

Tarea 5

Optimización de flujo en redes

Beatriz Alejandra García Ramos

7 de mayo de 2018

1. Introducción

En esta práctica se tiene un análisis del algoritmo de Ford-Fulkerson que ya se ha trabajado en la tercer práctica [1]. Para realizar el análisis se tienen mallas en donde se realizan percolaciones [2] de nodos o de aristas y se lleva a cabo el algoritmo de Ford-Fulkerson para verificar si existe diferencia o no en el flujo máximo que se envía desde el nodo inicial hasta el nodo final.

Además de analizar el cambio en la cantidad de flujo que se tiene al percolar nodos o aristas se tiene también el estudio en el tiempo de ejecución al realizar estos cambios realizando variaciones en el tamaño del umbral [3] que define las conexiones de los nodos a sus vecinos dependiendo del umbral que se indique.

2. Creación del grafo en python

Para realizar una malla de las que se requieren en esta práctica se tiene un plano donde los ejes están dados por números enteros positivos. La malla es de tamaño $n \times n$, donde n es un número dado por el usuario y $n \times n$ representa la cantidad total de nodos que habrá en esa malla.

Una vez definida la cantidad total de nodos para esa malla se tiene que el primer nodo que se crea es el que tiene como coordenada $(1, 1)$ y el último en crearse es el que tiene como coordenada (n, n) , además los nodos intermedios se van creando hacia la derecha y hacia arriba.

Ya que los nodos son creados se realizan las conexiones dependiendo del umbral deseado. El umbral definirá cuáles serán los vecinos de los nodos teniendo en cuenta la distancia manhattan [4]. Si el umbral es igual a uno entonces los vecinos del nodo serán aquellos que están a una unidad en distancia manhattan, cuando se tenga umbral de dos los vecinos serán aquellos que estén a dos unidades en distancia manhattan, y así sucesivamente. De esta manera se realizan conexiones bidireccionales con capacidades que siguen una distribución normal con media igual a cuatro.

Ya que se tienen todas las conexiones dependientes del umbral se crean aristas donde un nodo se conecta con otro de manera aleatoria con una probabilidad de uno en cien, estas conexiones tienen dirección y tienen capacidades que siguen una distribución exponencial. Para que sean creadas correctamente se verifica que la conexión no sea alguna de las que ya existen al tomar en cuenta el umbral.

3. Percolación de nodos y conexiones

Cuando se tiene una malla existen diversos caminos que se pueden tomar desde el nodo inicial hasta el nodo final, lo cual ayuda a que se tenga un buen flujo al momento de realizar el algoritmo de Ford-Fulkerson.

Supongamos que la malla representa las tuberías que se tienen en una colonia, las cuales transportan el agua que llega a los hogares de las familias que viven ahí. Lo ideal sería que todas las tuberías funcionen a la perfección, sin embargo, en la vida real eso no sucede, existe la posibilidad de que

alguna tubería no funcione, que alguna de las conexiones entre una tubería y otra esté tapada, o ambas al mismo tiempo.

Como las situaciones mencionadas pueden llegar a suceder no podemos tener una malla perfecta para analizar el flujo máximo con el algoritmo de Ford-Fulkerson así que se realiza una percolación de nodos, los cuales representan las conexiones entre las tuberías, y las conexiones, que representan la tubería. Una vez realizados los cambios en las mallas se vuelve a utilizar el algoritmo y se nota una diferencia en el flujo transferido entre el nodo inicial y el nodo final.

3.1. Percolación de nodos

Para realizar percolación en nodos se tiene que crear una lista en donde se guarden aquellos nodos que se eliminan de los nodos originales, cada vez que se agrega a la lista el nodo percolado se recorre el diccionario que contiene las conexiones entre el nodo y sus vecinos y se eliminan las conexiones que existían, ya que esos caminos ya no están disponibles para el algoritmo de Ford-Fulkerson. Si las conexiones son bidireccionales se eliminan ambas direcciones, si es unidireccional se elimina la dirección, como se muestra en la siguiente parte del código.

```
for n in self.auxNodo:
    for m in self.nodos:
        if (n,m) in self.aristas:
            del self.aristas[(n,m)]
        if (m,n) in self.aristas:
            del self.aristas[(m,n)]
```

En la figura 1 se puede observar cómo en una malla de tamaño 8×8 con un umbral igual a uno se realiza la percolación de algunos nodos y cómo se desvanecen también las aristas que acompañan a dichos nodos.

