Tarea 6
Optimizacion de Flujo de Redes

L. A. Gutierrez

21 de mayo de 2018

Introducción

Para ésta actividad utilizamos el lenguaje de programación *Python* [1] y para realizar las gráficas se usp *GNUplot* [2] y se partió de una versión anterior del programa sobre la distancia Manhattan [3], y se le agregó la funcionalidad de seleccionar un arista aleatoria, eliminar esa arista y unir los vértices que vinculaba ese arista, después se creaba un nuevo vértice con el nombre de los dos vértices eliminados, se tomaban las conexiones de los vértices eliminados y se le daban al nuevo vértice. Quise comprobar si nuestra heurística de contracción de aristas nos entregaba, o se aproximaba a, el valor del flujo máximo que pasaba por aquel grafo. El valor del flujo máximo de cada grafo fue calculado con el algoritmo de Ford-Fulkerson

Contracción de aristas

Una contracción de aristas[4] es una operación que elimina una arista del grafo al mismo tiempo que fusiona los dos vértices extremos. La contracción es una operación fundamental en la teoría de grafos. La operación de contracción de aristas toma un arista $e = \{i, j\}$, la cual es removida del grafo y los dos vértices incidentes i y j son fusionados en un nuevo vértice k, de forma que las aristas que se conectarán a k son las aristas que tenían de i y j

Ford-Fulkerson

El algoritmo de Ford-Fulkerson[5] propone buscar caminos en los que se pueda aumentar el flujo, hasta que se alcance el flujo máximo. La idea es encontrar una ruta con un flujo positivo neto que una los vértices origen y destino. Su nombre viene dado por sus creadores, L. R. Ford, Jr. y D. R. Fulkerson.

Pseudocódigo de Ford-Fulkerson

```
Ford-Fulkerson(G,s,t) {
   for (cada arista (u,v) de E) {
       f[u,v]= 0;
       f[v,u]= 0;
   }
   while (exista un camino p desde s a t en la red residual Gf) {
       cf(p) = min{cf(u,v): (u,v) está sobre p};
       for (cada arista (u,v) en p) {
           f[u,v]= f[u,v] + cf(p);
           f[v,u]= - f[u,v];
       }
   }
}
```

Método

En la actividad se realizaron entre cinco y quince iteraciones y consistió en dos etapas:

Segmento de generación de grafo y obtención flujo máximo real:

1.- Generar un grafo aleatorio.

2.- Determinar dos vértices, el primero seria nuestro origen y el segundo nuestro sumidero, los cuales llamaremos s y t respectivamente.

3.- Calcular el flujo máximo del grafo desde s a t con el algoritmo Ford-Fulkerson.

Segmento de heurística de contracción de aristas:

1.- Seleccionar de manera aleatoria una arista.

2.- Crear un nuevo vértice k entre los dos vértices del arista seleccionada, los llamaremos i y j respectivamente.

3.- Reasignar las aristas de los vértices i y j al vértice k.

4.- Eliminar los vértices i y j

5.- Repetir pasos del 1 al 4 hasta que solo queden dos vértices.

Así el flujo máximo que obtendremos a través de la contracción seria la suma del flujo de todas las aristas entre los dos "mega-vértices"

Resultados

Para obtener los resultados creé un archivo *"main.py"* donde mande a crear un archivo con extensión dat, donde almacené tres valores para graficar, (i)la cantidad de iteraciones, (ii)el calculo del flujo máximo según la heurística de contracción de aristas, y (iii)el tiempo promedio de las iteraciones, es decir, se sumaba el tiempo de ejecución de todas las iteraciones y se dividía entre la cantidad de iteraciones.

En la Tabla 1 se puede observar los resultados obtenidos.

Instancias	Flujo Máximo	Tiempo de Ejecucion(s)
5	25.3586	1.4352700094
6	21.4711	2.0151922985
7	17.9656	2.5978837541
8	14.9831	3.1507142047
9	11.7471	3.7281362636
10	9.1085	4.2806541556
11	6.8798	4.872371086
12	4.918	5.6480084986
13	3.325	6.0235371668
14	2.0001	6.6164945402
15	1.021	7.2406491422

Tabla 1: Resultados.dat[6]

En la siguientes graficas se observa el comportamiento del flujo en relación a la cantidad de iteraciones y la segunda grafica muestra la relacion entre el tiempo de ejecución y la cantidad de iteraciones.

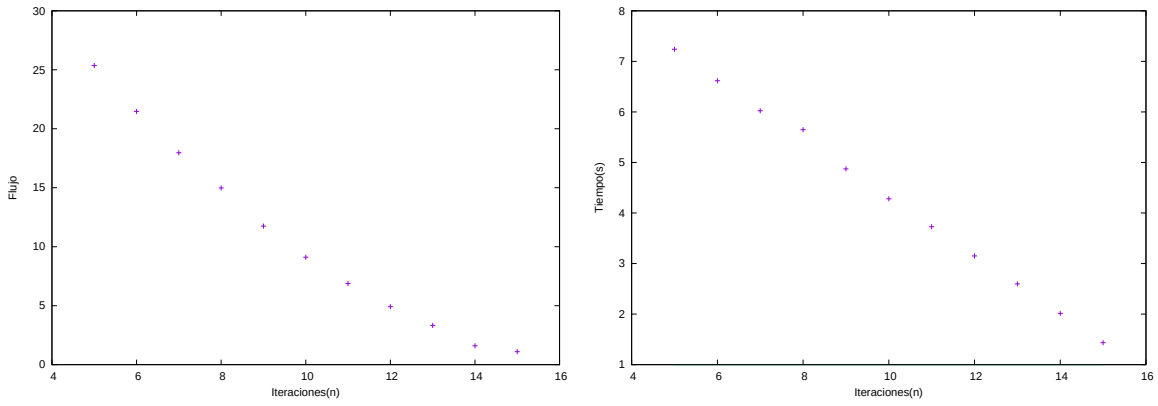

Figura 1: A la izquierda la gráfica Flujo-Iteraciones, a la derecha la gráfica Tiempo-Iteraciones

Referencias

[1] *Python versión 3* https://www.python.org/

[2] *GNUPlot versión 5* http://www.gnuplot.info/

[3] *Tarea5.pdf* https://github.com/SamatarouKami/OPTIMIZACION-DE-FLUJOS-DE-REDES/tree/master/Tarea5

[4] *Contracción de Aristas* https://es.wikipedia.org/wiki/Contracci%C3%B3n_de_aristas

[5] *Algoritmo Ford-Fulkerson* https://es.wikipedia.org/wiki/Algoritmo_de_Ford-Fulkerson

[6] *Resultados.dat* https://github.com/SamatarouKami/OPTIMIZACION-DE-FLUJOS-DE-REDES/tree/master/Tarea6/Resultados.dat