Representación de redes a través de la teoría de grafos

Brenda Yaneth Sotelo Benítez

12 de febrero de 2019

En esta práctica se realizó una investigación de tipos de grafos simples y multígrafos, donde para cada uno de ellos se presenta una definición,representación de un ejemplo y aplicaciones prácticas.

El lenguaje de programación que se ha utilizado para la creación y representación de los ejemplos es `Python`[1] con el uso del paquete NetwokX [2] y Matplotlib [3].

Se comenzará definiendo algunos conceptos importantes basados en [4, 5] con el objetivo de tener una estructura organizada para una clara comprensión del tema.

Conceptos básicos

Un ***grafo*** G es un par G=(V,A) donde V es el conjunto de vértices o nodos y A un conjunto de aristas o arcos que se representan como pares de vértices (v_i,v_j) v_i,$v_j \in V$.

Si las aristas tienen dirección, es un grafo ***dirigido u orientado*** también llamado *digrafo*. En cada arista dirigida (v_i,v_j) el primer elemento representa el origen(o fuente) de la arista y el segundo es el destino. Si las aristas no tienen dirección se trata de un grafo ***no dirigido*** donde (v_i, v_j)=(v_j, v_i) $\forall v_i, v_j \in V$. A los nodos y aristas se les puede asignar nombres o etiquetas, teniendo así un *grafo etiquetado*, mientras que asignar pesos a las aristas se denomina *grafo ponderado*.

Un ***ciclo*** es una secuencia de aristas consecutivas que empieza y termina en el mismo vértice. Un grafo que no contiene ciclos es ***acíclico***. Los ciclos de longitud 1 se llaman *bucles o lazos*, ya que estos inician y terminan en el mismo vértice sin pasar por ningun otro.

Si un grafo cuenta con al menos un bucle , el grafo es ***reflexivo***.

En un grafo ***simple*** no existe más de una arista para cada par de vértices, por otro lado, si el grafo no es simple se le llama grafo múltiple o ***mutigrafo***.

1. Grafo simple no dirigido acíclico

Son usados para representar árboles genealógicos, organigramas, redes de distribución, carreteras(aristas) y ciudades(vértices) o bien en la química orgánica para representar compuestos. En la figura 1 se muestra un tipo de hidrocarburo saturado donde los vértices representan los atómos de carbono e hidrógeno y las aristas los enlaces entre ellos.

```
G=nx.Graph()
V={0:(0,0),1:(0.4,0),2:(0,1),3:(0.4,1),4:(0,2),5:(0.4,2)
      ,6:(0.2,5.0),7:(0.2,-3.0),8:(0.2,0),9:(0.2,1),
    10:(0.2,2)}
A=[(0,1),(2,3),(4,5),(6,7),(8,9)]

nx.draw_networkx_nodes(G,V,nodelist=[0,1,2,3,4,5,6,7,8,9,10],
      node_color='black')
nx.draw_networkx_edges(G,V,width=1,edgelist=A)
```

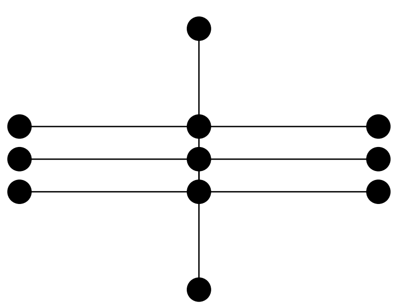

Figura 1: *Propano (C_3H_8)*

2. Grafo simple no dirigido cíclico

Entre sus aplicaciones está la representación de polígonos, redes eléctricas, en el problema del agente viajero, esquemas, representar calles en doble sentido, etc.

