8.5

Tarea 5

Jesus Angel Patlán Castillo (5261)

30 de abril de 2019

En esta tarea se analizan las características de grafos que son generados por un algoritmo de generación de grafos proporcionado por librerías de Python, además de estudiar como estas características afectan el tiempo de ejecución al momento de obtener el flujo máximo entre dos nodos dentro del grafo. Los algoritmos se obtuvieron por medio de la librería NetworkX [3] de Python [1], la librería Matplotlib [4] es utilizada para generar las gráfica de correlación entre los efectos estudiados. El código empleado se obtuvo consultando la documentación oficial de la librería NetworkX [2]. Las imágenes y el código se encuentran disponibles directamente en mi repositorio [5].

1. Metodología

Basándonos en las tareas anteriores, se decide utilizar de algoritmo de generador de grafos de llamado grafo de cavernícola conexo, el cual tiene una mayor aplicación para el problema de flujo máximo dado que es posible representar cada cluster de nodos como una isla en la que se necesita transportar recursos a otro cluster. Además, tenemos que el algoritmo de flujo máximo que utilizaremos será el tradicional flujo máximo entre dos nodos, dado que fue el algoritmo que mostró un mejor desempeño en tiempo de ejecución. Para una mejor visualización se eligió el algoritmo de acomodo resorte, el cual da una mejor visualización de los nodos y las aristas del grafo. Para las pruebas, se crearon 5 instancias distintas de grafo de cavernícola conexo, los cuales se muestran a continuación:

