

Tarea 1 Optimización de Flujo en Redes

Orlando Lázaro Ruiloba Torres

11 de febrero de 2019

1. Grafo simple no dirigido acíclico

Un ejemplo de la aplicación práctica de un grafo simple no dirigido acíclico, pudiera ser el organigrama de una empresa o institución, de acuerdo a [1], ya que este constituye una representación gráfica de la estructura de la misma, donde los vértices o nodos son los puestos de trabajo, y los arcos representan las relaciones existentes entre ellos ya sea directa o indirectamente.

El código en Python se muestra a continuación:

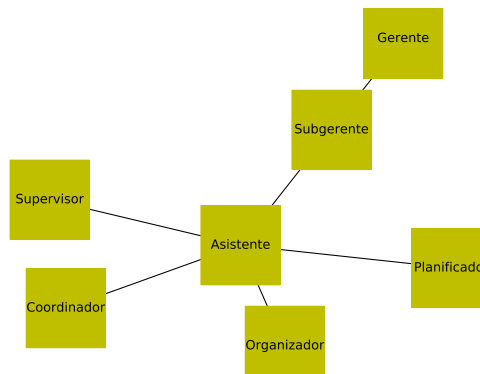
```
import networkx as nx
import matplotlib.pyplot as plt

GNDA = nx.Graph()

GNDA.add_node("Gerente")
GNDA.add_nodes_from(["Subgerente", "Asistente", "Coordinador", "Planificador", "Organizador", "Supervisor"])

GNDA.add_edges_from([("Gerente", "Subgerente"), ("Subgerente", "Asistente")])
GNDA.add_edges_from([("Asistente", "Coordinador"), ("Asistente", "Planificador")])
GNDA.add_edges_from([("Asistente", "Organizador"), ("Asistente", "Supervisor")])

nx.draw_spring(GNDA, node_size = 5500, node_color = 'y', node_shape = 's', with_labels=True)
plt.draw()
plt.savefig("GNDA.eps")
plt.show()
```



2. Grafo simple no dirigido cíclico

Un ejemplo práctico de la aplicación de este tipo de grafo, pudiera ser la representación de una red de distribución directa según [2], donde los productos salen de la fábrica y van directamente al consumidor final, los vértices serían los clientes y la fábrica, y los arcos el camino que los une, debiendo volver finalmente al vértice que representa el punto inicial (fábrica), repitiendo este proceso tantas veces como sea necesario, sin seguir un orden específico.

El código en Python se muestra a continuación:

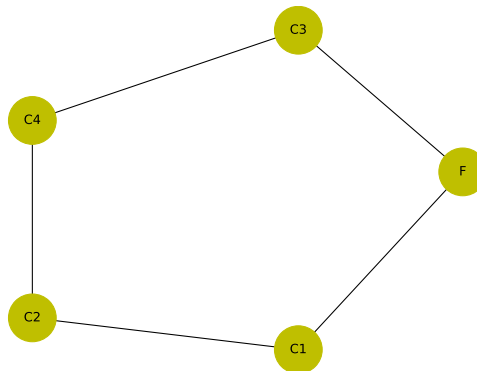
```
import networkx as nx
import matplotlib.pyplot as plt

GNDC = nx.Graph()

GNDC.add_node("F")
GNDC.add_nodes_from(["C1", "C2", "C3", "C4"])

GNDC.add_edges_from([("F", "C3"), ("C3", "C4"), ("C4", "C2"), ("C2", "C1"), ("C1", "F")])

nx.draw_spectral(GNDC, node_size = 2000, node_color = 'y', node_shape = 'o', with_labels=True)
plt.draw()
plt.savefig("GNDC.eps")
plt.show()
```



3. Grafo simple no dirigido reflexivo

Para este caso, un ejemplo en la práctica sería, de acuerdo a [2] una red de carreteras que interconecta diferentes ciudades, donde los vértices serían las ciudades y los arcos las carreteras que las unen.