Un ejemplo red de ocho ordenadores se observa en la figura 2 que pueden conectarse de múltiples maneras y dan lugara un grafo no dirigido cíclico.

```
G=nx.Graph()
V={0:(0,0),1:(0,1),2:(1,0),3:(1,1),4:(0.5,0), 5:(0.5,1),6:(-0.5,1)
    ,7:(-0.5,2)}
A=[(0,1),(2,3),(4,5),(0,2),(1,5),(5,3),(1,6),(6,7)]
labels={2:'1',3:'2',4:'3',5:'4',0:'5',1:'6',6:'7',7:'8'}

nx.draw_networkx_nodes(G,V,nodelist=[0,1,2,3,4,5,6,7],node_color='b
    ')
nx.draw_networkx_edges(G,V,width=1,edgelist=A,alpha=1)
nx.draw_networkx_labels(G,V,labels, front_size=12)
```

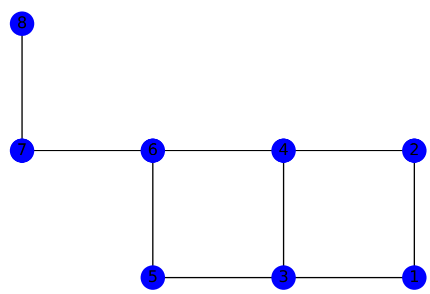

Figura 2: *Red de ordenadores*

3. Grafo simple no dirigido reflexivo

Durante la conducción es típico presentarse con situaciones en las que es necesario pasar de una avenida a otra o cambiar la dirección de la ruta. Para ello, nos vemos en la necesidad de buscar retornos ya establecidos o en su caso, buscar alguna calle para retornar y continuar con el viaje. Esta situación puede ser representada como un grafo tal y como se muestra en la figura 3 donde las aristas representan las calles y el vértice rojo el lazo o retorno.

```
G=nx.Graph()
V={0:(0,0),1:(1,0),2:(0.5,1),3:(0,1),4:(0,2)}
A=[(0,1),(1,2),(2,3),(3,4)]
labels={1:'L'}

nx.draw_networkx_nodes(G,V, nodelist=[1])
nx.draw_networkx_nodes(G,V, nodelist=[0,2,3,4], node_color='b')
nx.draw_networkx_edges(G,V, width=1, edgelist=A, alpha=1)
nx.draw_networkx_labels(G,V, labels,  front_size=12)
```

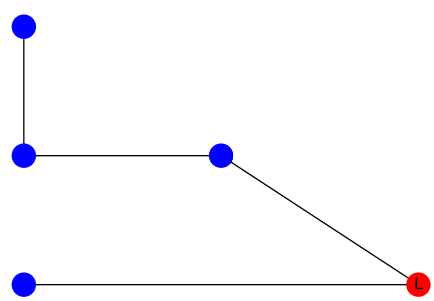

Figura 3: *Representación de calles con un retorno*

Otro ejemplo en la vida real se encuentra cuando brindamos ayuda para realizar alguna tarea o trabajo y al mismo tiempo se obtiene el beneficio de reforzar conocimiento(lazo). Pueden planterse muchos otros problemas más de esta manera pues hasta el organizar nuestras actividades del día puede ser representada por un grafo.

4. Grafo simple dirigido acíclico

En la figura 4 se plantea un modelo de conjuntos de tareas que necesitan una secuencia particular para realizarse y donde es importante que no existan ciclos para que las tareas no se repitan. Los problemas de transporte y asignación, asi como las redes neuronales pueden ser planteados de forma muy sencilla mediante este tipo de grafo.

```
G=nx.DiGraph()
V={0:(-2,0.2),1:(-1,0.2),2:(0,0.2),3:(0,0),4:(1,0.2),5:(1,0),6:(1.5,-0.2),7:(2,0.2),8:(2.5,0),9:(4,0),10:(5,0)}
A=[(0,1),(1,2),(2,3),(3,4),(3,5),(3,6),(4,7),(5,8),(8,6),(7,9),(9,10),(6,9),(1,3)]
labels={0:'1',1:'2',2:'3',3:'4',4:'5',5:'6',6:'9',7:'7',8:'8',9:'10',10:'11'}

nx.draw_networkx_nodes(G,V,nodelist=[0,1,2,3,4,5,6,7,8,9,10],node_color='b')
nx.draw_networkx_edges(G,V,width=1,edgelist=A,alpha=1)
nx.draw_networkx_labels(G,V,labels, front_size=12)
```