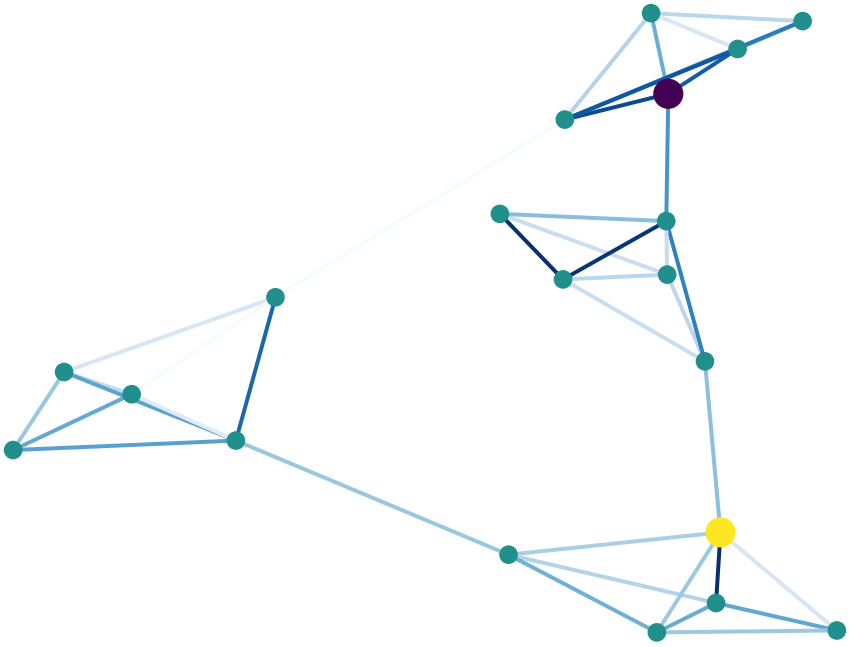

Figura 1: Grafo 1

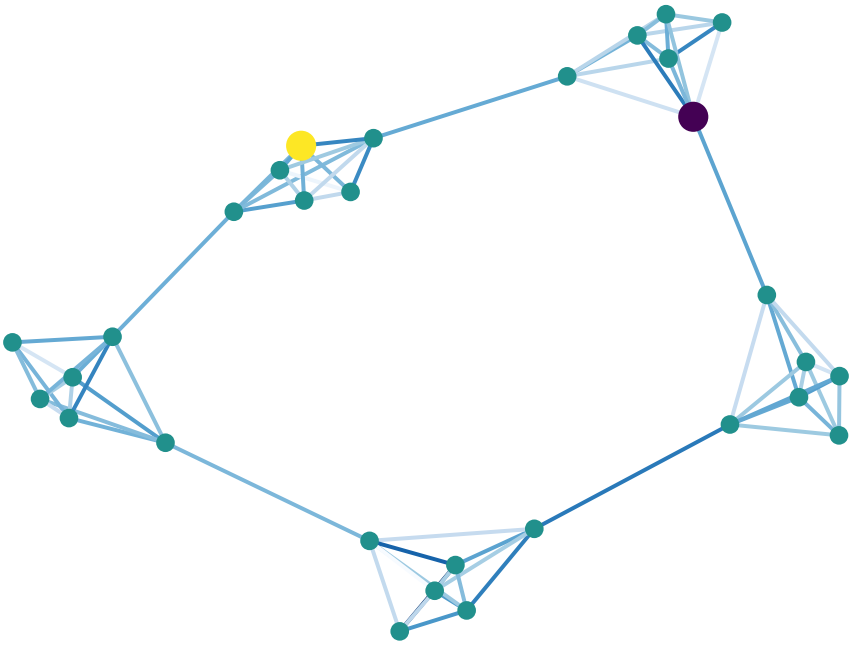

Figura 2: Grafo 2

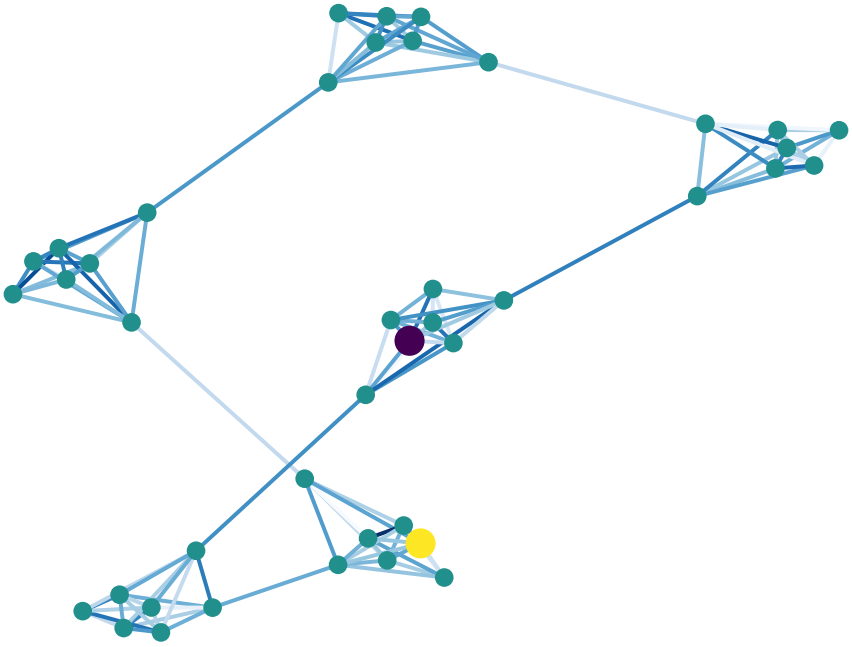

Figura 3: Grafo 3

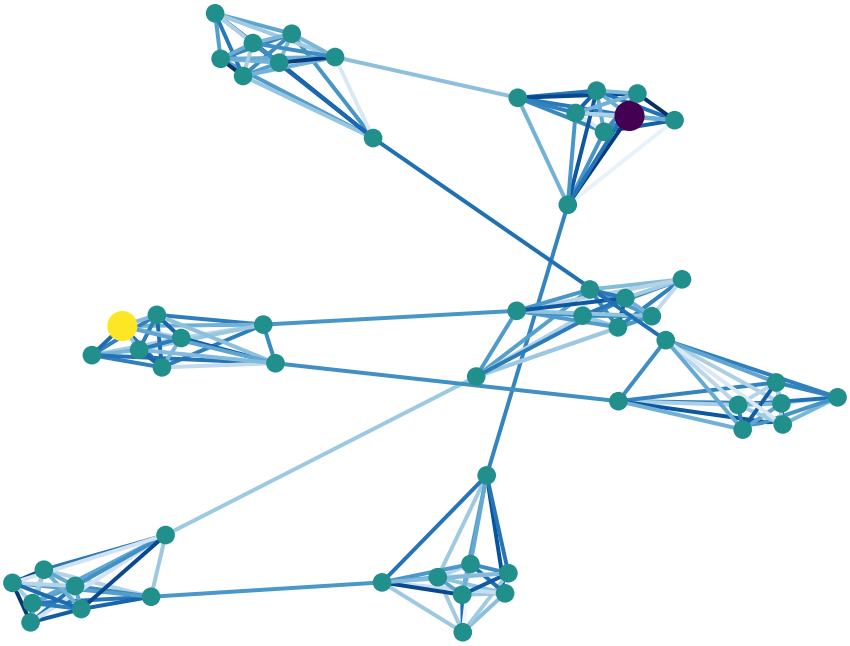

Figura 4: Grafo 4

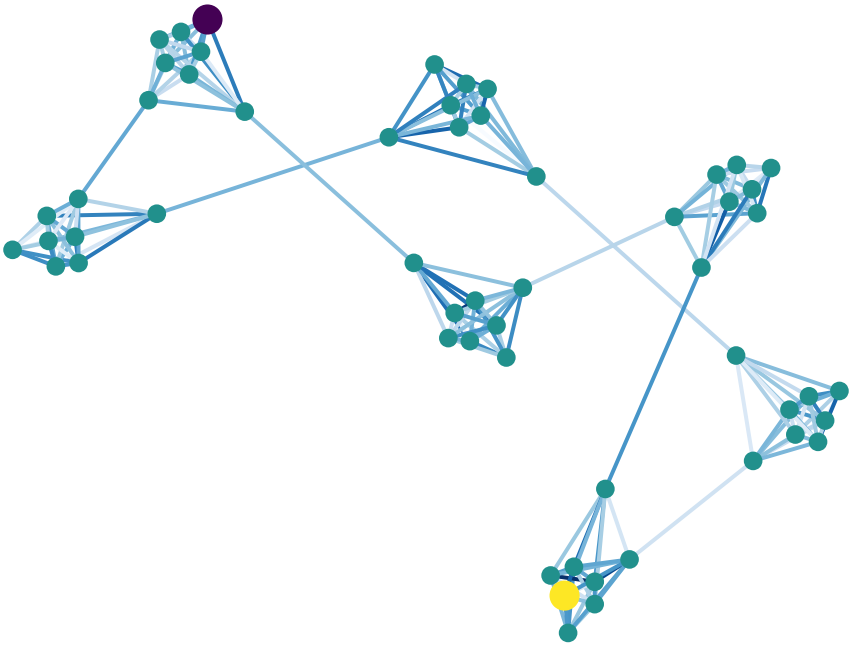

Figura 5: Grafo 5

Para cada grafo, señalamos un nodo fuente (en amarillo) y un nodo sumidero (en negro) de ejemplo para el problema de flujo máximo, además que resaltamos cada arista con un tono de azul más oscuro cuanto mayor sea el peso que tenga. Para cada uno de los grafos, se realizaron distintas pruebas intercambiando los nodos fuente-sumidero para determinar el rendimiento del algoritmo de flujo máximo. Además, a cada grafo se le determinan 5 características:

- Distribución de grado: Es el número de aristas asociadas a un nodo.
- Coeficiente de agrupamiento: Determina que tanto un nodo está asociado a sus nodos vecinos.
- Centralidad de cercanía: Determina qué tan cercano o alejado esta un nodo con respecto a los demás nodos vecinos.
- Centralidad de carga: Determina el número de caminos más cortos que se pasan a través del nodo.
- Excentricidad: Determina la longitud máxima que hay entre el nodo y otro nodo del grafo.
- PageRank: Similar a la distribución de grado, pero da distinto pesos a las aristas de acuerdo al grado que tenga el nodo al que se conecta.

Para la obtención de estas características y la generación de los grafos, se utilizo el siguiente código:

```
def graph_attributes(G):
    global num_fig
    nodes=G.nodes()
    degrees=[G.degree(i) for i in nodes]
    clustering=[nx.clustering(G,i) for i in nodes]
    closeness=[nx.closeness_centrality(G,i) for i in nodes]
    load=[nx.load_centrality(G,i) for i in nodes]
    eccentricity=[nx.eccentricity(G,i) for i in nodes]
    pageRank=nx.pagerank(G,weight="capacity")
    pageRank_G=[pageRank[i] for i in nodes]
    plt.figure(0)
    plt.ylabel('Replicas')
    plt.xlabel('Grado del nodo')
    plt.hist(degrees)
    plt.figure(1)
    plt.savefig('hist-grado-'+str(num_fig)+'.eps', format='eps',
    dpi=1000)
    #plt.clf()
    plt.ylabel('Replicas')
    plt.xlabel('Coeficiente de agrupamiento del nodo')
    plt.hist(clustering)
    plt.figure(2)
    plt.savefig('hist-agrupamiento-' + str(num_fig) + '.eps',
    format='eps', dpi=1000)
    plt.clf()
    plt.ylabel('Replicas')
    plt.xlabel('Coeficiente de cercania del nodo')
    plt.hist(closeness)
    plt.figure(3)
    plt.savefig('hist-cercania-' + str(num_fig) + '.eps', format='
    eps', dpi=1000)
    plt.clf()
    plt.ylabel('Replicas')
    plt.xlabel('Centralidad del nodo')
    plt.hist(load)
    plt.figure(4)
    plt.savefig('hist-centralidad-' + str(num_fig) + '.eps', format
    ='eps', dpi=1000)
    plt.clf()
    plt.ylabel('Replicas')
    plt.xlabel('Excentricidad del nodo')
    plt.hist(eccentricity)
    plt.figure(5)
    plt.savefig('hist-excentricidad-' + str(num_fig) + '.eps',
    format='eps', dpi=1000)
    plt.clf()
    plt.ylabel('Replicas')
    plt.xlabel('PageRank del nodo')
    plt.hist(pageRank_G)
    plt.figure(6)
    plt.savefig('hist-pagerank-' + str(num_fig) + '.eps', format='
    eps', dpi=1000)
    plt.clf()
```

En base a estas 5 características, se realiza una prueba ANOVA para determi-

nar como influyen estos en el tiempo de ejecución del algoritmo. Un ejemplo aplicando el flujo máximo en cada grafo se muestra a continuación, donde se muestran solamente las aristas por donde el flujo se distribuye:

