, si el flujo que se va calculando es menor o igual que el anterior se guarda en el archivo, si es mayor, se vuelve a hacer el cálculo del flujo.

En la figura 1 se tiene una línea horizontal verde, que representa el flujo máximo calculado con Ford-Fulkerson, la curva azul representa los flujos que se fueron calculando al estar haciendo iteraciones para el corte mínimo. Se realizaron cinco repeticiones con cincuenta iteraciones de corte mínimo por repetición.

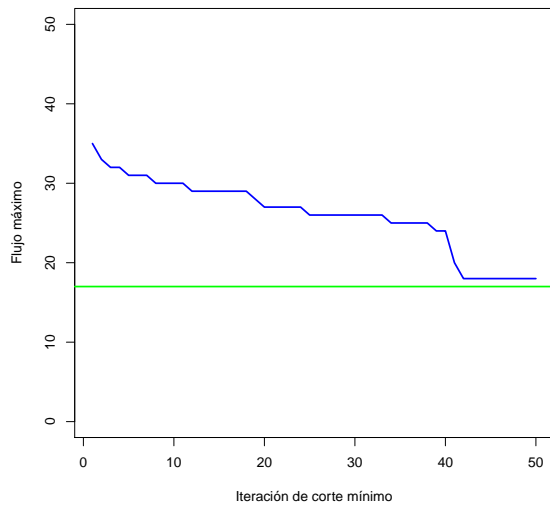
Como se puede observar, se tiene en un inicio que el flujo es muy grande con respecto al flujo máximo obtenido con el algoritmo de Ford-Fulkerson. Esto se debe a que el algoritmo de corte mínimo es una cota superior del algoritmo de Ford-Fulkerson. Existe una gran diferencia al inicio, pero conforme se van realizando las iteraciones del corte mínimo el flujo se va acercando cada vez más al valor del flujo máximo.



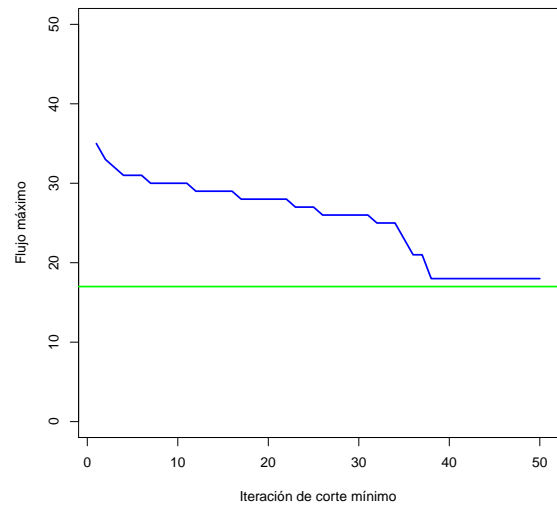
(a) Repetición 1.



(b) Repetición 2.



(c) Repetición 3.



(d) Repetición 4.

Figura 1: Flujo máximo realizando un corte mínimo.

3. Tiempos de ejecución

Los tiempos de ejecución fueron tomados en cuenta al realizar las cinco repeticiones y las cincuenta iteraciones que se obtuvieron del flujo máximo. Dado que es un proceso rápido los tiempos están medidos en segundos.

En la figura 2 se muestran las cinco repeticiones en donde las cajas de bigote tienen información sobre el tiempo que tardó en ejecutarse cada una de las cincuenta iteraciones.

Como se puede observar no existe un cambio tan notorio al realizar las repeticiones, sin embargo, existen datos atípicos en cada una de las repeticiones, que indica que existe o existen iteraciones que tardan mucho más en realizar el proceso que otras. Sin embargo, se tiene que los tiempos de ejecución del proceso de corte mínimo son pequeños, lo que es una buena señal.

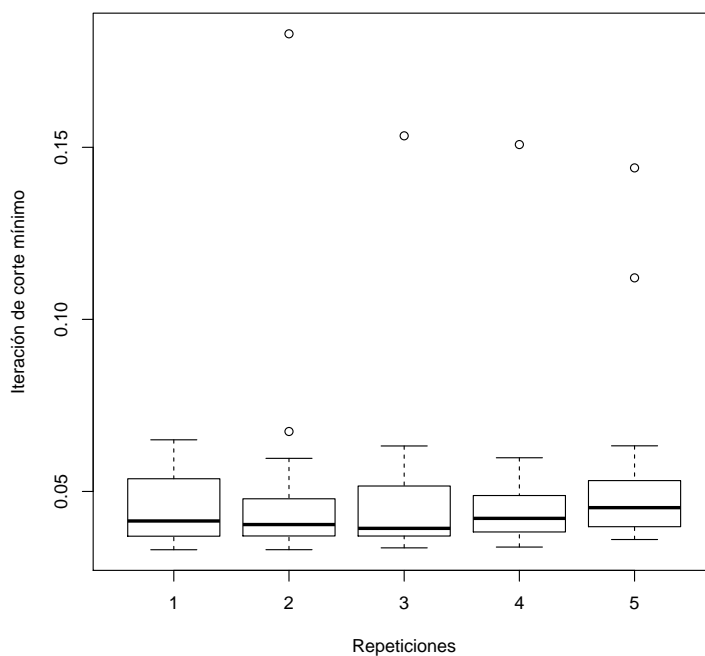


Figura 2: Tiempos de ejecución de cinco repeticiones realizando cincuenta iteraciones del corte mínimo.

Referencias

- [1] Beatriz García. Tarea 5. Optimización de flujo en redes. Mayo de 2018.
<https://github.com/BeatrizGarciaR/FlujoEnRedes/blob/master/Tarea5/Percolacion.py>