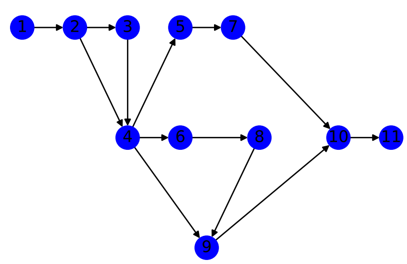

Figura 4: *Secuencia de tareas*

5. Grafo simple dirigido cíclico

El problema de VRP(ruteo de vehículos) es un problema clásico en la optimización de operaciones en el cual se tiene un depósito, un conjunto de vehículos que siguen una ruta y clientes que deben ser atendidos, donde cada uno representan las aristas y vértices respectivamente. Cada una de las rutas debe regresar al depósito. Ver figura 5.

```
G=nx.DiGraph()

V={0:(0,0),1:(1,1),2:(3,2),3:(4,1),4:(2,-1),5:(-1,2),6:(-3,1),7:(-4,-1),8:(-2,-2)}
A=[(0,1),(1,2),(2,3),(3,4),(4,0),(0,5),(5,6),(6,7),(7,8),(8,0)]
labels={0:'Centro'}

nx.draw_networkx_nodes(G,V,nodelist=[0,1,2,3,4],node_color='c')
nx.draw_networkx_nodes(G,V,nodelist=[5,6,7,8],node_color='y')
nx.draw_networkx_edges(G,V,width=1,edgelist=A,alpha=1)
nx.draw_networkx_labels(G,V,labels, front_size=12)
```

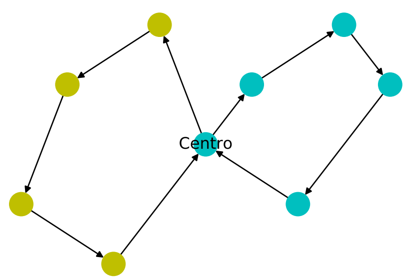

Figura 5: *VRP*

Los grafos simples dirigidos cíclicos también suelen presentarse en los diagramas de flujo.

6. Grafo simple dirigido reflexivo

Un ejemplo está dado por la representación de una tienda proveedora que vende productos a varias sucursales pequeñas y que a su vez vende para si misma. Ver figura 6. El vértice P es la tienda proveedora que tiene un lazo, mientras que los vétices etiquetados representan las tiendas.

```
G=nx.DiGraph()

V={0:(0,0),1:(1,5),2:(5,3),3:(7,0),4:(7,-3)}
A=[(0,1),(0,2),(0,3),(0,4),(3,4)]
labels={0:'P',1:'T1',2:'T2',3:'T3',4:'T4'}

nx.draw_networkx_nodes(G,V,nodelist=[1,2,3,4],node_color='c')
nx.draw_networkx_nodes(G,V,nodelist=[0],node_color='g')
nx.draw_networkx_edges(G,V,width=1,edgelist=A,alpha=1)
nx.draw_networkx_labels(G,V,labels, front_size=12)
```

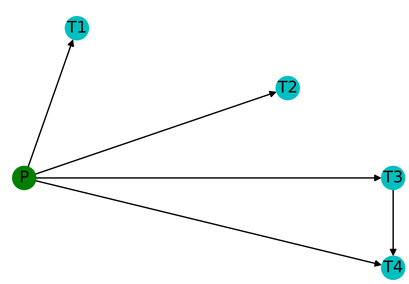

Figura 6: *Red de distribución*

7. Multigrafo no dirigido acíclico

El camino que sigue un río cuando pasa por ciertas localidades se puede representar mediante un multigrafo tal como se observa en figura 7. La arista roja representa en realidad dos aristas, dando lugar aun multigrafo. Esto debido a que un río suele dividirse cuando tiene de por medio obstáculos como tierra, piedras y se disparse en diferentes caminos beneficiando a varias comunidades.

```
G=nx.MultiGraph()

V={0:(0,0),1:(1,1.5),2:(2,1.5),3:(0,-3),4:(3,4), 5:(1,-3.5),6:(3,0)
    }
A=[(0,1),(1,2),(0,3),(2,4),(3,5),(2,6)]

nx.draw_networkx_nodes(G,V, nodelist=[0,1,2,3,4,5,6],node_color='b')
nx.draw_networkx_edges(G,V,width=1,edgelist=A,alpha=1,edge_color='c
    ')
nx.draw_networkx_edges(G,V,width=3,edgelist=[(0,3)],alpha=1,
    edge_color='r')
nx.draw_networkx_labels(G,V,front_size=12)
```

