Tarea 5

5280

30 de abril de 2019

Generador de grafos

En los grafos presentados, los nodos representan enrutadores en redes locales de comunicación. Se asume que la capacidad de los enlaces está determinada por el tipo de enlace físico que une los enrutadores y no por las características de los mismos.

Para la generación se emplea el algoritmo `dense_gnm_random`, que recibe como parámetros directamente el número de nodos y el número de aristas, lo que resulta cómodo para simular este tipo de redes.

Algoritmo de flujo máximo

El algoritmo de flujo máximo elegido fue el Edmond Karp, pues fue el que mejores resultados mostró en la tarea anterior. Este algoritmo recibe como parámetros el grafo, el nodo fuente y el nodo sumidero.

Algoritmo de ordenamiento

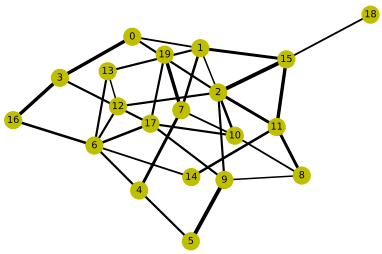
El algoritmo de ordenamiento empleado es el propuesto por Fruchterman y Reingold [1]. Este muestra buenos resultados en la distribución de nodos, tamaño de las aristas uniforme y simetría de manera general, enfocándose en la velocidad y simplicidad.

Generación de grafos

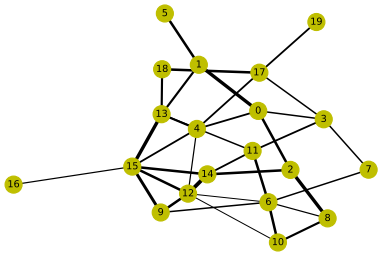
Para realizar el experimento son generados 5 grafos aleatoriamente, con la capacidad de flujo en sus aristas, distribuida normalmente, como puede observarse en la figura 1. El ancho de las aristas es proporcional a su capacidad.

Características estructurales de los nodos

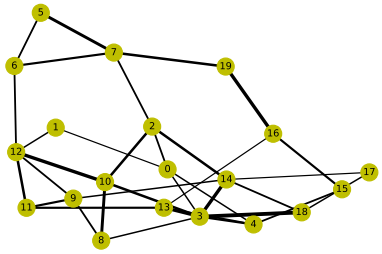
A cada uno de los grafos se les calculan las siguientes características estructurales de sus nodos:



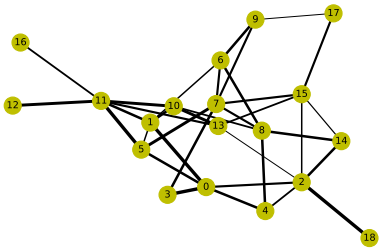
(a) Grafo 1



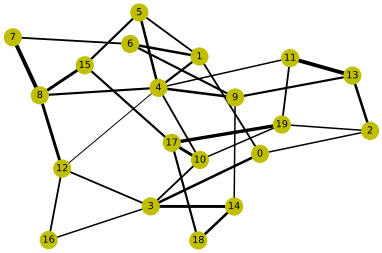
(b) Grafo 2



(c) Grafo 3



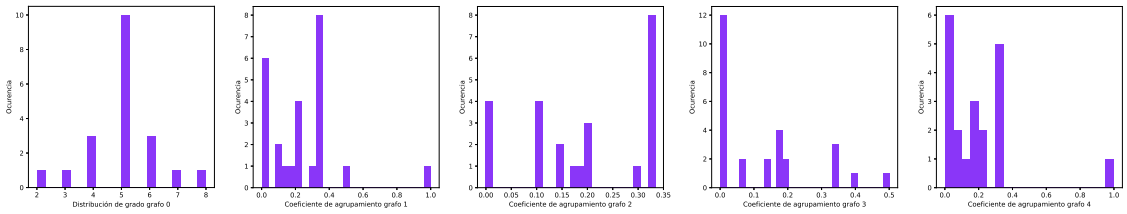
(d) Grafo 4



(e) Grafo 5

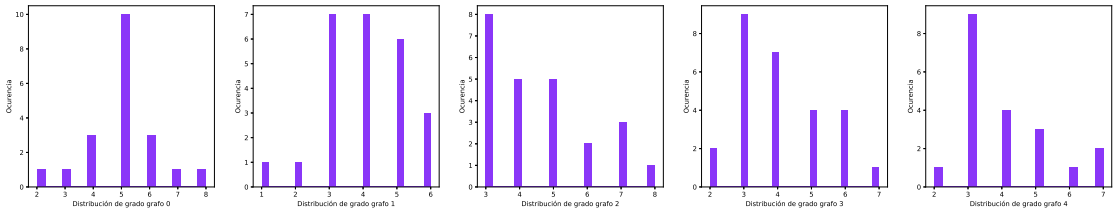
Figura 1: Grafos generados

- Distribución de grado (figura 2)
- Coeficiente de agrupamiento (figura 3)
- Centralidad de cercanía (figura 4)
- Centralidad de carga (figura 5)
- Excentricidad (figura 6)
- Rango de página (figura 7)



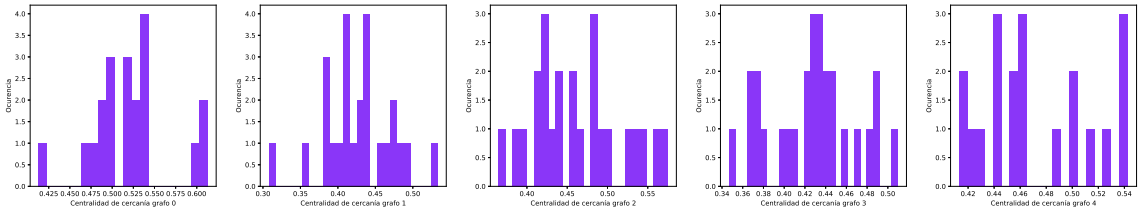
(a) Grafo 1 (b) Grafo 2 (c) Grafo 3 (d) Grafo 4 (e) Grafo 5

Figura 2: Histogramas de la distribución de grado para cada grafo



(a) Grafo 1 (b) Grafo 2 (c) Grafo 3 (d) Grafo 4 (e) Grafo 5

Figura 3: Histogramas del coeficiente de agrupamiento para cada grafo

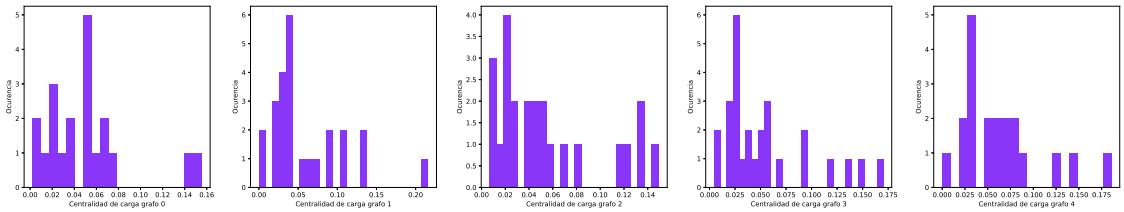


(a) Grafo 1 (b) Grafo 2 (c) Grafo 3 (d) Grafo 4 (e) Grafo 5

Figura 4: Histogramas de la centralidad de cercanía para cada grafo

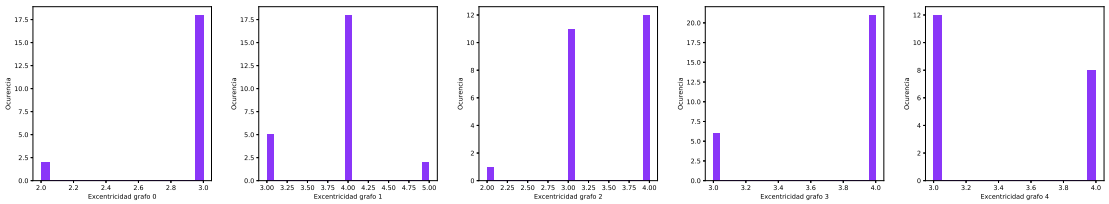
Flujo máximo

Para cada combinación de fuente-sumidero en cada grafo se calcula el flujo máximo. En las figuras 8, 9, 10, 11 y 12, se visualizan los flujos para la mejor y peor pareja de fuente-sumidero para los grafos 1, 2, 3, 4 y 5, respectivamente. Los nodos rojos representan las mejores fuente-sumidero, como los