En la figura existe aún camino para llegar del nodo inicial, marcado en color naranja, y el nodo final, marcado en color rojo, por lo que se tiene aún un flujo positivo al momento de realizar el algoritmo de Ford-Fulkerson. Puede suceder, como se muestra, que exista algún nodo que no fue percolado pero que no tiene conexiones con otros nodos ya que el umbral no permite que logre conectarse con los nodos disponibles. También puede suceder que ya no exista camino entre el nodo inicial y el nodo final.

Para analizar el cambio que se genera en el algoritmo de Ford-Fulkerson al momento de percolar nodos se tienen variaciones en el tamaño del umbral y además se tienen iteraciones en donde cada vez que ocurre una iteración se va percolando un nodo.

Para lograr ver una diferencia entre un flujo y otro se realizan repeticiones al quitar cada nodo y se tienen cajas de bigote que incluyen las repeticiones obtenidas.

Como se puede ver en la figura 2, se tienen diferencias en las repeticiones aunque se haya percolado la misma cantidad de nodos, esto sucede por el hecho de que el nodo eliminado no es el mismo por cada repetición y las capacidades de las conexiones puede variar de tal manera que aumente o disminuya el flujo máximo, sin embargo, se tiene una visualización de la disminución de flujo al ir percolando los nodos, por lo que se tiene un efecto descendiente cada vez que se van eliminando nodos y conexiones en la malla y es muy notable la diferencia de flujos entre un umbral y otro.

3.2. Percolación de conexiones

Para realizar la percolación de conexiones se tiene que la eliminación se va realizando de veinte conexiones por iteración, dado que el número de conexiones es muy elevada. Cuando se eliminan las conexiones no se deben eliminar los nodos involucrados ya que los nodos pueden tener conexiones con algunos otros vecinos. De la misma manera que en la percolación de nodos se debe tener una lista en la cual se agregan las conexiones que se han eliminado para que no existan repeticiones de eliminación ya que no se puede eliminar lo ya eliminado, esto se realiza como se tiene a continuación.

```
if arista not in self.lqq:
    self.lqq.append(arista)
del self.aristas[arista]
```

En la figura 3 se muestra cómo se tiene una malla con percolaciones de aristas y en este caso ya no se encuentra un camino para llegar del nodo inicial al nodo final por lo que no se tiene un flujo al momento de realizar el algoritmo de Ford-Fulkerson.

Para poder analizar la percolación de conexiones se tiene que se eliminan veinte conexiones diez veces, se varía el umbral para ver la diferencia entre flujos de un umbral pequeño a uno más grande y se hacen repeticiones para cada eliminación de veinte conexiones.

Como se muestra en la figura 4 el flujo máximo va aumentando a medida que se incrementa el umbral, pero cada vez que se realizan percolaciones de conexiones se va perdiendo flujo a través de las conexiones restantes, en total se realiza percolación de doscientas conexiones, pero en comparación con la percolación de nodos, al percolar conexiones no se tiene una disminución drástica entre un flujo máximo y otro.