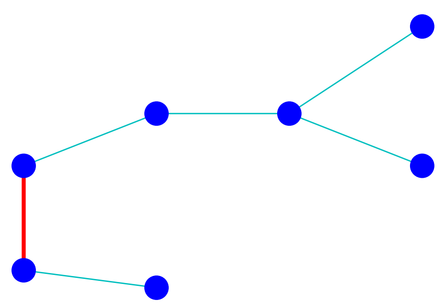

Figura 7: *Representación de un río*

8. Multigrafo no dirigido cíclico

En los aeropuertos existen diversas aerolíneas y cada una de ellas cuenta con cierta cantidad de aviones que parten a su destino y regresan al aeropuerto. Se puede plantear esta situación como un multigrafo dado que puede haber más de un avión que vaya al mismo destino, dirigido y cíclico por el hecho de regresar al origen después de hacer todas sus paradas. Ver figura 8.

```
G=nx.MultiGraph()

V={0:(-3,0),1:(1,0),2:(1,3),3:(-1,3)}
A=[(0,1),(0,2),(2,1),(3,1),(3,2)]

nx.draw_networkx_nodes(G,V, nodelist=[0,1,2,3],node_color='b')
nx.draw_networkx_edges(G,V,width=1,edgelist=A,alpha=1)
nx.draw_networkx_edges(G,V,width=3,edgelist=[(2,1)],alpha=1,
    edge_color='r')
nx.draw_networkx_labels(G,V,front_size=12)
```

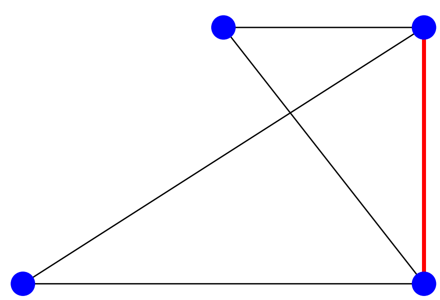

Figura 8: *Vuelos de aerolíneas*

9. Multigrafo no dirigido reflexivo

En el ámbito económico, la figura 9 presenta un multigrafo donde el vértice rojo representa a un inversionista que tiene a varios socios. Él puede decidir si invertir, que alguien invierta en él o que el mismo invierta en si mismo, lo cual esto último crearía el lazo en el grafo.

```
G=nx.MultiDiGraph()

V={0:(0,3),1:(0,0),2:(1,0.5),3:(2,2),4:(-2,2)}
A=[(0,1),(0,2),(0,3),(0,4)]
labels={0:'I', 1:'S1',2:'S2',3:'S3',4:'S4'}

nx.draw_networkx_nodes(G,V,nodelist=[1,2,3,4],node_color='b')
nx.draw_networkx_nodes(G,V,nodelist=[0],node_color='r')
nx.draw_networkx_edges(G,V,width=1,edgelist=A,alpha=1)
nx.draw_networkx_edges(G,V,width=3,edgelist=[(0,3)],alpha=1)
nx.draw_networkx_labels(G,V,labels, front_size=12)
```

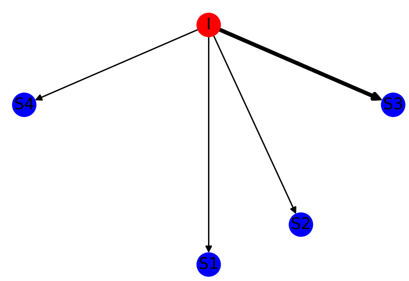

Figura 9: *Proceso de inversión*

10. Multigrafo dirigido acíclico

Un ejemplo de multigrafo dirigido cíclico se puede observar en los ductos de agua, que su representación se muestra en la figura 10 que estan conectados de forma consecutiva sin formar ciclos y donde las aristas celestes son dos tuberías conectadas que llegan al mismo destino.

```
G=nx.MultiDiGraph()

V={0:(0,0),1:(1,0),2:(2,0.2),3:(3,0),4:(4,0)}
A=[(0,1),(1,2),(2,3),(3,4)]

nx.draw_networkx_nodes(G,V, nodelist=[0,1,2,3,4],node_color='b')
nx.draw_networkx_edges(G,V,width=1,edgelist=A,alpha=1)
nx.draw_networkx_edges(G,V,width=3,edgelist=[(0,1),(2,3)],alpha=1,
    edge_color='c')
```