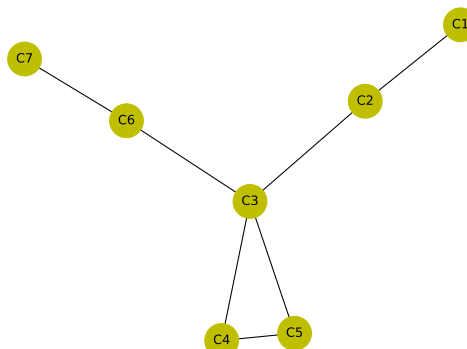
El código en Python es:

```
import networkx as nx
import matplotlib.pyplot as plt

GNDR = nx.Graph()

GNDR.add_nodes_from(["C1", "C7"])
GNDR.add_edges_from([("C1", "C2"), ("C2", "C3"), ("C3", "C4"), ("C4", "C5"), ("C5", "C3"), ("C3", "C6"),
                    ("C6", "C7"), ("C7", "C7")])

nx.draw(GNDR, node_size = 1000, node_color = 'y', node_shape = 'o', with_labels=True)
plt.draw()
plt.savefig("GNDR.eps")
plt.show()
```



4. Grafo simple dirigido acíclico

Un ejemplo para este caso en particular, pudiera ser de acuerdo a [2], una estructura jerárquica de datos, donde los vértices corresponden a entidades y los arcos a las relaciones existentes entre las mismas.

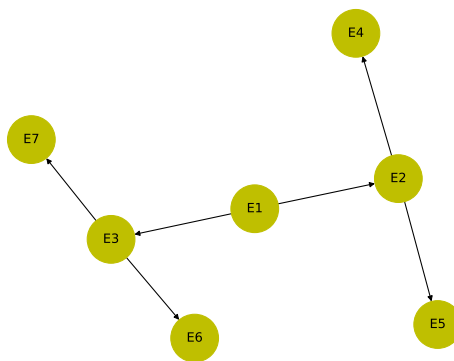
El código en Python es el siguiente:

```
import networkx as nx
import matplotlib.pyplot as plt

GDA = nx.DiGraph()

GDA.add_nodes_from(["E1", "E7"])
GDA.add_edges_from([("E1", "E2"), ("E1", "E3"), ("E2", "E4"), ("E2", "E5"), ("E3", "E6"), ("E3", "E7")])

nx.draw(GDA, node_size = 2000, node_color = 'y', node_shape = 'o', with_labels=True)
plt.draw()
plt.savefig("GDA.eps")
plt.show()
```



5. Grafo simple dirigido cíclico

Un ejemplo en la práctica de este tipo de grafo de acuerdo a [2], puede ser la representación del proceso para la confección en masa de un determinado producto en una fábrica, ya que esto implica el desarrollo de varios subprocesos que tienen una secuencia determinada y que se van a repetir un número específico de veces, donde los vértices serían los subprocesos y los arcos la relación ordenada existente entre ellos.

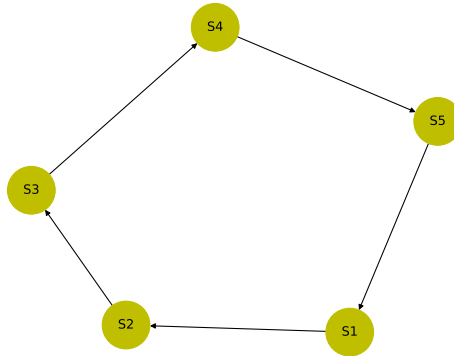
El código en Python se presenta a continuación:

```
import networkx as nx
import matplotlib.pyplot as plt

GDC = nx.DiGraph()

GDC.add_nodes_from(["S1", "S5"])
GDC.add_edges_from([("S1", "S2"), ("S2", "S3"), ("S3", "S4"), ("S4", "S5"), ("S5", "S1")])

nx.draw(GDC, node_size = 2000, node_color = 'y', node_shape = 'o', with_labels=True)
plt.draw()
plt.savefig("GDC.eps")
plt.show()
```



6. Grafo simple dirigido reflexivo

Un ejemplo en la práctica de este tipo de grafo se vería reflejado de acuerdo a [2], en una red de aviación donde los vértices serían las ciudades o países a visitar y los arcos representan el camino recorrido por el avión.