4. Análisis en el tiempo de ejecución

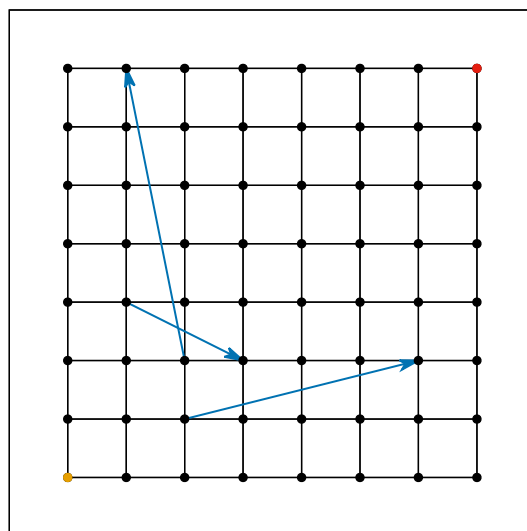
Al momento de realizar las percolaciones de nodos y conexiones se lleva a cabo un tiempo de ejecución mayor que si se tiene solamente la malla original ya que el algoritmo de Ford-Fulkerson es llevado a cabo cada vez que se percola un nodo o que se percolan conexiones.

Como se muestra en la figura 5, se tiene el tiempo de ejecución en cajas bigote cuando se realizan diez repeticiones para realizar percolaciones de nodos al realizar variaciones en el tamaño del umbral. Conforme aumenta el tamaño del umbral aumenta también el tiempo de ejecución dado que existen muchas más aristas entre un tamaño de umbral y otro.

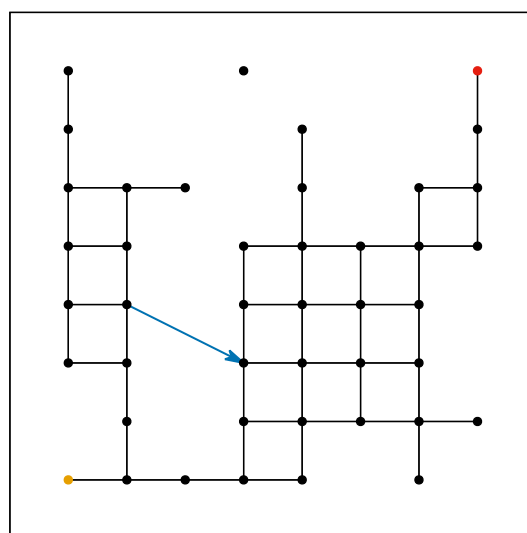
En la figura 6 se tienen las mismas variaciones de umbral y se calcula el tiempo de ejecución en segundos de la percolación de conexiones con sus repeticiones. Es notable que el tiempo de ejecución es mejor que en la percolación de nodos, esto se debe a que en la percolación de nodos se realizan muchas más iteraciones que en la percolación de conexiones. Sin embargo, se puede observar que ambos tiempos de ejecución tienen el mismo comportamiento al aumentar el umbral. Mientras más conexiones se tengan más tiempo tardará en procesarse el algoritmo de Ford-Fulkerson, como ya se había mencionado en la tercer práctica.

Referencias

- [1] Beatriz García. Tarea 3. Optimización de flujo en redes. Abril de 2018.
<https://github.com/BeatrizGarciaR/FlujoEnRedes/tree/master/Tarea3>
- [2] Percolación en el plano con simulaciones. Laura Esvala. Mayo de 2010.
<http://tikhonov.fciencias.unam.mx/presentaciones/2010may27.pdf>
- [3] Significado de umbral. Octubre de 2013.
<https://www.significados.com/umbral/>
- [4] Las distancias en Manhattan. Septiembre de 2017.
<https://metode.es/revistas-metode/secciones/cajon-de-ciencia/les-distancias-a-manhattan.html>

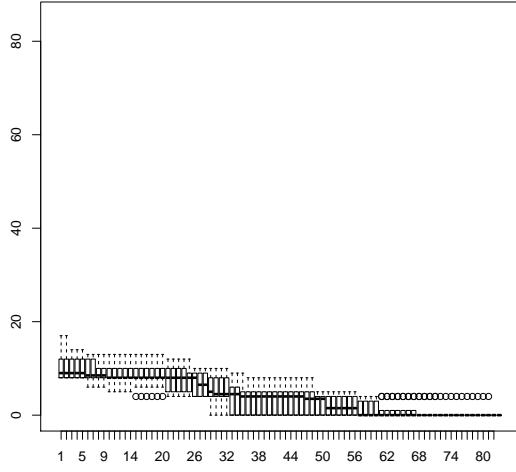


(a) Original.

