

Visualización de grafos simple, ponderados y dirigidos con gnuplot

Yessica Reyna Fernández

Flujo en Redes

4 de marzo de 2018

1. Introducción

En la práctica se realizó la diferenciación entre grafo simple, *grafo dirigido* y *grafo ponderado*. Un *grafo dirigido* es un grafo en el cual las aristas entre cada par de nodo tiene una orientación o sentido; mientras que un *grafo ponderado* es un grafo en el cual se ha añadido un peso a cada una de las aristas en él.

Además de que a partir de la modificación del código en `python` el usuario pueda determinar el tipo de grafo, ya sea simple, solamente ponderado, o que sea dirigido o la combinación de estas dos opciones, usando `gnuplot` para generar imágenes en un formato `eps`. Teniendo como trabajo previo un reporte que se encuentra en el mismo repositorio.

2. Grafo Simple

Con la elaboración de un *grafo simple* el cual existe si a lo sumo hay una arista que une a un par de nodos específicos.

Se realiza la modificación del código a una estructura de clases para un mejor orden establecido dentro del código.

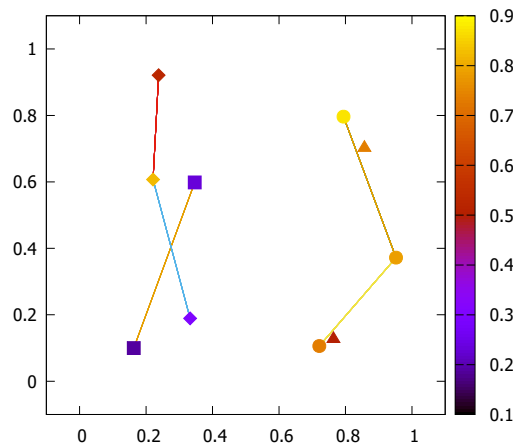


Figura 1: Grafo simple.

3. Clases y funciones

Ademas de tener consideraciones adicionales para el código ya usado para la creación de un grafo simple:

1. Dirección en las aristas para la modificación de un grafo simple a la construcción de un grafo dirigido.
2. Consideración de los tipos de línea para la determinación de la ponderación de cada arista, denotando cinco tipos diferentes, en donde el tipo de línea determina el valor del peso en un rango [1,5] con números enteros.

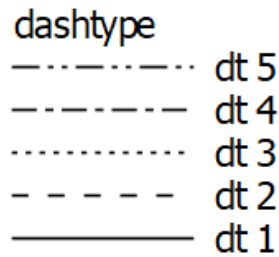


Figura 2: Tipos de líneas.

Para la distribución de la estructura del código se tiene una clase **GrafoYessica** con las siguientes funciones:

- **init** donde se inicializarán los parámetros del grafo.

```
1 n = 20 #cambia cantidad de nodos
2 aristas = []
3 grafoorientado = 0
4 grafocapacitado = 0
5 G = GrafoYessica()
6 G.nodoscrear(n)
7 G.conectar(n)
```

- **nodoscrear** la cual cumple con la función de dar valores a las características de los nodos.
- **conectar** crea las aristas para hacer la unión de los nodos que se deban conectar, ademas de generar el archivo que se lee por **gnuplot**.

Donde con las siguientes líneas de código se define si habrá o no dirección en las aristas del grafo sin ninguna otra consideración adicional en ellas:

```
1 grafoorientado=1 #para que sea dirigido
```

en la parte del código que va fuera de la clase se define el valor de esa variable, después con lo siguiente se determina en **gnuplot** la dirección y las propiedades de la arista. Para una mayor referencia del resultado véase la figura 3.

```
1 print("set arrow {d} from {f}, {f} to {f}, {f} head filled size 0.04,7 lt {d}"
      .format(num, self.x[num], self.y[num], self.x[num+1], self.y[num+1], f, v),
      file =archivo)
```

Y para la obtención de ponderación, como se había mencionado anteriormente se buscará el cambio de tipo de línea en el grafo. Lo cual se puede apreciar con las siguientes líneas de código, obteniendo resultados como los que se verán en la figura 4.

```

1 if grafocapacitado is 1:
2     num = 1
3     for a in aristas:
4         (self.x[num], self.y[num], self.x[num+1], self.y[num+1], f, v) = a
5         print("set_arrow_{:d}_from_{:f},_{:f}_to_{:f},_{:f}_nohead_lt_{:d}_dashtype_
6             {:d}").format(num, self.x[num], self.y[num], self.x[num+1], self.y[num+1], f,
            v), file=archivo)
7         num += 1