El código en Python es:

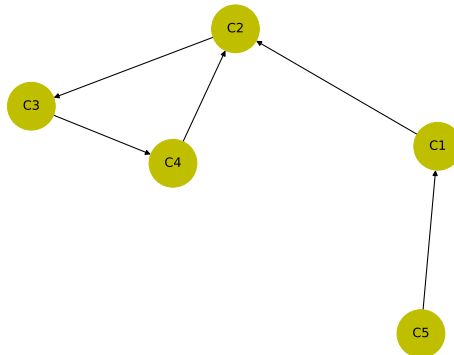
```

import networkx as nx
import matplotlib.pyplot as plt

GDR = nx.DiGraph()

GDR.add_nodes_from(["C1", "C5"])
GDR.add_edges_from([("C1", "C2"), ("C2", "C3"), ("C3", "C4"), ("C4", "C2"), ("C5", "C1"), ("C5", "C5")])

nx.draw(GDR, node_size = 2000, node_color = 'y', node_shape = 'o', with_labels=True)
plt.draw()
plt.savefig("GDR.eps")
plt.show()
  
```



7. Multigrafo no dirigido acíclico

Un ejemplo en la práctica de este tipo de multigrafo según lo expuesto en [4] pudiera ser el caso de un museo, que desarrolla una exposición en varias salas al unísono, donde la situación ideal sería poder transitar por todas las salas sin tener que pasar necesariamente dos veces por la misma, se consideran las salas como los vértices, y las aristas, los caminos que las unen.

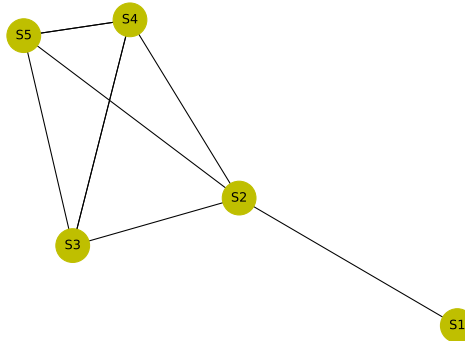
El código en Python es:

```
import networkx as nx
import matplotlib.pyplot as plt

MND A = nx.MultiGraph()

MND A.add_nodes_from(["S1", "S5"])
MND A.add_edges_from([("S1", "S2"), ("S2", "S3"), ("S2", "S4"), ("S2", "S5"), ("S3", "S4"), ("S3", "S4"), ("S4", "S5"),
                      ("S4", "S5"), ("S3", "S5")])

nx.draw(MND A, node_size = 1000, node_color = 'y', node_shape = 'o', with_labels=True)
plt.draw()
plt.savefig("MND A.eps")
plt.show()
```



8. Multigrafo no dirigido cíclico

Un ejemplo práctico para este caso en particular, se evidencia de acuerdo a [3] en la generación de redes tipo pequeño mundo, concepto que de hecho es muy común y nada alejado de nuestras experiencias diarias, ya que poco después de conocer a un extraño, uno se da cuenta de que tiene un amigo en común con él, este concepto se basa en el principio de los seis grados de separación, que establece que entre cualesquiera dos personas del mundo, existe una media de seis conexiones de amistad.