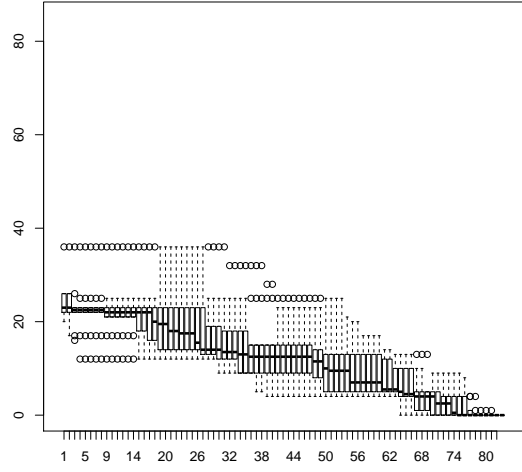


(b) Con percolación.

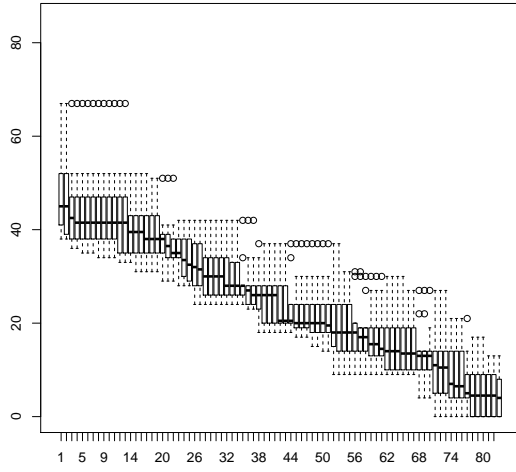
Figura 1: Percolación de nodos para un grafo de 8×8 .



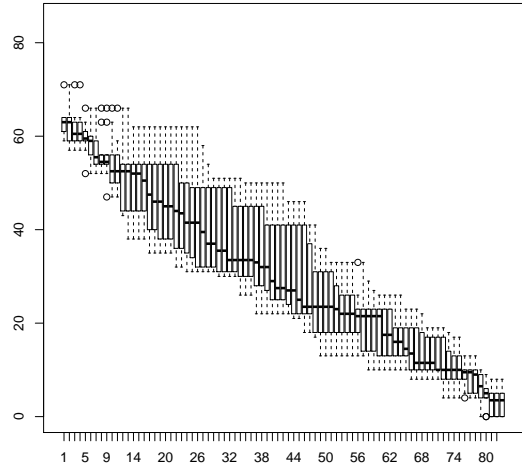
(a) Umbral igual a uno.



(b) Umbral igual a dos.

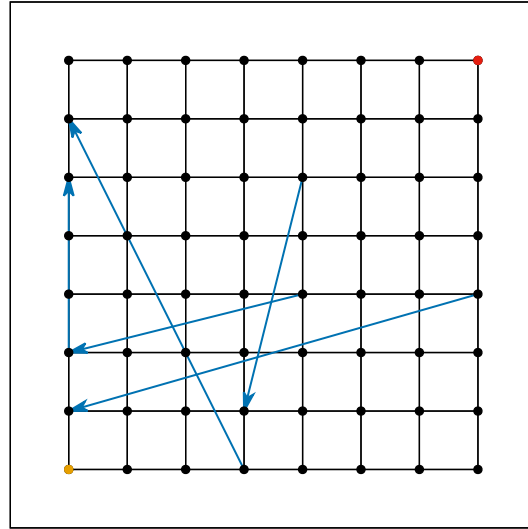


(c) Umbral igual a tres.