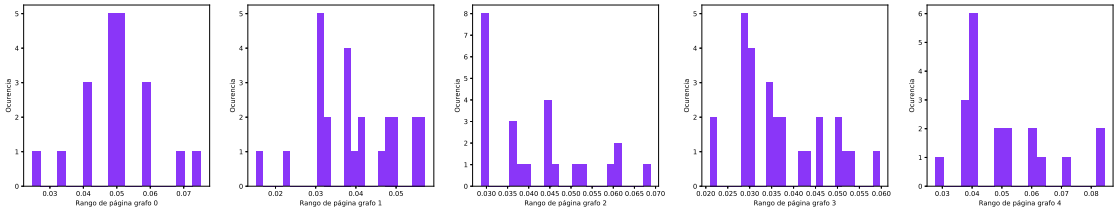
(a) Grafo 1 (b) Grafo 2 (c) Grafo 3 (d) Grafo 4 (e) Grafo 5

Figura 5: Histogramas de la centralidad de carga para cada grafo



(a) Grafo 1 (b) Grafo 2 (c) Grafo 3 (d) Grafo 4 (e) Grafo 5

Figura 6: Histogramas de la excentricidad para cada grafo

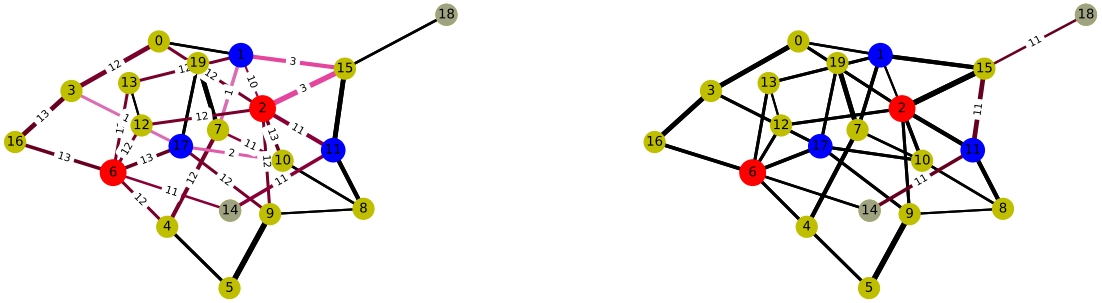


(a) Grafo 1 (b) Grafo 2 (c) Grafo 3 (d) Grafo 4 (e) Grafo 5

Figura 7: Histogramas de la excentricidad para cada grafo

grafos son no dirigidos, estos son intercambiables. Los nodos azules son buenas fuente-sumidero y los nodos grises son las peores fuentes-sumideros, en cada grafo.

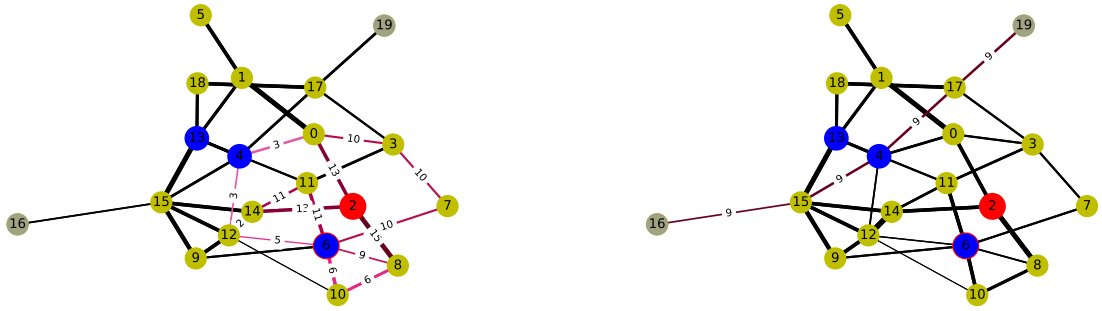
En rosado se muestra cuanto flujo pasa por cada arista una vez aplicado al algoritmo de flujo máximo, aunmentando la intensidad del color proporcionalmente a la cantidad de flujo. En las aristas negras no pasa flujo.



