Tarea 1
Optimización de flujo en redes.

Beatriz Alejandra García Ramos

A 19 de Febrero de 2018

1. Introducción.

En esta práctica se desea enriquecer el conocimiento del programa `python` con un enfoque hacia la realización de grafos. Lo que se requiere es que mediante un código realizado en este programa se puedan generar archivos que después se procesen en el programa `gnuplot` que es un apoyo visual de nuestros resultados.

Un grafo puede ser o no dirigido, pero en esta práctica se hará un enfoque a grafos sin dirección, lo que se requiere es un código en el que se incluyan nodos que se conectarán mediante cierto criterio con algunos otros nodos, con uno, o con ninguno, dependiendo del criterio que se tome.

Lo que se desea realizar son dos tipos de grafos, el primero es un grafo cíclico y el segundo un grafo bosque, el cual es un conjunto de árboles y tiene como propiedad que los nodos busquen distancias cercanas a ellos.

2. Creación de un grafo en `python`.

Para la realización de los grafos principalmente se requieren nodos y aristas, los nodos son aquellos puntos en nuestro espacio que serán conectados o no dependiendo del criterio que se les indique. Las conexiones que se tienen entre ellos son las aristas, pueden ser de manera aleatoria o con alguna decisión que se indique en el código.

Para la práctica primero se realizaron pruebas en donde se tenía que los nodos eran conectados de manera aleatoria unos con otros, se indicaba una probabilidad p de existencia de una arista y era comparada con cifras aleatorias, si se cumplía el criterio de que la cifra aleatoria era menor a la probabilidad dada entonces los nodos comparados eran conectados por una arista.

Una vez que se realizó esa prueba se introdujeron y modificaron ciertas funciones. El proceso para realizar un grafo es primeramente crear los nodos en nuestro espacio, el cual es de dimensión 1×1, los nodos son creados con un ciclo `for` que nos ayuda a hacer tantos nodos como queramos, cada vez que se crea un nodo se va guardando en un archivo y el nodo se construye con dos variables, las cuales se toman de manera aleatoria un valor entre cero y uno y se hacen pares ordenados con estas variables. Una vez que tenemos un vector en donde se guardan estos nodos además de que se guarden en el archivo se hace el emparejamiento de los nodos, para los grafos que se tomarán en esta práctica se implementaron criterios distintos.

Para el primer tipo de grafo que es el cíclico se realizaron dos criterios, el primero fue que el último nodo creado debía ser conectado con el primer nodo para que cuando se hiciera el ciclo `for` que une al resto de los nodos fuera más sencillo realizar el trabajo. Una vez que se conectó el último nodo con el primero se llevó a cabo el ciclo `for` que determina que el primer nodo se conecta con el segundo nodo, el segundo con el tercero y así sucesivamente hasta llegar al nodo $n-1$, el cual se conecta con el nodo n. Éste es un tipo de grafo particular de los grafos cíclicos ya que se tiene que las conexiones son de manera secuencial.

Para el segundo tipo de grafo lo que se requiere hacer es encontrar para el último nodo el nodo más cercano a él y emparejarlo con éste. Después creando un ciclo `for` para cada uno de los nodos se buscan todas las distancias de los demás nodos a él, se localiza el mínimo de todas estas distancias

y se verifica que la arista creada entre estos dos nodos no se encuentre ya en la lista de aristas, si ya se encuentra se elimina la mínima distancia del nodo en proceso y teniendo en cuenta el resto de las distancias vuelve a buscar el mínimo y se crea la arista correspondiente, si no se encuentra la arista dentro de la lista entonces se tendrá solamente la instrucción de incluirla en ella. Sin embargo se tiene el problema de que pueden existir aristas repetidas, ya que en la decisión del código ir, por ejemplo, del nodo uno al nodo dos no es lo mismo que ir del nodos dos al uno, lo que ocasiona que esa arista se incluya dos veces ya que este grafo no es dirigido, por lo que se modificó el código de manera que no existieran repeticiones de aristas, para lo cual se realiza una manera distinta de abordar el problema: primero se calculan las distancias de todos los nodos hacia todos los nodos, es decir, se tendrán n^2 distancias y se colocan en una lista llamada `dis`, después nodo por nodo se calculan de nuevo las distancias hacia los otros nodos con un ciclo `for` y se tiene que en cada etapa si el mínimo de las distancias calculadas para el nodo en esa etapa se encuentra en la lista `dis` este nodo se conectará con el nodo correspondiente a la distancia calculada y ésta se eliminará de la lista `dis` para que cuando se realicen las siguientes etapas del ciclo `for` no se tome en cuenta la misma arista más de una vez, para el caso en que se calculen las distancias de de un nodo a sí mismo se tendría que la distancia es cero, entonces con el criterio que se está realizando las conexiones serían de los nodos consigo mismos, para evitar ésto se coloca un valor muy grande de la distancia en la lista `dis`.