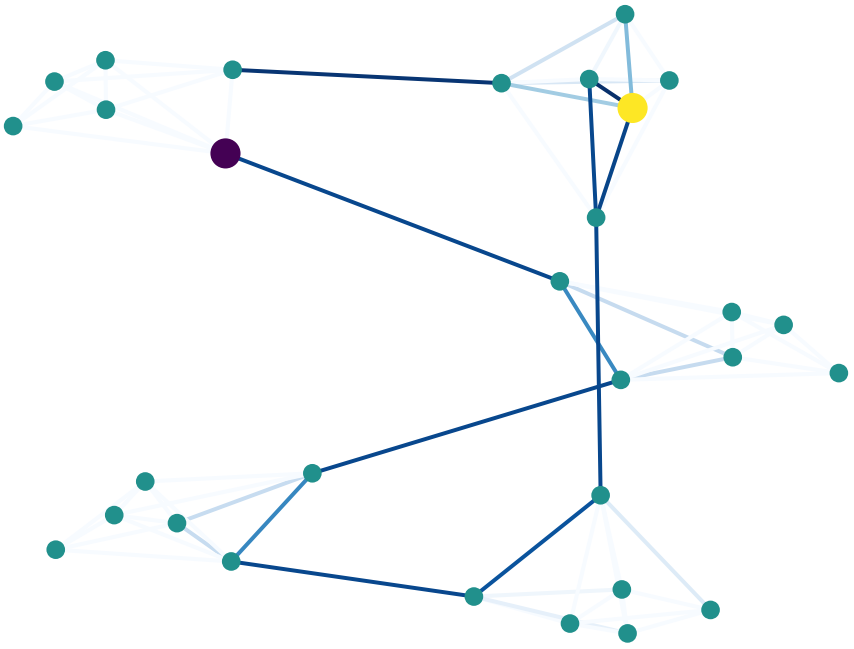

Figura 6: Grafo 1

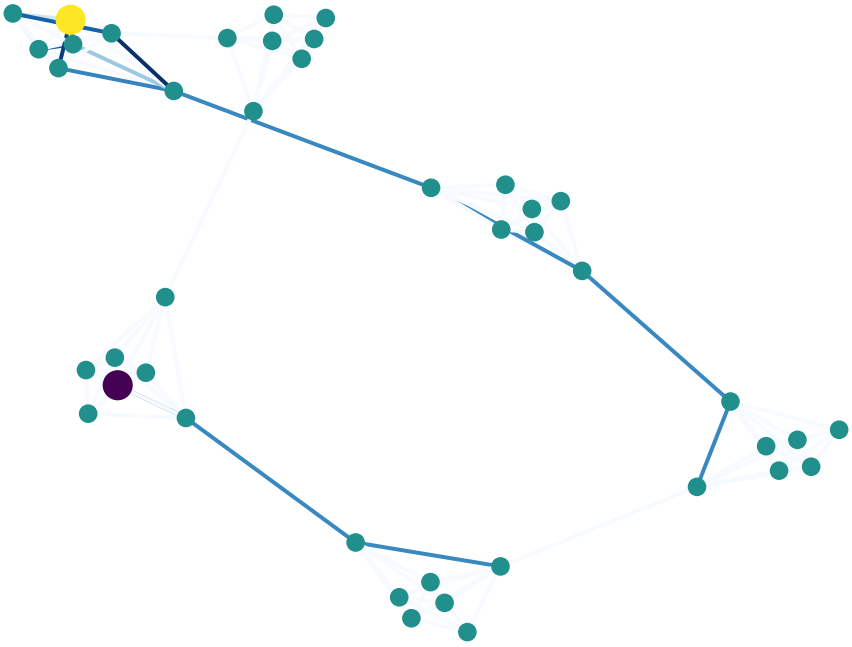

Figura 7: Grafo 2

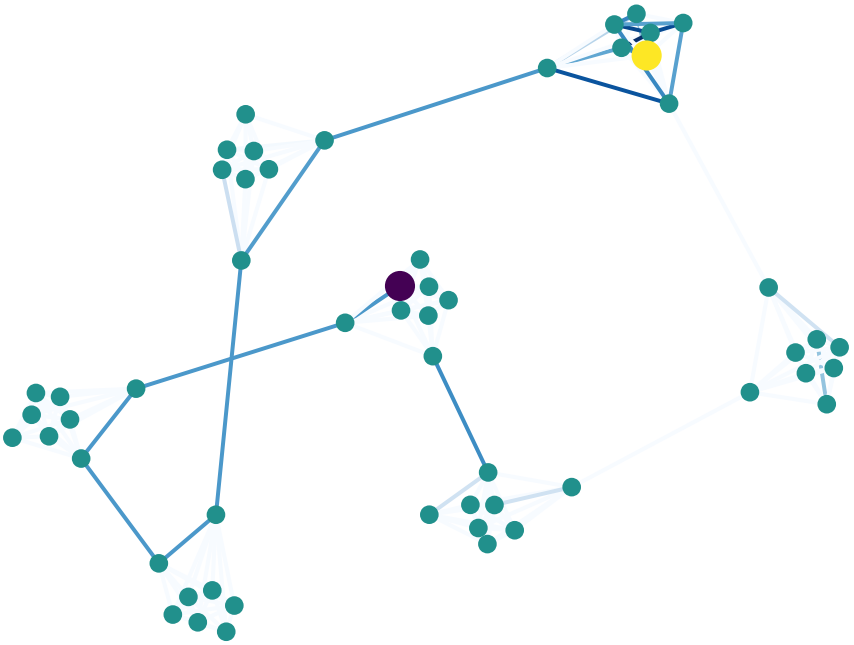

Figura 8: Grafo 3

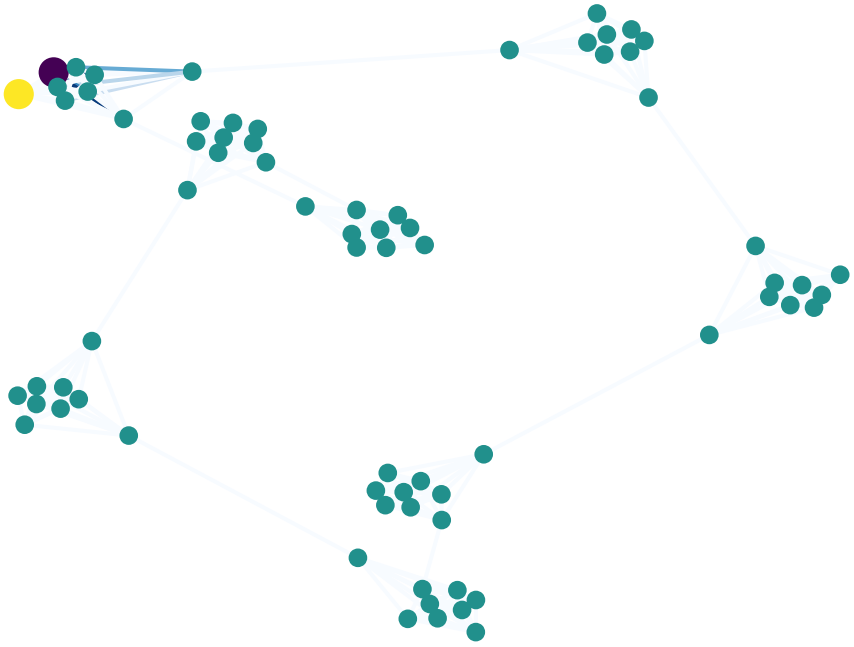

Figura 9: Grafo 4

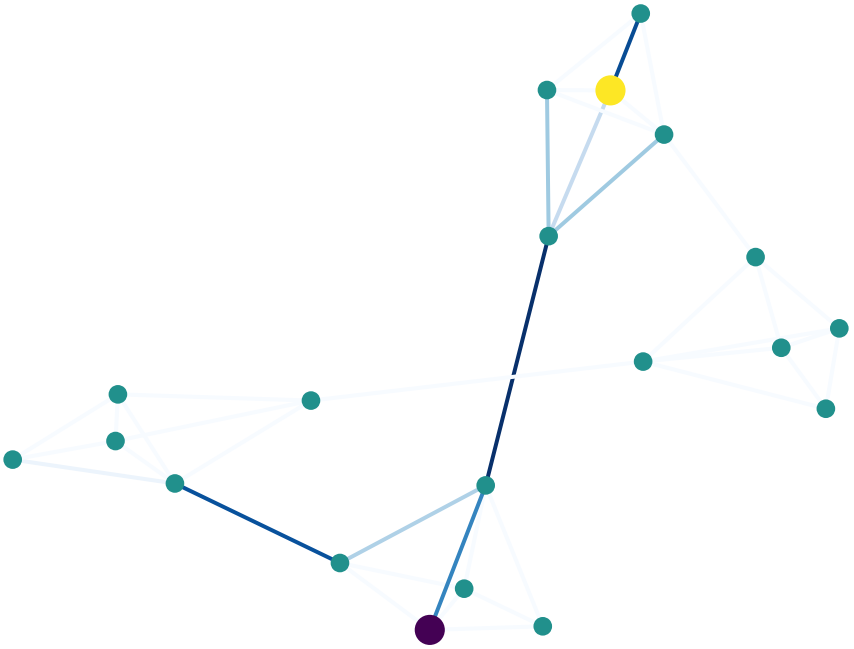

Figura 10: Grafo 5

Este proceso se realizó en una laptop con las siguientes características:

- Procesador: Intel Core i7-7500U 2.7GHz
- Memoria RAM: 16GB
- Sistema Operativo: Windows 10 64 bits

Los resultados fueron determinados por medio de un ANOVA utilizando el siguiente código:

```
dataframe = pd.DataFrame(total_data)
dataframe.to_csv('datos.xls',sep='\t')
model = ols('Tiempo~G*A*C*Ce*E*P',data=dataframe).fit()
anova = anova_lm(model,typ=2)
