Tarea 6
Optimización de flujo en redes

Victor Aram Dominguez Ramirez

21 de Mayo de 2018

1. Introducción

En esta práctica calcularemos el flujo máximo de un grafo con cierta cantidad de nodos. Anteriormente en prácticas pasadas se calculo el flujo máximo mediante el algoritmo de Ford-Fulkerson. En está ocasión lo haremos percolando nodos y aristas a tal grado de dejar 2 nodos conectados mediante una arista que tendrá su flujo máximo. En dicha percolación se harán uniones de nodos haciendo este nuevo nodo en una fución de nodos.

2. Flujo máximo

Teniendo los n nodos del grafo ya conectados con otros nodos, ubicaremos estas aristas de nodo a nodo con su peso y al obtener el peso más pequeño de las aristas, dichos nodos que están conectados se unirán en un nuevo nodo que se ubicara en su punto medio.

```
for (x1,y1) in self.nodos:
    for (x2,y2) in self.nodos:
        q= (x1,y1,x2,y2)in self.doc
        if q == True:
            gh=self.doc[(x1,y1,x2,y2)]
            if gh<h:
                h=gh
                A1=x1
                B1=y1
                A2=x2
                B2=y2
```

Ubicaremos las coordenadas de los puntos a unir para la fusión del nuevo nodo, con el objetivo de borrar la coordenada que las unía y encontrar las aristas que unían a esos dos nodos con el restante de nodos. Al tener las coordenadas que se unían a estos dos nodos, se borraran y crearan una solo una arista por cada nodo, sumando los pesos que estos tenían hacia cierto nodo. En pocas palabras, si el nodo A se unirá con el nodo B creando la fusión de un nuevo nodo AB, si la arista del nodo A con el nodo C tiene un peso de 3 y de B a C es de 5, entonces ahora habrá un nodo AB que conectara con C con una sola arista donde se sumaran dichos pesos de las aristas siendo igual a 8. Este proceso se hará periódicamente hasta que solo existan dos nodos en el grafo con una conexión y esta tendrá el flujo máximo.

```
for (x3,y3) in self.nodos:
    f1=(A1+A2)/2
    f2=(B1+B2)/2
    k1=(A1,B1,x3,y3)in self.doc
    k2=(x3,y3,A1,B1)in self.doc
    if k1==True:
```

```
            r1=self.doc[(A1,B1,x3,y3)]
            del self.doc[(A1,B1,x3,y3)]
        else:
            r1=0
        if k2==True:
            r3=self.doc[(x3,y3,A1,B1)]
            del self.doc[(x3,y3,A1,B1)]
        else:
            r3=0
        R=r1+r3
        v1=(A2,B2,x3,y3)in self.doc
        v2=(x3,y3,A2,B2)in self.doc
        if v1==True:
            c1=self.doc[(A2,B2,x3,y3)]
            del self.doc[(A2,B2,x3,y3)]
        else:
            c1=0
        if v2==True:
            c2=self.doc[(x3,y3,A2,B2)]
            del self.doc[(x3,y3,A2,B2)]
        else:
            c2=0
        C=c1+c2
        Q=R+C
        if Q!=0:
            self.doc.setdefault((f1,f2,x3,y3),Q)
    self.nodos.remove((A1,B1))
    self.nodos.remove((A2,B2))
```

Podemos decir que al eliminar siempre la arista con menos peso, vamos a ir dejando los flujos mayores para así obtener el máximo flujo como resultado como se muestra en la figura 1. Se obtuvo un flujo máximo de 21 unidades de una red de 7 nodos.

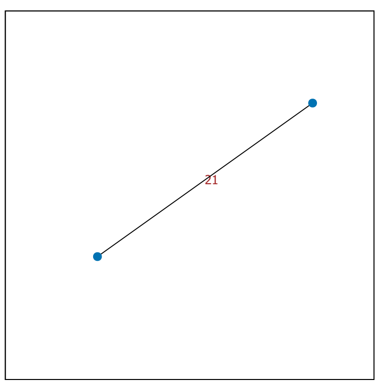

Figura 1: Flujo máximo.

3. Resultados

Con forme aumentamos los nodos el flujo va a ir creciendo como se muestra en la figura 2. Se puede considerar que a comparación del algoritmo de Ford-Fulkerson este método es mucho más rápido, pero menos eficiente. 1.

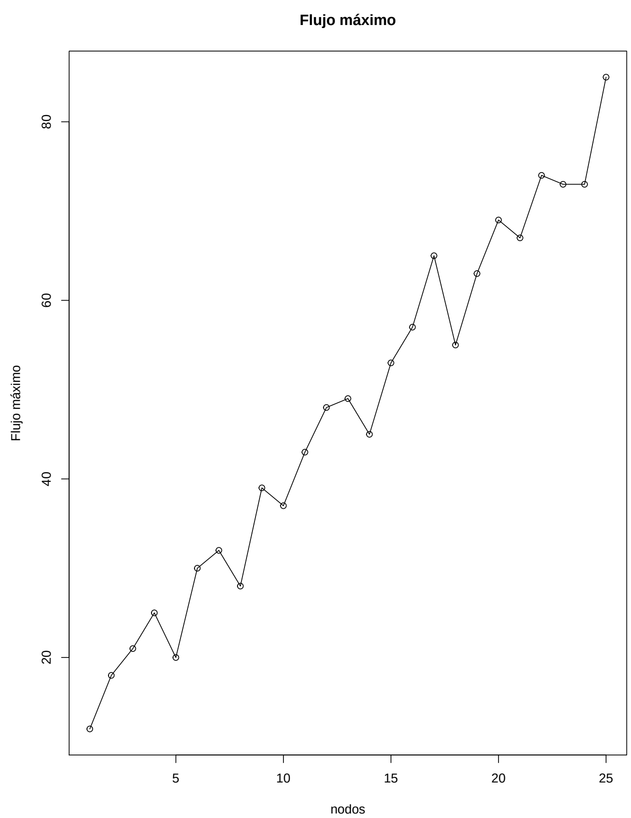

Figura 2: Crecimiento.