```

Y para la combinación de ambos tipos de grafo, dirigido y ponderado se usaron las siguientes líneas del código, resaltando que la parte descrita en verde contiene características particulares para la dirección y tipo de línea:

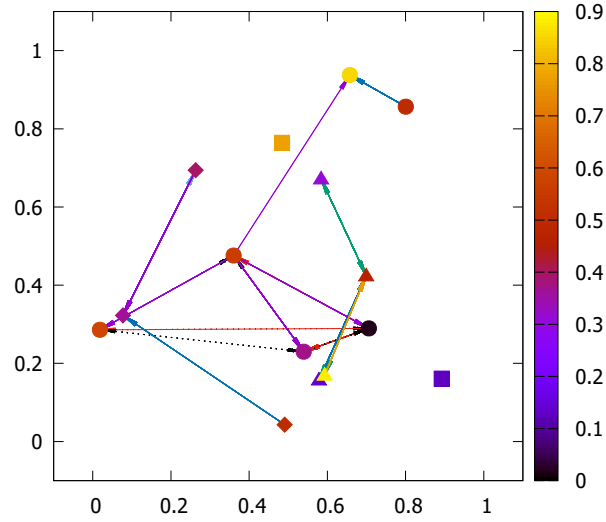
```

1 if grafoorientado is 1:
2     if grafocapacitado is 1:
3         num = 1
4         for a in aristas:
5             (self.x[num], self.y[num], self.x[num+1], self.y[num+1], f, v) = a
6             print("set_arrow_{:d}_from_{:f},_{:f}_to_{:f},_{:f}_head_filled_size_0.04,8_
7                 lt_{:d}_dashtype_{:d}").format(num, self.x[num], self.y[num],
                    self.x[num+1], self.y[num+1], f, v), file=archivo)
8             num += 1

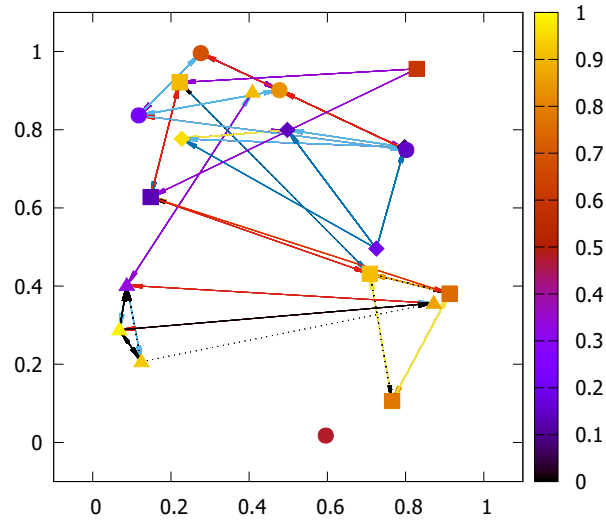
```

4. Grafo Dirigido

Dentro del grafo simple se puede estructurar lo que es el grafo dirigido ya que a partir del grafo simple se le da la dirección a las aristas que conectan a los pares de nodos. Haciendo la relación de que la dirección denota que hay comunicación entre el conjunto de personas denotadas por los nodos en el grafo. Obteniendo algunos resultados para diferentes tamaños de nodos en la siguiente figura.



(a) $n = 15$ nodos.

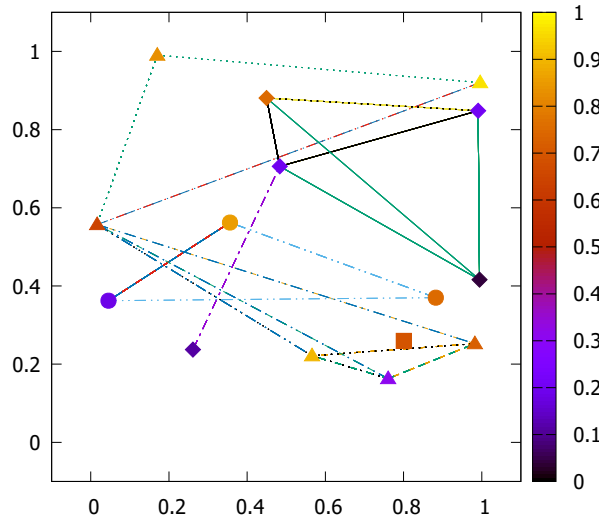


(b) $n = 20$ nodos.