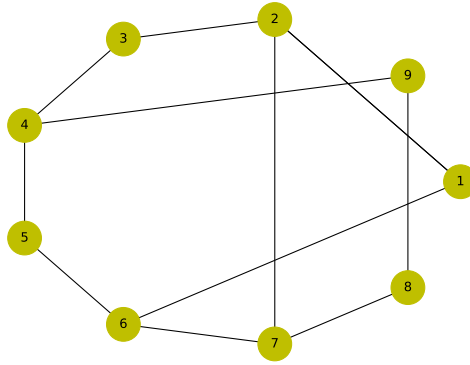
El código en Python es:

```
import networkx as nx
import matplotlib.pyplot as plt

MNDC = nx.MultiGraph()

MNDC.add_nodes_from(["1", "9"])
MNDC.add_edges_from([("1", "2"), ("2", "3"), ("3", "4"), ("4", "5"), ("5", "6"), ("6", "7"), ("7", "8"), ("8", "9"),
                     ("2", "1"), ("1", "6"), ("2", "7"), ("4", "9")])

nx.draw_circular(MNDC, node_size = 1000, node_color = 'y', node_shape = 'o', with_labels=True)
plt.draw()
plt.savefig("MNDC.eps")
plt.show()
```



9. Multigrafo no dirigido reflexivo

Un ejemplo de este tipo de multigrafo en la vida cotidiana puede ser según lo expuesto en [7] una red social de personas, donde los vértices pueden representar hombres o mujeres, personas con diferentes nacionalidades, edades, ingresos, etc y los arcos pueden simbolizar amistad, proximidad geográfica, conocimiento profesional, entre otros.

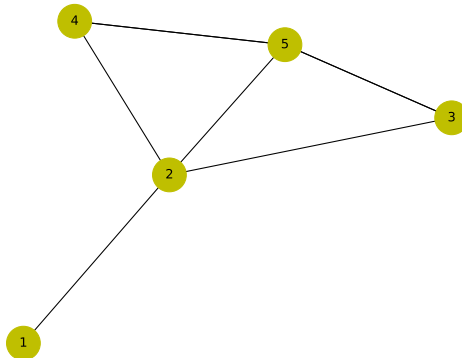
El código en Python es:

```
import networkx as nx
import matplotlib.pyplot as plt

MNDR = nx.MultiGraph()

MNDR.add_nodes_from(["1", "5"])
MNDR.add_edges_from([("1", "2"), ("2", "3"), ("2", "4"), ("2", "5"), ("3", "5"), ("4", "5"), ("3", "5"),
                    ("4", "5"), ("5", "5")])

nx.draw(MNDR, node_size = 1000, node_color = 'y', node_shape = 'o', with_labels=True)
plt.draw()
plt.savefig("MNDR.eps")
plt.show()
```



10. Multigrafo dirigido acíclico

Un ejemplo de una aplicación práctica de este tipo de multigrafo se pone de manifiesto en una pequeña red de electricidad entre cinco ciudades, donde los vértices serían las ciudades y el tendido eléctrico serían las aristas que las unen, en el caso de C4, recibe la energía a través de C3 en dos canales que alimentan a la totalidad de la ciudad.

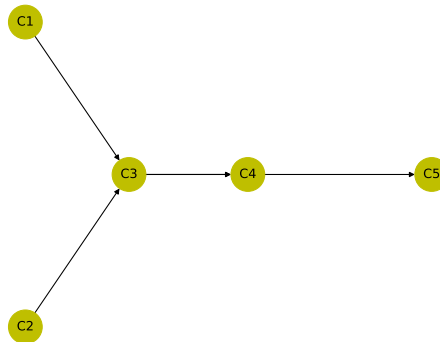
El código en Python es:

```
import networkx as nx
import matplotlib.pyplot as plt

MDA = nx.MultiDiGraph()

MDA.add_nodes_from(["C1", "C5"])
MDA.add_edges_from([("C1", "C3"), ("C2", "C3"), ("C3", "C4"), ("C3", "C4"), ("C4", "C5")])

nx.draw_spectral(MDA, node_size = 1000, node_color = 'y', node_shape = 'o', with_labels=True)
plt.draw()
plt.savefig("MDA.eps")
plt.show()
```



11. Multigrafo dirigido cíclico

Un ejemplo de la puesta en práctica de este tipo de multigrafo pudiera ser según [5] la modelación de 5 estaciones de una red de metro, donde cada estación da lugar a un vértice y los arcos simbolizan el camino de ida y vuelta entre cada una de las estaciones.