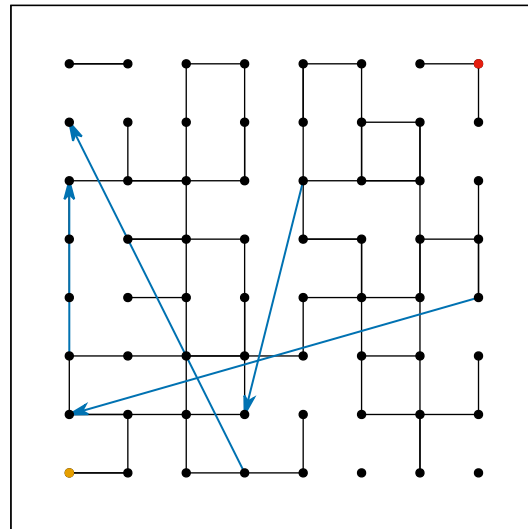


(d) Umbral igual a cuatro.

Figura 2: Flujo máximo percolando nodos haciendo variación en el umbral de conexiones entre nodos.

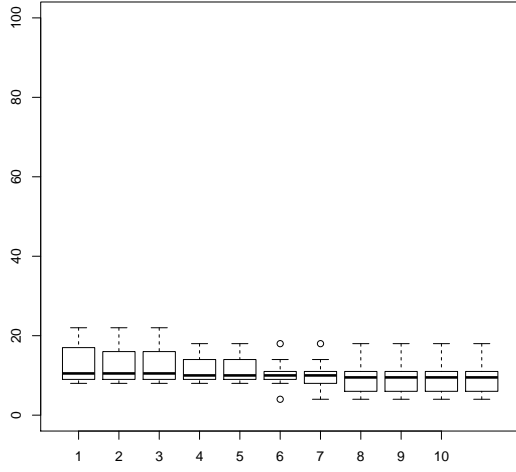


(a) Original.

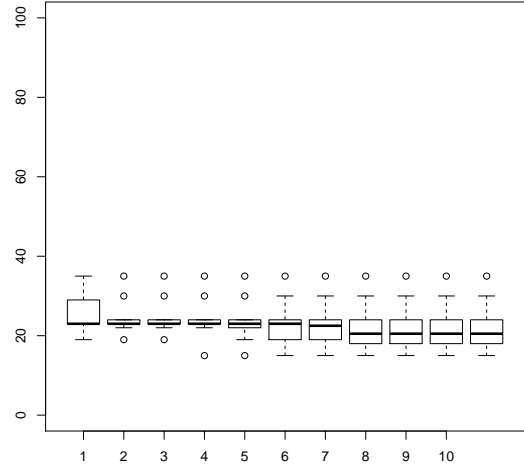


(b) Con percolación.

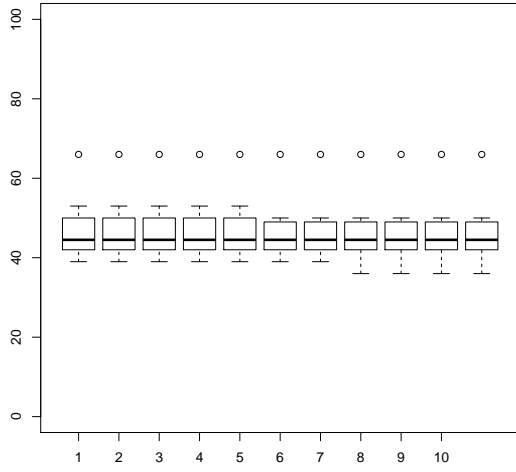
Figura 3: Percolación de aristas para un grafo de 8×8 .



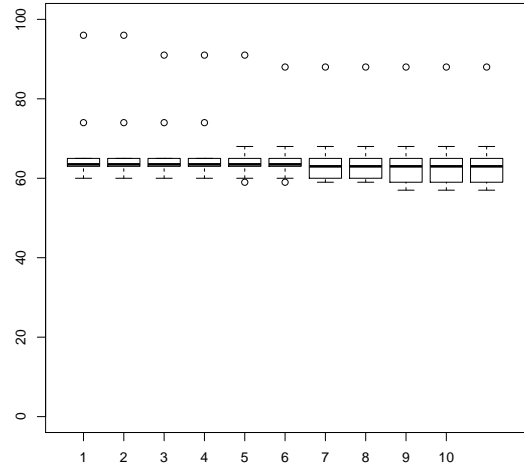
(a) Umbral igual a uno.



(b) Umbral igual a dos.