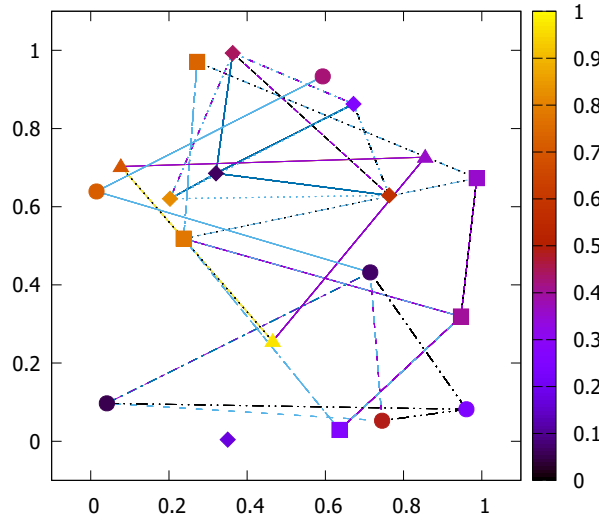
Figura 3: Grafos dirigidos.

5. Grafo Ponderado

De la creación del grafo simple, se elaboran las modificaciones adecuadas ya descritas en la sección 3, con las cuales se obtienen resultados para un grafo ponderado, sin dirección o alguna otra cualidad como en las figuras de abajo; donde el tipo de línea indicaría que tan profunda será la relación entre el par de nodos donde el tipo uno es íntimamente relacionado y el tipo cinco es que no hay un apego a la persona.



(a) $n = 15$ nodos.

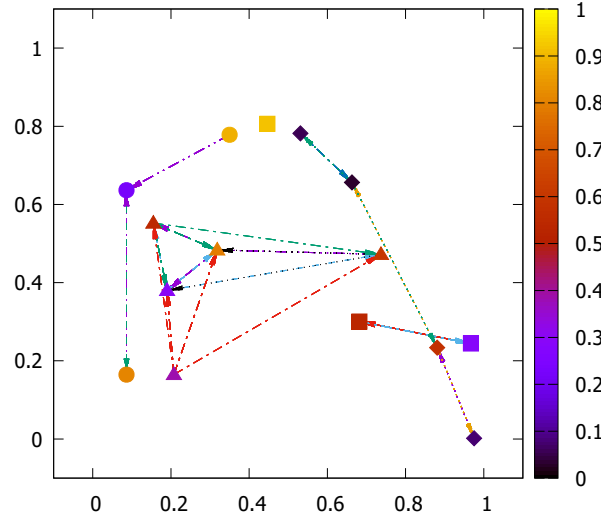


(b) $n = 20$ nodos.

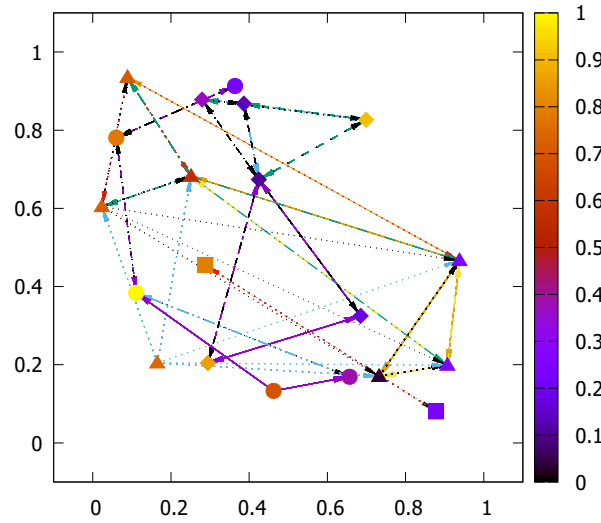
Figura 4: Grafos dirigidos.

6. Grafo Dirigido y Ponderado

Con la combinación de los tipos de grafos vistos anteriormente se puede crear este tipo de grafo que contenga dirección en sus aristas y una ponderación formada por el tipo de línea que tiene cada arista. Siendo que con el resultado de dirección y ponderación juntos se puede tener como consecuencia el tipo de relación que hay entre cada par de personas y si hay comunicación entre ellas o no.



(a) $n = 15$ nodos.



(b) $n = 20$ nodos.

Figura 5: Grafos dirigidos.