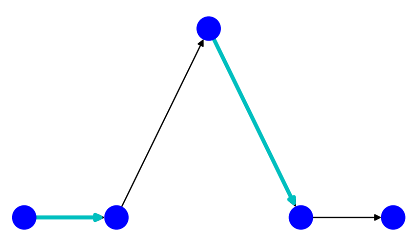

Figura 10: *Ductos de agua*

11. Multigrafo dirigido cíclico

Un ejemplo representativo está en las calles o avenidas, donde para llegar aun lugar existen varias alternativas y en su caso suele presentarse que se forme un ciclo para poder salir de una de ellas. O en el peor de los casos, toparse con calles sin salida o no saber a donde dirigirse. Las flechas resaltadas indican que entre ese par de vértices existen dos aristas o calles. Ver figura 11.

```
G=nx.MultiDiGraph()

V={0:(0,0),1:(1,0),2:(1.5,-1),3:(2,1),4:(-1,0),5:(-2,-1),6:(0,-1)}
A=[(0,6),(6,5),(5,4),(4,0),(1,0),(3,1),(2,1)]

nx.draw_networkx_nodes(G,V, nodelist=[0,1,2,3,4,5,6],node_color='b')
nx.draw_networkx_edges(G,V, width=1,edgelist=A, alpha=1)
nx.draw_networkx_edges(G,V, width=3,edgelist=[(1,0)],alpha=1)
```

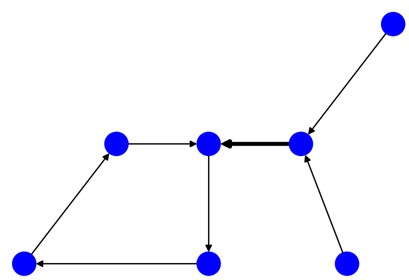

Figura 11: *Representación de avenidas*

Otra manera en la que se hace uso de este multigrafo es en la elaboración de esquemas donde se representen las tareas que cada trabajador debe realizar y considerando que una misma persona puede repetir tareas o más de una persona puede realizar la mismaz tarea.

12. Multigrafo dirigido reflexivo

En la figura 12 se plantea un modelo donde un hospital suminista medicinas a varias estaciones moviles, pero también se suministra el mismo.

```
G=nx.MultiDiGraph()

V={0:(0,0),1:(1,1),2:(1,-1),3:(-1.5,-1.5),4:(0,-3)}
A=[(0,1),(0,2),(0,3),(0,4)]
labels={0:'Hospital'}

nx.draw_networkx_nodes(G,V, nodelist=[1,2,3,4],node_color='pink')
nx.draw_networkx_nodes(G,V, nodelist=[0],node_color='r')
nx.draw_networkx_edges(G,V,width=1,edgelist=A,alpha=1)
nx.draw_networkx_edges(G,V,width=3,edgelist=[(0,3)],alpha=1,
    edge_color='c')
nx.draw_networkx_labels(G,V,labels,  front_size=12)
```

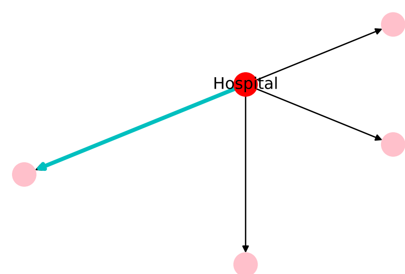

Figura 12: *Distibución de ayuda*

Referencias

[1] https://www.python.org/.

[2] https://networkx.github.io/documentation/networkx 1.10/reference/classes.html.

[3] https://matplotlib.org/.

[4] Alfredo Caicedo Barrero, Graciela Wagner de García, and Rosa María Méndez Parra. *Introducción a la Teoría de Grafos*. ELIZCOM SAS, 2010.

[5] Juana María Alonso Revenga. *Flujo en Redes y Gestión de Proyectos. Teoría y Ejercicios Resueltos.* Netbiblo, 2008.