(a) Mejor pareja, flujo: 73

(b) Peor pareja, flujo: 11

Figura 8: Visualización de la mejor y peor pareja de nodos a usar como fuente o sumidero en el grafo 1



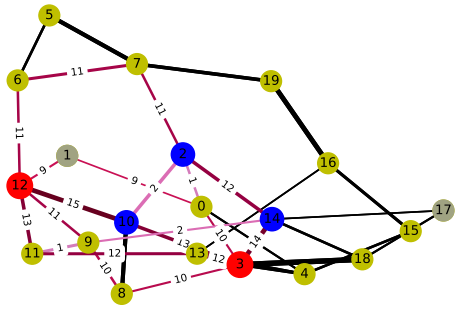
(a) Mejor pareja, flujo: 66

(b) Peor pareja, flujo: 9

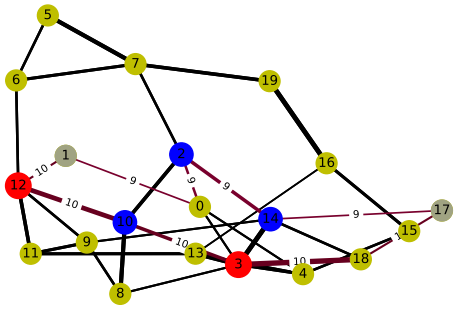
Figura 9: Visualización de la mejor y peor pareja de nodos a usar como fuente o sumidero en el grafo 2

Influencia de las características estructurales del grafo en el tiempo de ejecución

En el Cuadro 1 se muestran los resultados del análisis de varianza (ANOVA en lo adelante, por sus siglas en inglés) de las carácterísticas estructurales del grafo para ver su influencia en el tiempo de ejecución del algoritmo de flujo máximo.

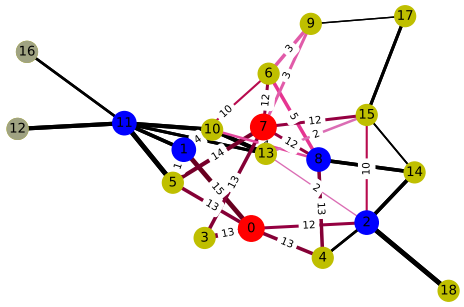


(a) Mejor pareja, flujo: 59

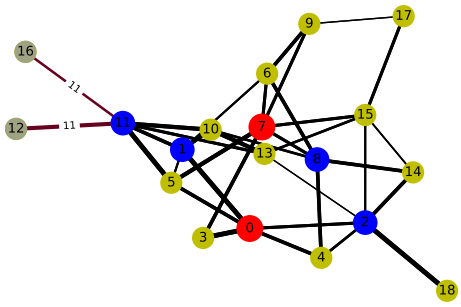


(b) Peor pareja, flujo: 19

Figura 10: Visualización de la mejor y peor pareja de nodos a usar como fuente o sumidero en el grafo 3

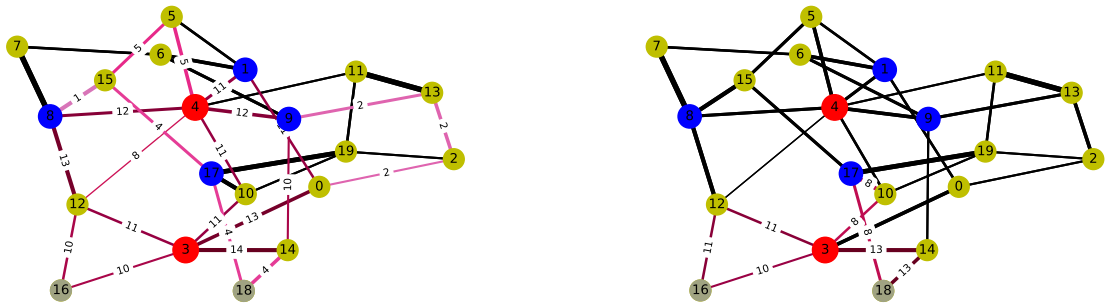


(a) Mejor pareja, flujo: 66



(b) Peor pareja, flujo: 11

Figura 11: Visualización de la mejor y peor pareja de nodos a usar como fuente o sumidero en el grafo 4



(a) Mejor pareja, flujo: 59

(b) Peor pareja, flujo: 21

Figura 12: Visualización de la mejor y peor pareja de nodos a usar como fuente o sumidero en el grafo 5.