3. Resultados.

Una vez que se realizó el código para el grafo cíclico se tenía que cuando la cantidad de nodos aumentaba no se veía con claridad cuáles nodos estaban conectados, por lo que se tomaron en cuenta solamente diez nodos para observar cómo se maneja el trabajo, como se muestra en la figura 1.

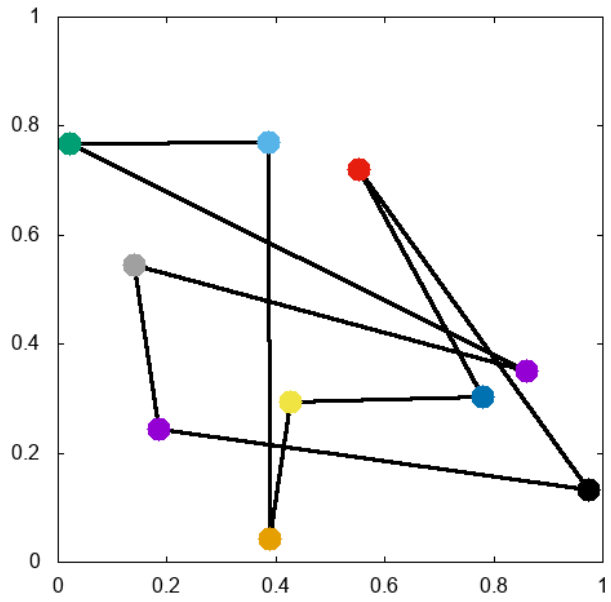


Figura 1: Grafo cíclico con 10 nodos.

Dado que el grafo cíclico está definido en el código con orden creciente se tienen los distintos tonos que `gnuplot` utiliza para el coloreo de los nodos los cuales se encuentran definidos en el cuadro 1. El color gris no se encuentra definido en el coloreo pero se localiza como el último nodo en el grafo.

Dado que no se tiene una buena visualización del grafo cíclico con los criterios que se tomaron en cuenta se decidió realizar los cambios del código para que se cumplieran otras características como el hecho de que se busquen los nodos cercanos y se conecten con ellos.

Por tanto, cuando se realizaron los cambios y se hizo la elección de los nodos cercanos se tomó en

Número	Color
1	morado
2	verde
3	azul celeste
4	naranja
5	amarillo
6	azul rey
7	rojo
8	negro

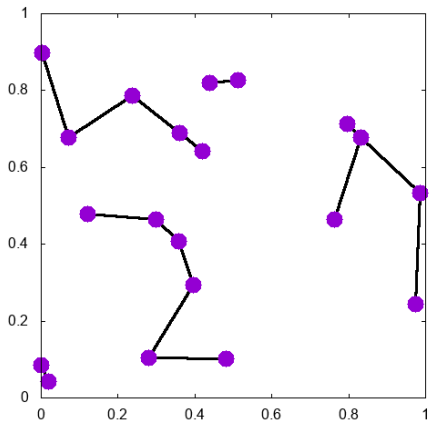
Cuadro 1: Asignación de colores en `gnuplot`.

cuenta que no todos los nodos estarían conectados entre sí, lo que provoca un grafo bosque, el cual está conformado con dos o más árboles no conexos entre sí.

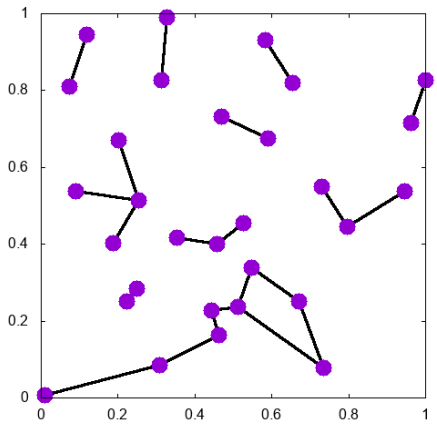
En la figura 2 se pueden observar estos grafos con variaciones en la cantidad de nodos, aún cuando existen más de ellos se tiene que ninguno queda sin conectarse con otro, todos los nodos tienen al menos un nodo con el cual se conectan, sin embargo en ese caso las aristas se repiten y si se repiten el tiempo de ejecución tanto en el código como en el gráfico es mayor y podría ocasionar problemas cuando se tengan casos donde los nodos sean muchos.