(c) Umbral igual a tres.



(d) Umbral igual a cuatro.

Figura 4: Flujo máximo percolando aristas haciendo variación en el umbral de conexiones entre nodos.

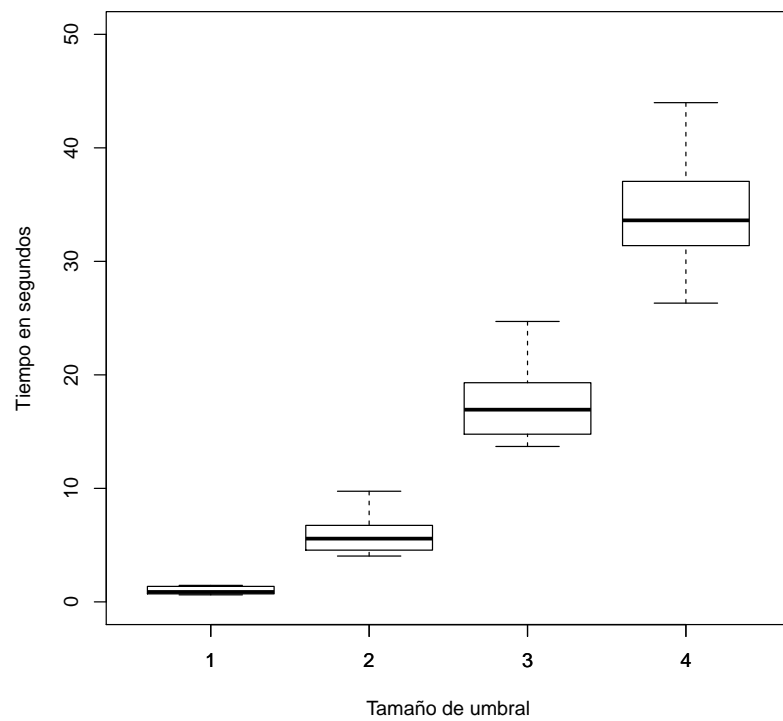


Figura 5: Tiempos de ejecución con variaciones en el umbral percolando nodos.

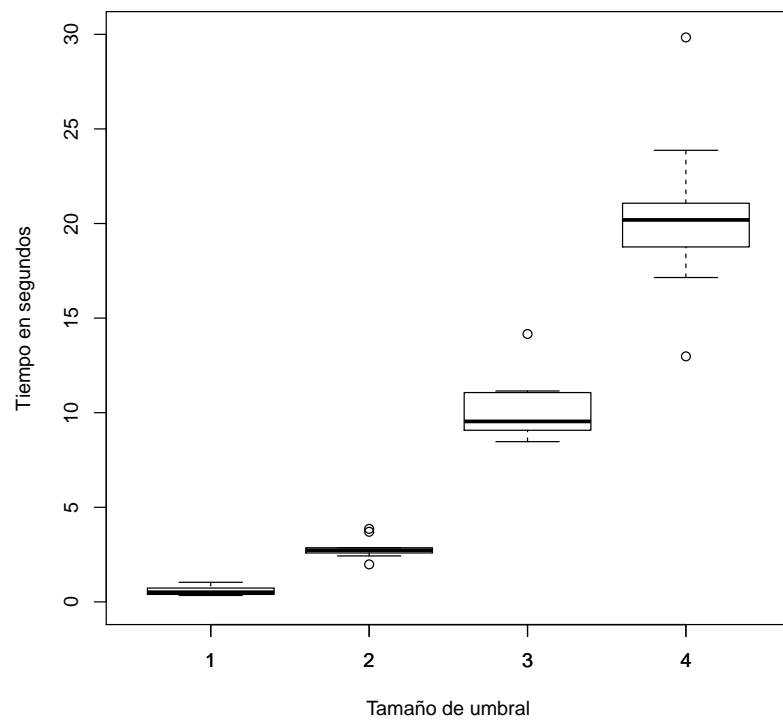


Figura 6: Tiempos de ejecución con variaciones en el umbral percolando aristas.