Se aprecia que las carácterísticas estructurales de los grafos no influyen en el tiempo de ejecución del alogoritmo de flujo máximo para los grafos generados. Se observa, además, que existe una relación entre las siguientes características:

- Distribución de grado y coeficiente de agrupamiento
- Distribución de grado y centralidad de cercanía
- Distribución de grado y centralidad de carga
- Rango de página y centralidad de cercanía
- Rango de página y centralidad de carga

Cuadro 1: Resultado del ANOVA para los valores de tiempo

Factor	suma_cuad	Grados de libertad	F	Prueba
Grado	8.12E-08	1	0.96	0.33
CoefA	6.78E-08	1	0.8	0.37
CentCe	5.12E-09	1	0.06	0.8
CentCa	2.16E-07	1	2.5	0.11
Excent	2.32E-08	1	0.27	0.6
RangoP	2.11E-07	1	2.48	0.11
Grado:CoefA	4.45E-07	1	5.23	0.02
Grado:CentCe	1.00E-06	1	11.8	0.0006
Grado:CentCa	1.55E-06	1	18.28	2.00E-05
Grado:Excent	1.84E-07	1	2.16	0.14
Grado:RangoP	2.30E-07	1	2.7	0.1
CentCe:CentCa	2.11E-07	1	2.48	0.12
CentCe:Excent	1.53E-07	1	1.8	0.18
CentCe:RangoP	7.70E-07	1	9.06	0.002
CentCa:Excent	1.30E-07	1	1.53	0.22
CentCa:RangoP	1.89E-06	1	22.25	2.50E-06
Excent:RangoP	1.52E-07	1	1.79	0.18
Residual	0.00015678	1844		

Influecia de las características estructurales del grafo en el flujo máximo

En el Cuadro 1 se muestra el ANOVA de las carácterísticas estructurales del grafo para analizar su influencia en el flujo máximo.

Se aprecia inciden en el flujo máximo las siguientes características estructurales:

- Coeficiente de agrupamiento
- Centralidad de carga
- Rango de página

Se observa, además, que existe una relación entre las siguientes características:

- Distribución de grado y coeficiente de agrupamiento
- Distribución de grado y rango de página
- Coeficiente de agrupamiento y centralidad de cercanía

- Coeficiente de agrupamiento y rango de página

Cuadro 2: Resultado del ANOVA para los valores de flujo máximo

Factor	sum_cuad	Grados de libertad	F	Prueba
Grado	38.7288463	1	0.459047	0.498155
CoefA	1363.23246	1	16.15817	6.06E-05
CentCe	143.623743	1	1.702349	0.192144
CentCa	976.245999	1	11.57129	0.000684
Excent	155.426353	1	1.842243	0.174855
PageR	508.236449	1	6.024044	0.014204
Grado:CoefA	650.232041	1	7.707095	0.005556
Grado:CentCe	93.006395	1	1.10239	0.29388
Grado:CentCa	112.234812	1	1.330301	0.248901
Grado:Excent	162.655867	1	1.927934	0.165153
Grado:PageR	660.247786	1	7.82581	0.005204
CoefA:CentCe	335.675561	1	3.978708	0.046226
CoefA:CentCa	49.9598544	1	0.592166	0.441682
CoefA:Excent	16.9158218	1	0.2005	0.65437
CoefA:PageR	553.464369	1	6.560123	0.010508
CentCe:CentCa	7.00601142	1	0.083041	0.77325
CentCe:Excent	23.2094891	1	0.275098	0.599995
CentCe:PageR	85.7199062	1	1.016024	0.313596
CentCa:Excent	66.437239	1	0.78747	0.374982
CentCa:PageR	106.077006	1	1.257314	0.262307
Excent:PageR	104.744206	1	1.241516	0.265325
Residual	155237.086	1840		