Es por eso que se hicieron modificaciones para que no existieran aristas repetidas que ayudarán a que solamente una vez se logren conectar los nodos entre sí pasando por el mismo camino. Los resultados que se obtuvieron al evaluar estos cambios están en la figura 3. Como se puede observar, no existe gran diferencia entre estos tipos de grafos y los que se tenían cuando las aristas sí estaba repetidas, lo cual nos indica que aunque se realizaron modificaciones en el código la esencia del grafo sigue siendo la misma y los resultados están presentes en esos gráficos con distinta cantidad de nodos.

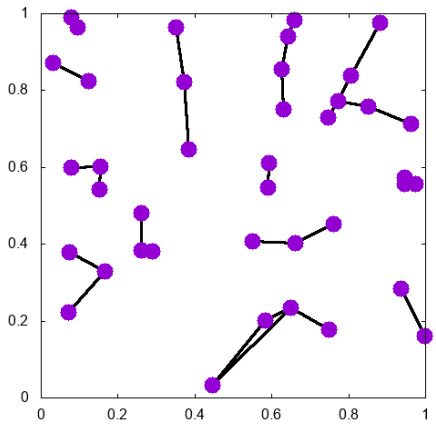
Se pueden hacer más variaciones en el código ya que existen muchos tipos de grafos. En los códigos que se realizaron para la práctica se obtuvieron resultados deseados, existen diversas aplicaciones que se les pueden dar a los ciclos como por ejemplo en ruteo cuando se debe pasar una sola vez por todos los caminos visitando a todos los clientes que requieren del servicio de alguna empresa, o los grafos de bosque que nos ayudan a visualizar por ejemplo las conexiones que tienen las personas en las redes sociales.



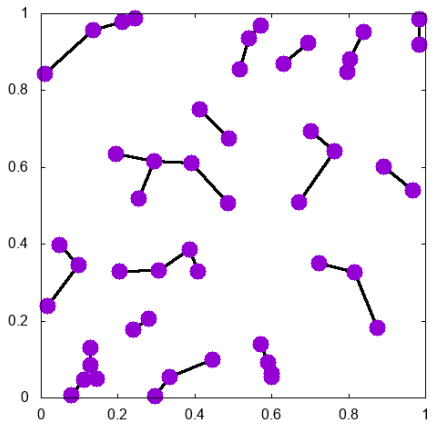
(a) 20 nodos



(b) 30 nodos

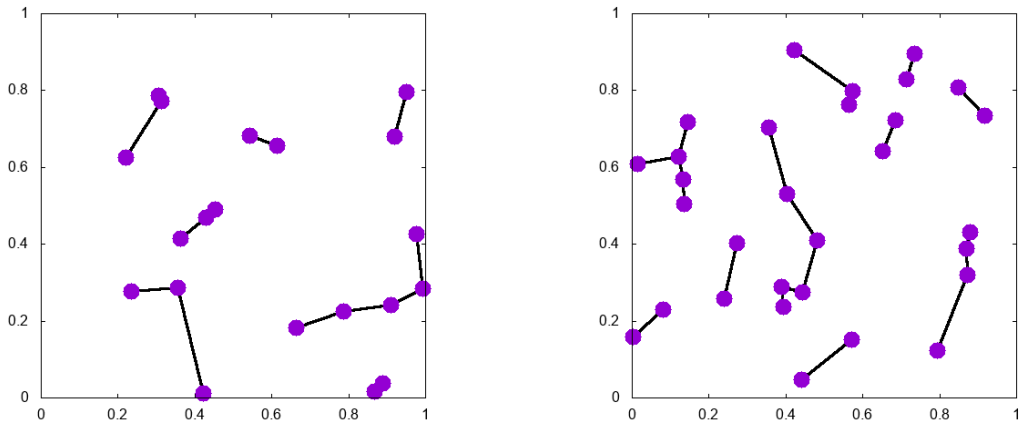


(c) 40 nodos



(d) 50 nodos

Figura 2: Grafos de bosque con variaciones en la cantidad de nodos.



(a) 20 nodos

(b) 30 nodos

(c) 40 nodos

(d) 50 nodos

Figura 3: Grafos de bosque con variaciones en la cantidad de nodos.