anova.to_csv('anova1.xls',sep='\t')
```

2. Resultados

Estas son las características encontradas de las instancias de grafos:

2.1. Grado de nodos

A continuación se muestran histogramas relacionadas a los grados de los nodos de cada grafo.

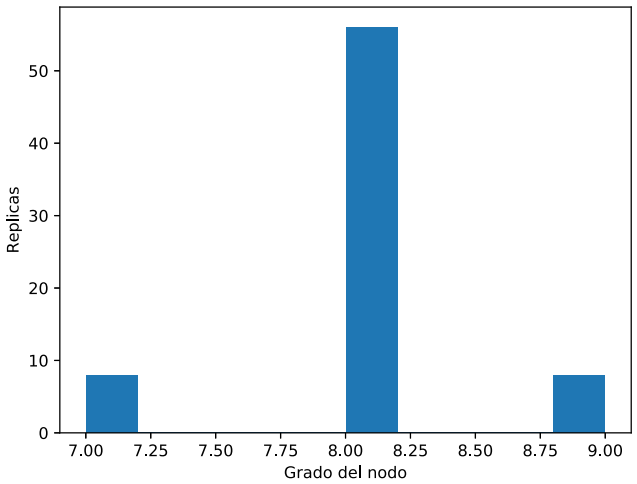

Figura 11: Grafo 1

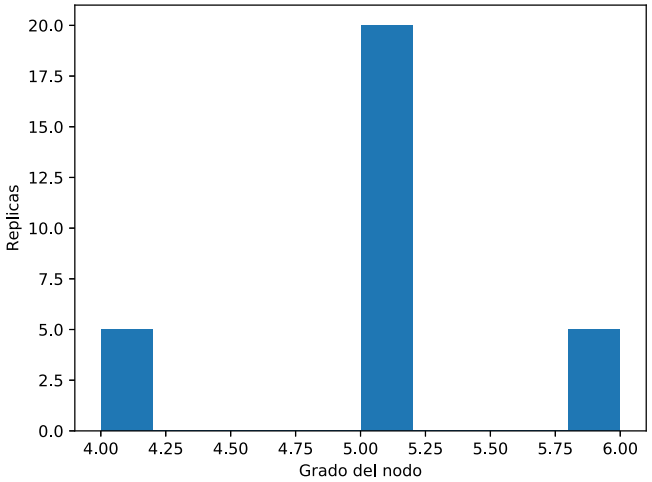

Figura 12: Grafo 2

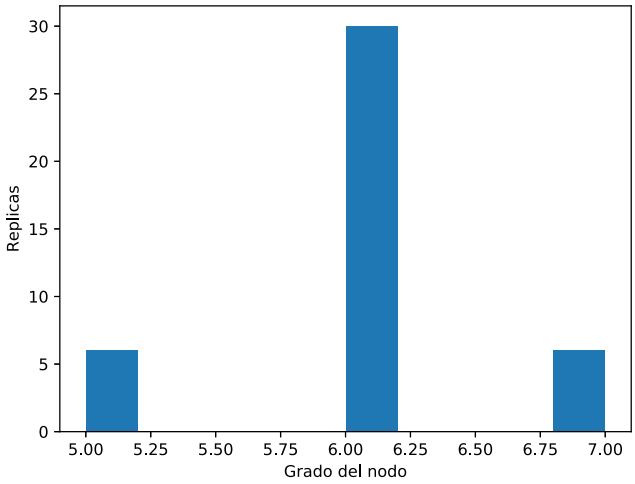

Figura 13: Grafo 3

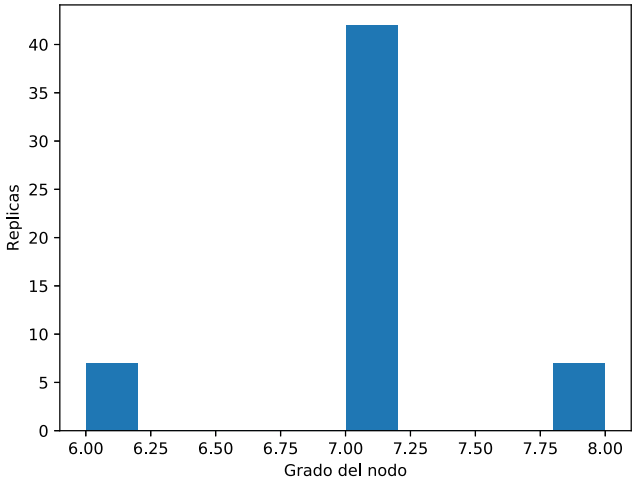

Figura 14: Grafo 4

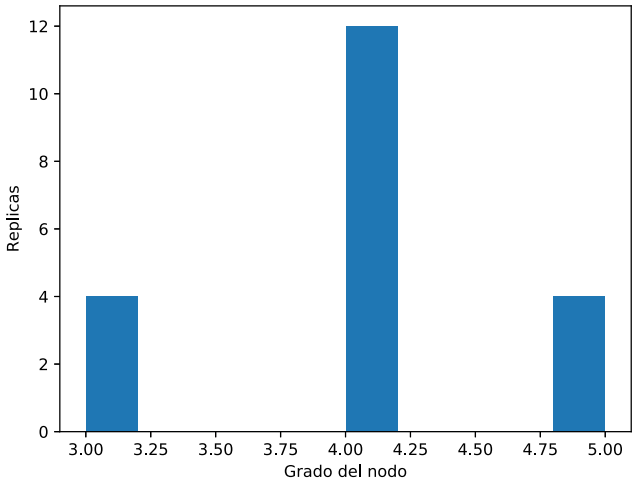

Figura 15: Grafo 5

2.2. Coeficiente de agrupamiento entre nodos

A continuación se muestran histogramas relacionadas a los coeficiente de agrupamiento de los nodos de cada grafo:

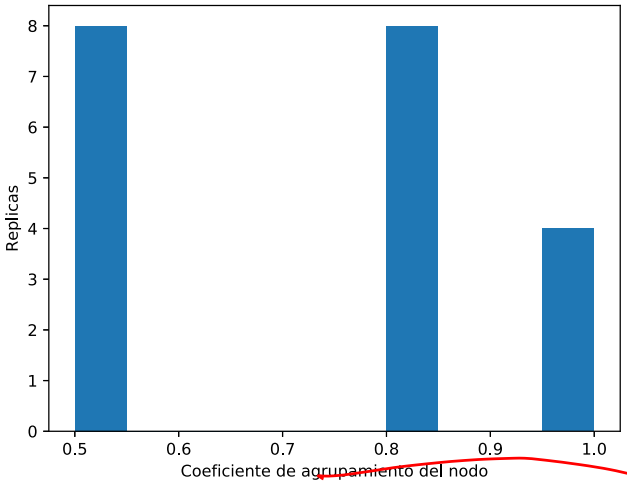

Figura 16: Grafo 1

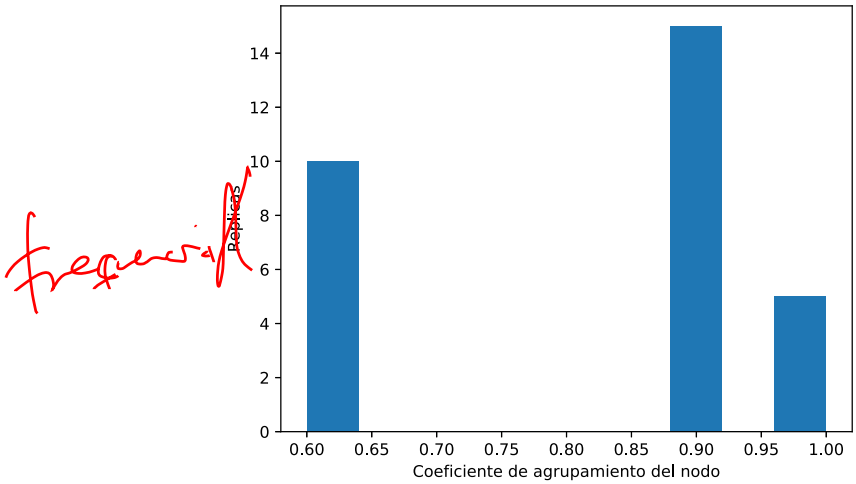

Figura 17: Grafo 2

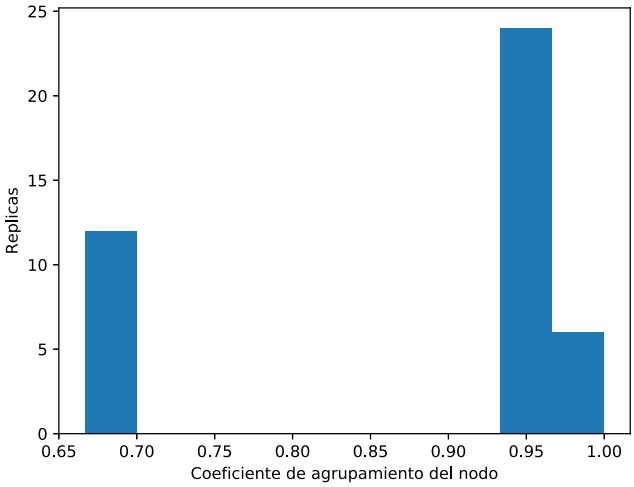

Figura 18: Grafo 3

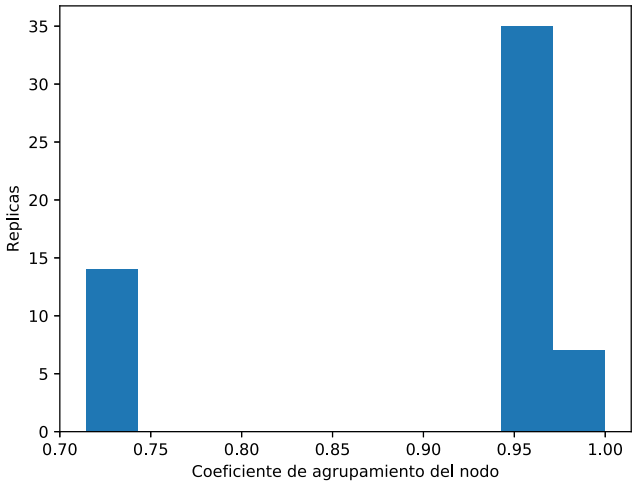

Figura 19: Grafo 4

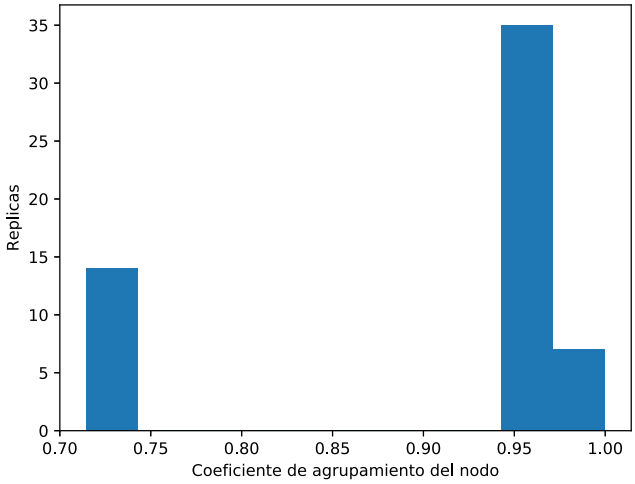

Figura 20: Grafo 5

2.3. Centralidad de cercanía entre nodos

A continuación se muestran histogramas relacionadas a la centralidad de cercanía de los nodos de cada grafo:

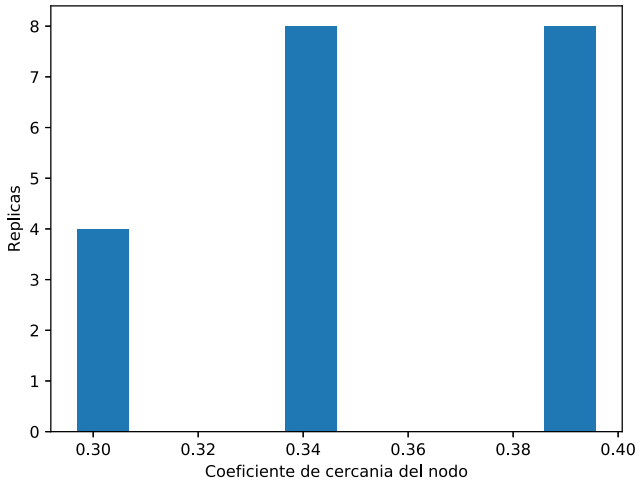

Figura 21: Grafo 1

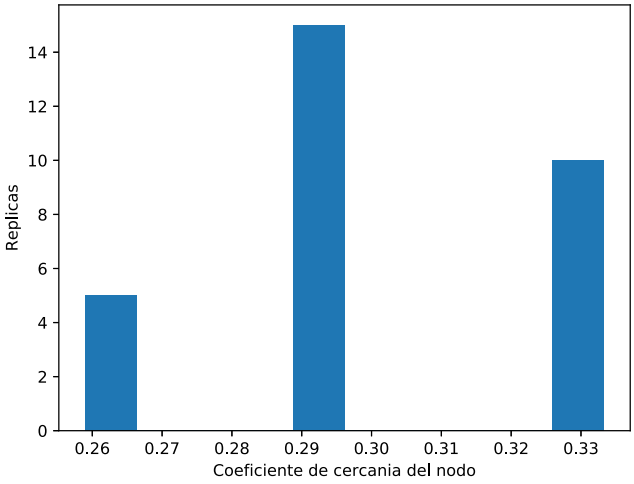

Figura 22: Grafo 2

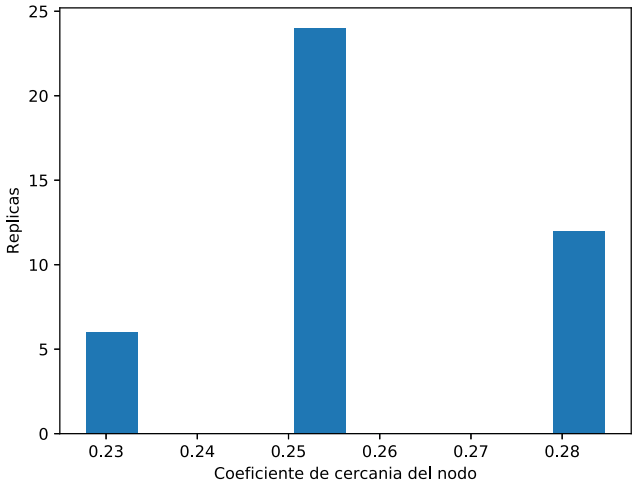

Figura 23: Grafo 3

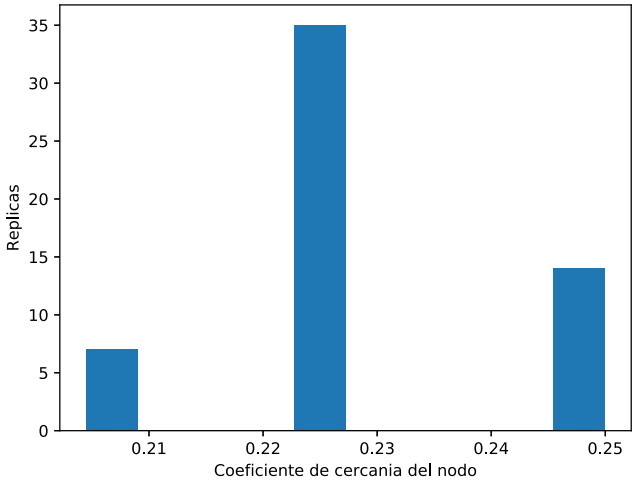

Figura 24: Grafo 4

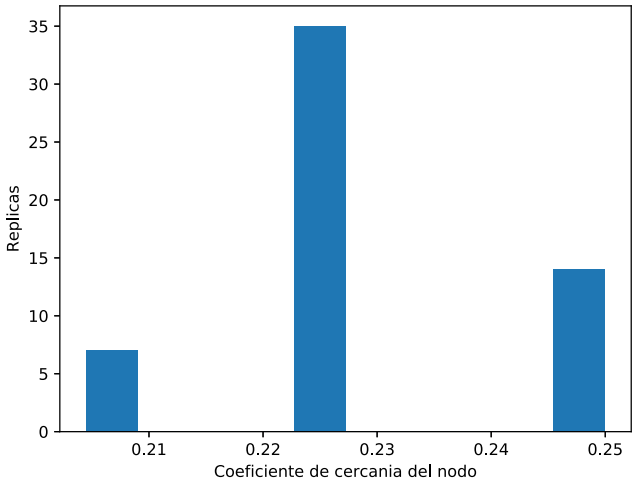

Figura 25: Grafo 5

2.4. Centralidad de carga entre nodos

A continuación se muestran histogramas relacionadas a la centralidad de carga de los nodos de cada grafo:

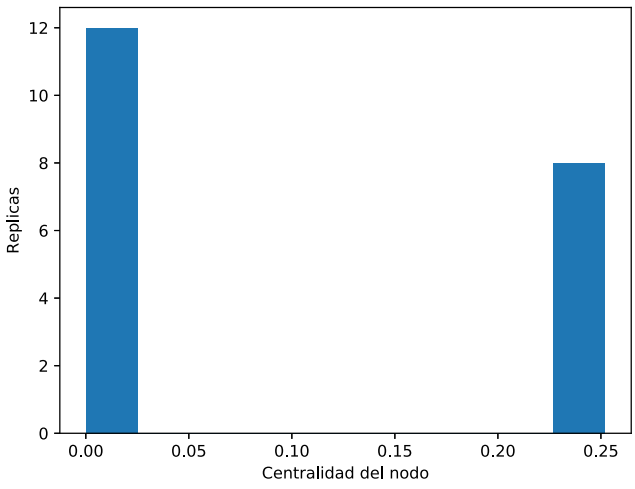

Figura 26: Grafo 1

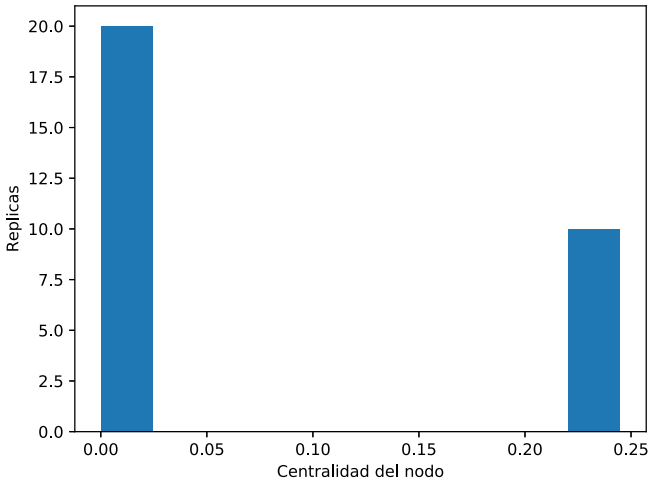

Figura 27: Grafo 2

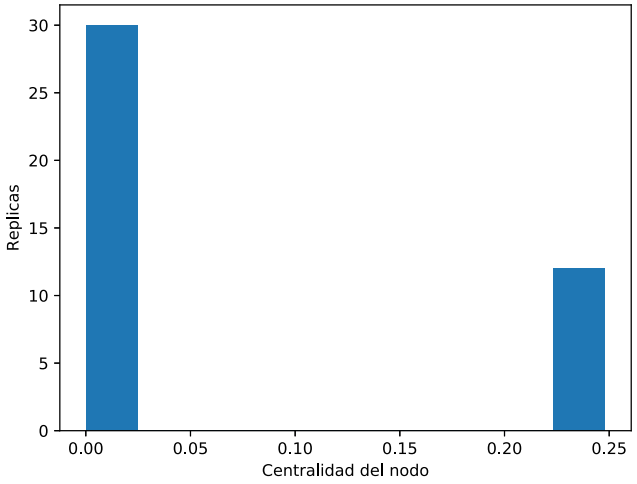

Figura 28: Grafo 3

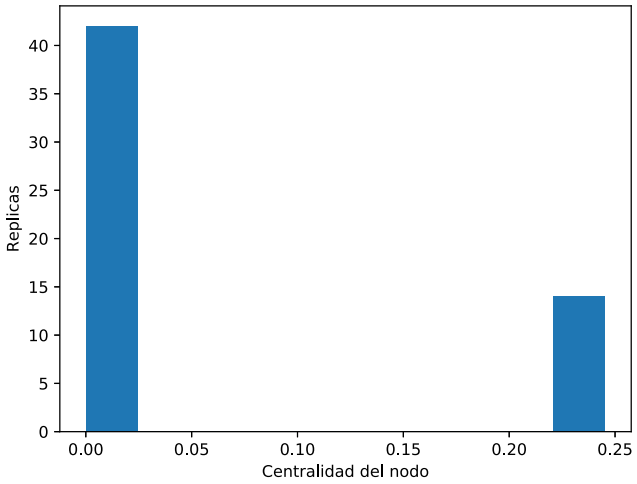

Figura 29: Grafo 4

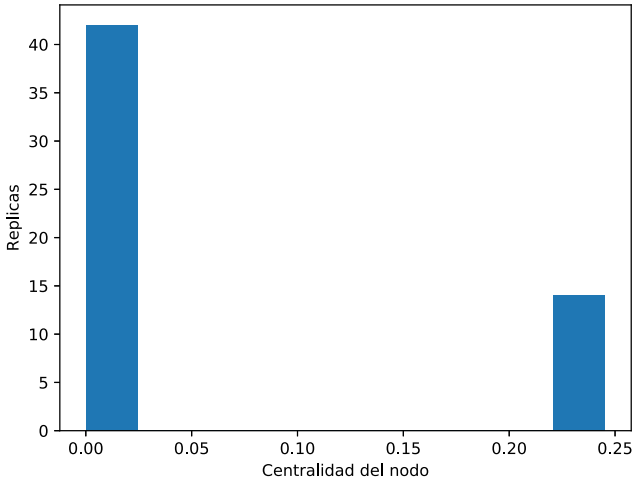

Figura 30: Grafo 5

2.5. Excentricidad entre nodos

A continuación se muestran histogramas relacionadas a la excentricidad de los nodos de cada grafo:

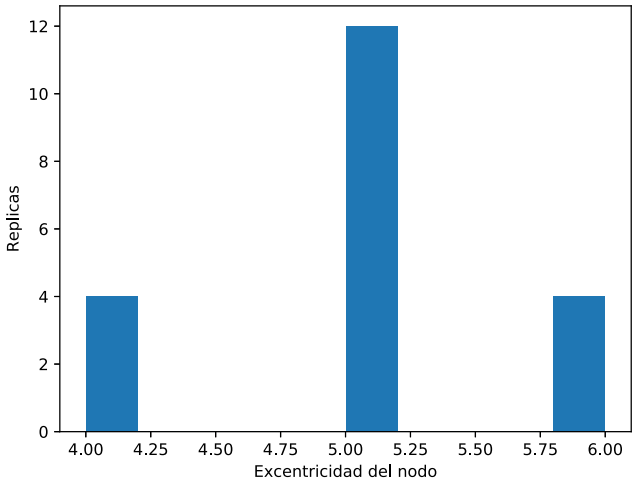

Figura 31: Grafo 1

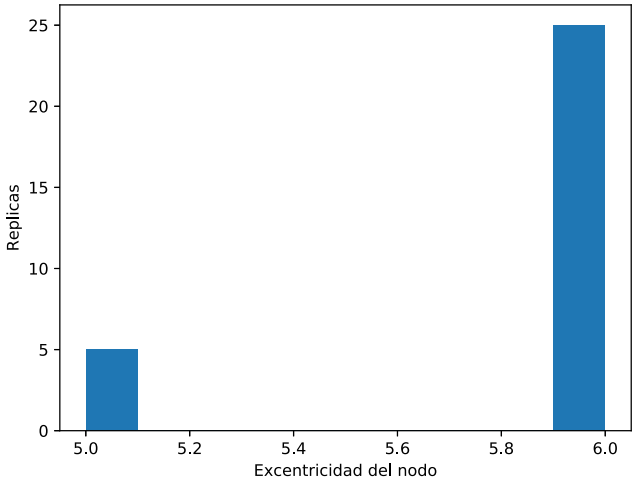

Figura 32: Grafo 2

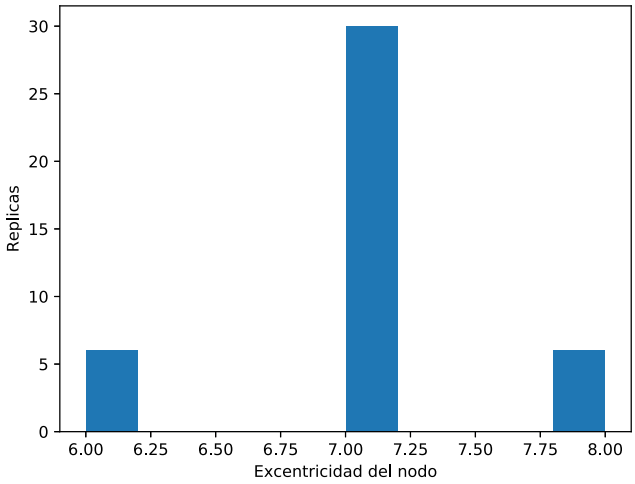

Figura 33: Grafo 3

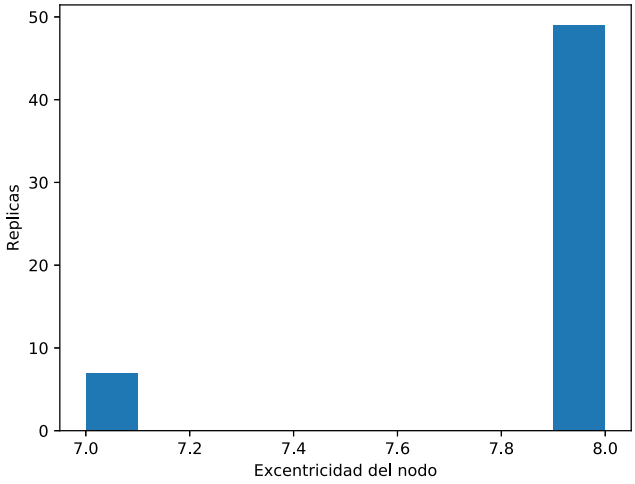

Figura 34: Grafo 4

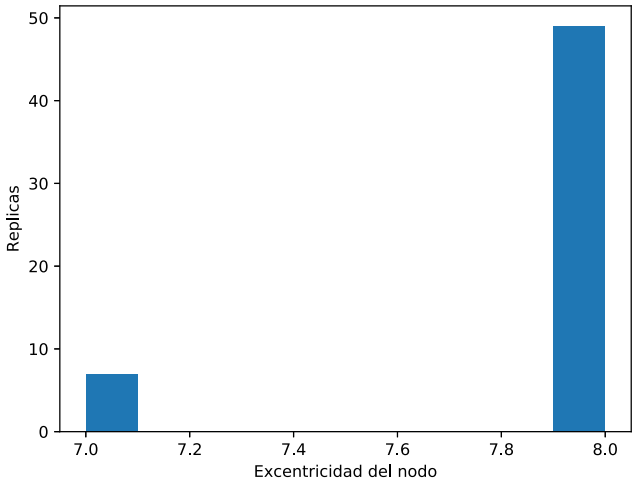

Figura 35: Grafo 5

2.6. PageRank entre nodos

A continuación se muestran histogramas relacionadas al PageRank de los nodos de cada grafo:

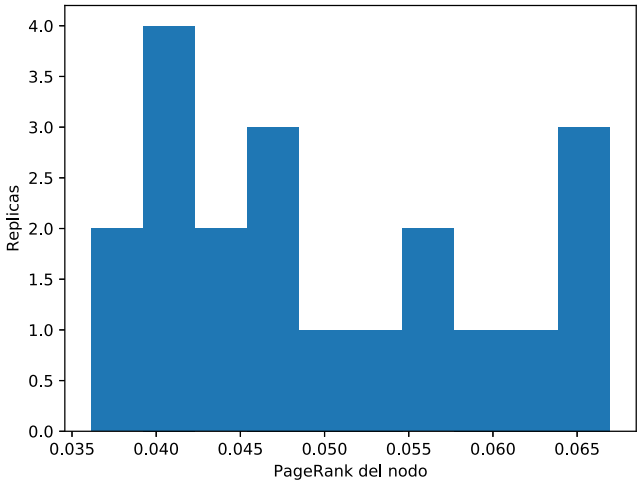

Figura 36: Grafo 1

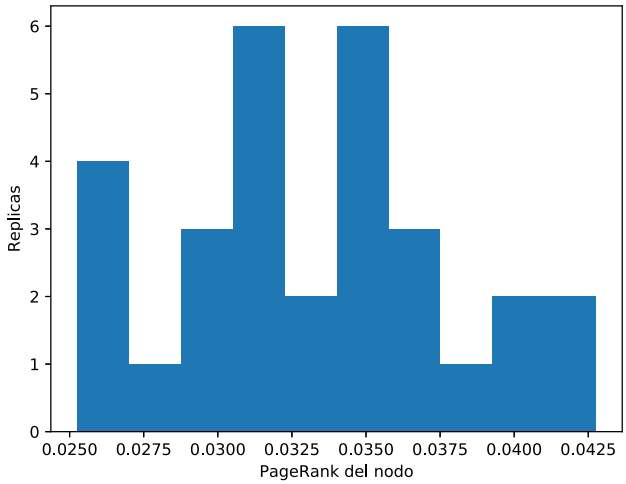

Figura 37: Grafo 2

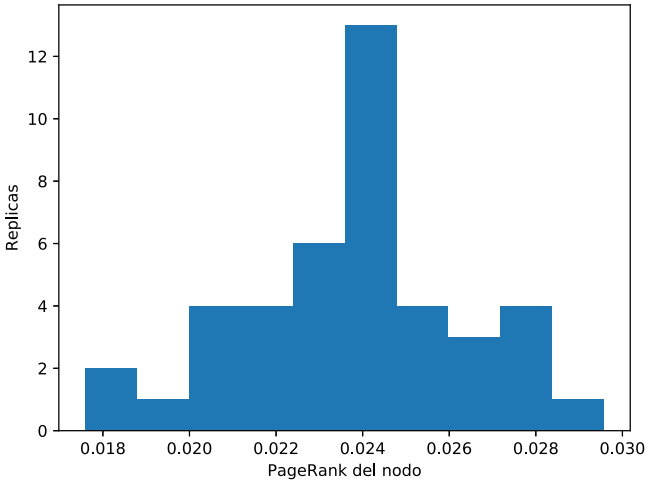

Figura 38: Grafo 3

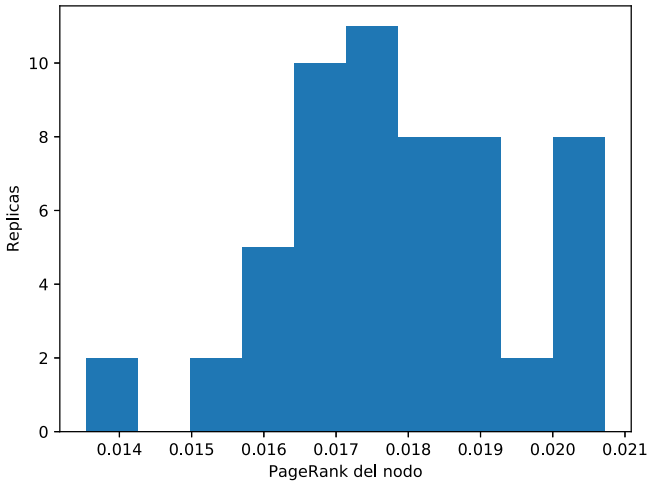

Figura 39: Grafo 4

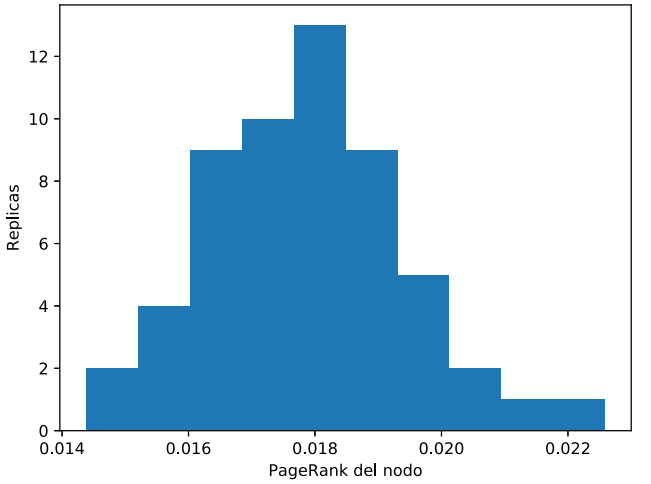

Figura 40: Grafo 5

Se obtuvieron los siguientes resultados de las replicas, las pruebas estadísticas de ANOVA para determinar los efectos de los factores:

	GL	PR(>F)
G	1	1
A	1	1
G:A	1	1
C	1	1
G:C	1	1
A:C	1	0.993
G:A:C	1	1
Ce	1	1
G:Ce	1	1
A:Ce	1	1
G:A:Ce	1	1
C:Ce	1	1
G:C:Ce	1	1
A:C:Ce	1	1
G:A:C:Ce	1	0.093
E	1	1
G:E	1	1
A:E	1	1
G:A:E	1	1
C:E	1	1
G:C:E	1	1
A:C:E	1	1
G:A:C:E	1	0
Ce:E	1	1
G:Ce:E	1	1
A:Ce:E	1	1
G:A:Ce:E	1	0.301
C:Ce:E	1	1
G:C:Ce:E	1	0.66
A:C:Ce:E	1	0
G:A:C:Ce:E	1	0
P	1	0.993
G:P	1	0.97
A:P	1	0.992
G:A:P	1	0.96
C:P	1	0.872
G:C:P	1	1
A:C:P	1	0.992
G:A:C:P	1	0.003
Ce:P	1	1
G:Ce:P	1	1
A:Ce:P	1	1
G:A:Ce:P	1	0.041
C:Ce:P	1	1
G:C:Ce:P	1	0.537
A:C:Ce:P	1	0.676
G:A:C:Ce:P	1	0
E:P	1	0.99
G:E:P	1	1
A:E:P	1	1
G:A:E:P	1	0.003
C:E:P	1	0.958
G:C:E:P	1	0
A:C:E:P	1	0
G:A:C:E:P	1	0
Ce:E:P	1	1
G:Ce:E:P	1	0.635
A:Ce:E:P	1	0.014
G:A:Ce:E:P	1	0
C:Ce:E:P	1	0.519
G:C:Ce:E:P	1	0
A:C:Ce:E:P	1	0
G:A:C:Ce:E:P	1	0.54
Residual	45	0

G	Grado
A	Agrupamiento
C	Centralidad
Ce	Cercanía
E	Excentricidad
P	PageRank

3. Conclusiones

Como conclusión de esta investigación, se tiene que es solamente en combinación de las características estudiadas cuando afecta el rendimiento de tiempo de ejecución del algoritmo de flujo máximo, particularmente el PageRank es la característica que más afecta en combinación con otros. La tabla ANOVA respalda los resultados con los valores de P dados.

Referencias

[1] Python Software Foundation Versión 3.7.2. https://www.python.org/.

[2] NetworkX developers con última actualización el 19 de Septiembre 2018. https://networkx.github.io/documentation/stable/index.html.

[3] NetworkX developers Versión 2.0. https://networkx.github.io/.

[4] The Matplotlib development team Versión 3.0.2. https://matplotlib.org/.

[5] Jesús Angel Patlán Castillo. Repositorio Optimización Flujo en Redes. https://github.com/JAPatlanC/Flujo-Redes.