Código fuente utilizado para realizar el experimento

Generación de grafos

```
def GenerateGraph(nodes,edges,address):
    S=nx.dense_gnm_random_graph(nodes,edges)
    scale = 2
    rang = 10
    e=S.edges(nbunch=None, data=True, default=None)
    X = truncnorm(a=-rang/scale, b=+rang/scale, scale=scale).rvs(size=edges)
    X = X.round().astype(int)+rang+2
    G=nx.Graph()
    count=0;
    for i in e:
```

```
            G.add_edge(i[0],i[1],capacity=X[count])
            count+=1
    df = pd.DataFrame()
    df = nx.to_pandas_adjacency(G, dtype=int, weight='capacity')
    df.to_csv(address, index=None, header=None)

def Print(graph,address, pos_address,fig):
    ds = pd.read_csv(graph, header=None)
    G = nx.from_pandas_adjacency(ds)
    pos=nx.fruchterman_reingold_layout(G)
    X=[]
    for edge in G.edges():
        X.append( G.edges[edge]['weight'] )
    X[:] = [x/10*x/8 for x in X]
    labels = {}
    for i in G.nodes:
        labels[i]=str(i)
    nx.draw_networkx_nodes(G, pos, node_size=200, node_color='y', node_shape='o')
    nx.draw_networkx_edges(G, pos, edge_color='black', width=X)
    nx.draw_networkx_labels(G, pos, labels, font_size=8 )
    plt.axis('off')
    plt.savefig(fig,dpi=500)
    df = pd.DataFrame(pos)
    df.to_csv(address, index=None, header=None)
    return(G)
```

Grafos.py

Cálculo de atributos

```
def Atributes(G):
    dic={}
    Nodes=G.nodes;
    dic["Nodo"]=Nodes
    dic["Grado"]=[G.degree(i) for i in Nodes]
    dic["CoefA"]=[nx.clustering(G,i) for i in Nodes]
    dic["CentCe"]=[nx.closeness_centrality(G,i) for i in Nodes]
    dic["CentCa"]=[nx.load_centrality(G,i) for i in Nodes]
    dic["Excent"]=[nx.eccentricity(G,i) for i in Nodes]
    PageR=nx.pagerank(G,weight="capacity")
    dic["PageR"]=[PageR[i] for i in Nodes]
    df=pd.DataFrame(dic)
    df.to_csv("matrix5.csv", index=None)
```

Programa.py

Cálculo de tiempo de ejecución y flujo máximo

```
def Time(G):
    dic={"Fuente":[], "Sumidero":[] , "Media":[], "Mediana":[], "Std":[], "MaxFlow"
    :[]}
    Nodes=G.nodes;
    for i in Nodes:
        for j in Nodes:
            if i!=j:
                t=[]
                for k in range(10):
                    t.append(Edmond(G,i,j))
                dic["Fuente"].append(i)
                dic["Sumidero"] .append(j)
```

```
                    dic["Media"].append(np.mean(t))
                    dic["Mediana"].append(np.median(t))
                    dic["Std"].append(np.std(t))
                    dic["MaxFlow"].append(nx.maximum_flow_value(G,i,j,capacity="weight")
    )
    df=pd.DataFrame(dic)
    df.to_csv("times5.csv", index=None)
```

Programa.py

ANOVA

```
modelo = ols('Flujomaximo ~ Grado+CoefA+CentCe+CentCa+Excent+PageR+Grado*CoefA+Grado
    *CentCe+Grado*CentCa+Grado*Excent+Grado*PageR+CentCe*CentCa+CentCe*Excent+CentCe
    *PageR+CentCa*Excent+CentCa*PageR+Excent*PageR',data=df).fit()
print(modelo.summary())
modelo_csv = open("Anova_Mult.csv", 'w')
aov_table = sm.stats.anova_lm(modelo, typ=2)
df1=pd.DataFrame(aov_table)
```

procesamiento.py

Referencias

[1] Thomas MJ Fruchterman and Edward M Reingold. Graph drawing by force-directed placement. *Software: Practice and experience*, 21(11):1129–1164, 1991.