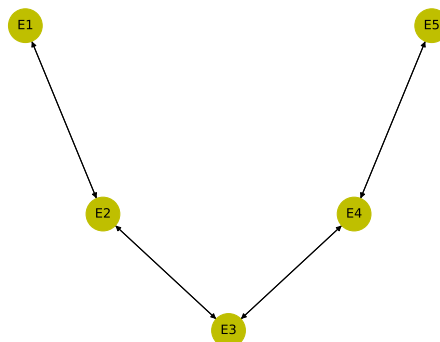
El código en Python es:

```
import networkx as nx
import matplotlib.pyplot as plt

MDC = nx.MultiDiGraph()

MDC.add_nodes_from(["E1", "E5"])
MDC.add_edges_from([("E1", "E2"), ("E2", "E1"), ("E2", "E3"), ("E3", "E2"), ("E3", "E4"), ("E4", "E3"),
                    ("E4", "E5"), ("E5", "E4")])

nx.draw_spectral(MDC, node_size = 1000, node_color = 'y', node_shape = 'o', with_labels=True)
plt.draw()
plt.savefig("MDC.eps")
plt.show()
```



12. Multigrafo dirigido reflexivo

Un ejemplo de la puesta en práctica de este tipo de multigrafo pudiera ser según se muestra en [6] una red de comunicación en donde los vértices son los dispositivos de red y los arcos son los medios de networking (medios de transmisión basados en cobre o fibra óptica) que aseguran el envío y recepción de los mensajes y la información a través de la red de datos.

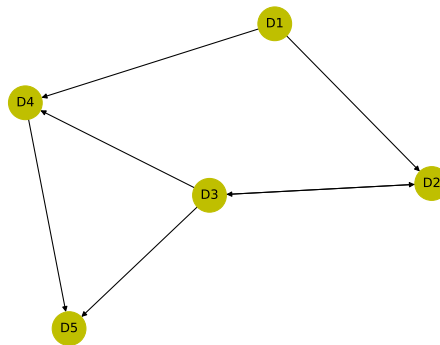
El código en Python es:

```
import networkx as nx
import matplotlib.pyplot as plt

MDR = nx.MultiDiGraph()

MDR.add_nodes_from(["D1", "D5"])
MDR.add_edges_from([("D1", "D2"), ("D1", "D4"), ("D2", "D3"), ("D3", "D2"), ("D3", "D4"), ("D3", "D5"), ("D4", "D5"),
                    ("D5", "D5")])

nx.draw(MDR, node_size = 1000, node_color = 'y', node_shape = 'o', with_labels=True)
plt.draw()
plt.savefig("MDR.eps")
plt.show()
```



Referencias

- [1] Universidad de Valladolid. Grafos no dirigidos acíclicos: Árboles. 2008.
- [2] Prof. Kryscia Daviana Ramírez Benavides. Teoría de grafos y Árboles. Universidad de Costa Rica. Escuela de Ciencias de la Computación e Informática.
- [3] J.Torres. Tema: 2 red compleja como ejemplo de sistema complejo. 2018.
- [4] Hernán S. Nottoli. Teoría de grafos. aplicaciones al diseño arquitectónico. 1998.
- [5] Alfonso Recuero. Aplicaciones de la teoría de grafos: Búsqueda de caminos en una red y análisis de su conectividad.
- [6] C. Ortiz y J. Sendra E. Martín, A. Méndez. Tema 5: Grafos. 2011.
- [7] Nidia Lizzeth Gómez Duarte. Efectos estructurales en la sincronización de sistemas. Master's thesis, Universidad Autónoma de Nuevo León, 2010.