

Experimentos con el algoritmo de Floyd-Warshall en python

Francisco Gerardo Meza Fierro

1. Introducción

En esta práctica se estudiarán tanto las distancias que hay entre los nodos de un grafo propuesto como la densidad que hay en el mismo. El grafo es creado en base a métodos usados en prácticas de este mismo repositorio [1], tal como la función de Floyd-Warshall; por último se medirán los tiempos de ejecución del código y se discutirán los resultados.

Igual que en la previa práctica, las imágenes que presenta ésta práctica fueron obtenidas a través de `gnuplot` y `R` mediante archivos generados a través de `python`.

2. Creación de los nuevos grafos

A diferencia de los grafos anteriores, los grafos propuestos para la presente práctica no pinta los nodos de manera aleatoria, sino que primero fija un centro en el espacio, se crea una círculo imaginario con ese centro y sobre la circunferencia del mismo se pintan los nodos, todos separados a una misma distancia.

Para conectar los nodos, se usa un parámetro k : si k es igual a 1, los nodos se conectan con los que estén a un nodo de distancia, si k es igual a 2, se conectan con los que estén a dos nodos de distancia y así sucesivamente.

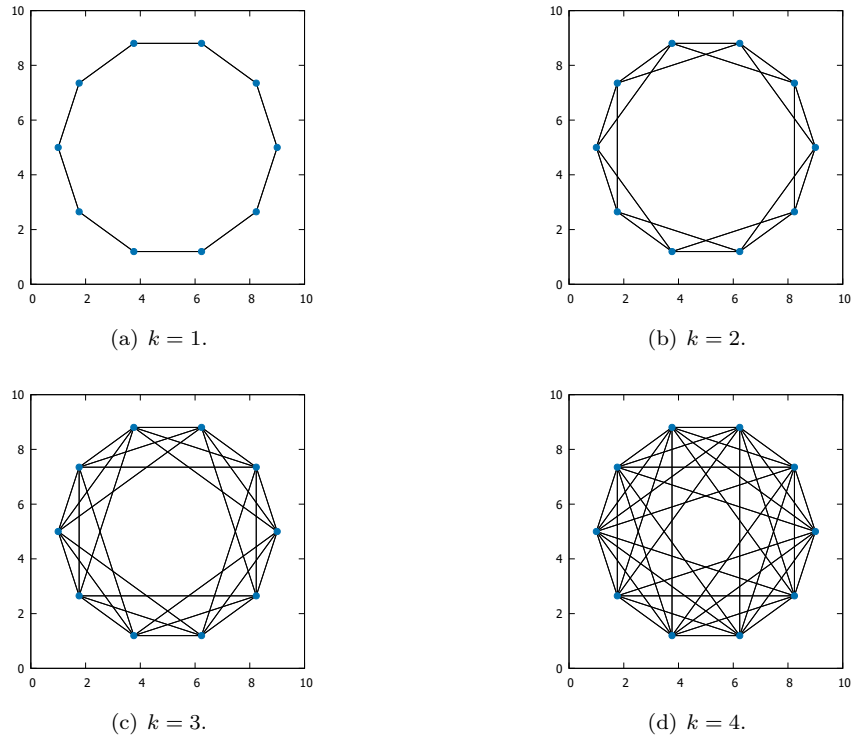


Figura 1: Grafo de 10 nodos con distintos valores de k .

Además de esas conexiones, los grafos tienen la probabilidad p de que, de manera aleatoria, dos de sus nodos se conecten entre ellos creando una arista adicional a las creadas con el parámetro k .

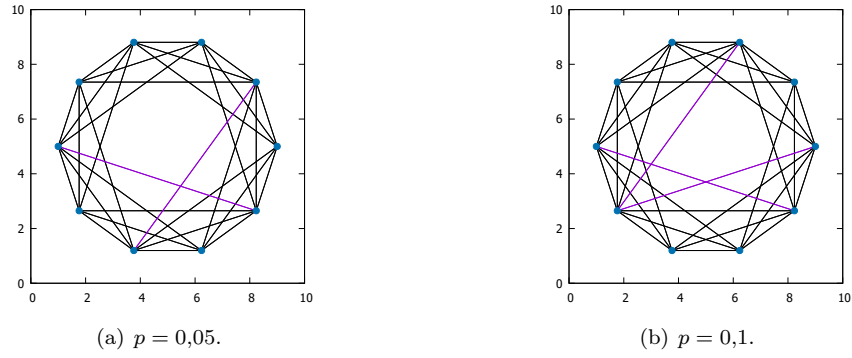


Figura 2: Grafo de 10 nodos con distintos valores de p .

Las aristas en color morado representan las aristas que fueron agregadas con la probabilidad p . Para cada arista creada, la ponderación que se le asignará es la distancia euclidiana entre ambos nodos.

3. Densidades y distancias

Para calcular las densidades de los grafos se procede de la siguiente manera:

- Se toma un nodo (x, y) del grafo y se encuentran sus vecinos.
- Para cada uno de los vecinos, se encuentran su respectivos vecinos.
- Se suman todas las aristas que conecten a estos segundos vecinos con los vecinos del nodo (x, y) siempre y cuando estos segundos vecinos también sean vecinos del nodo (x, y) .
- La suma obtenida se divide entre el número total posible de aristas que se pueden conectar dependiendo del número de vecinos que (x, y) tiene.
- Se repite para el siguiente nodo del grafo.

Para calcular las distancias que hay entre los nodos se hace uso del algoritmo Floyd-Warshall.

Debido a que las densidades se encuentran en valores entre cero y uno y a que se desea de cierta forma comparar y graficar juntos estos resultados, se normalizan las distancias dividiéndolas entre una cota en función de la cantidad de nodos que tiene el grafo y el parámetro k , misma que se fijó en $\lfloor (3 * k) - (n/4) \rfloor$ donde k y n representan el parámetro k y la cantidad de nodos, respectivamente. Esta cota termina siendo la k máxima que cualquier grafo puede tener hasta antes que alguna arista sea repetida.

Para esta comparación se fijó la cantidad de nodos en 50 y se hizo variar el parámetro k de 1 hasta 12 y la probabilidad p de 0,05 a 0,5.

La figura 3 muestra esta comparación.

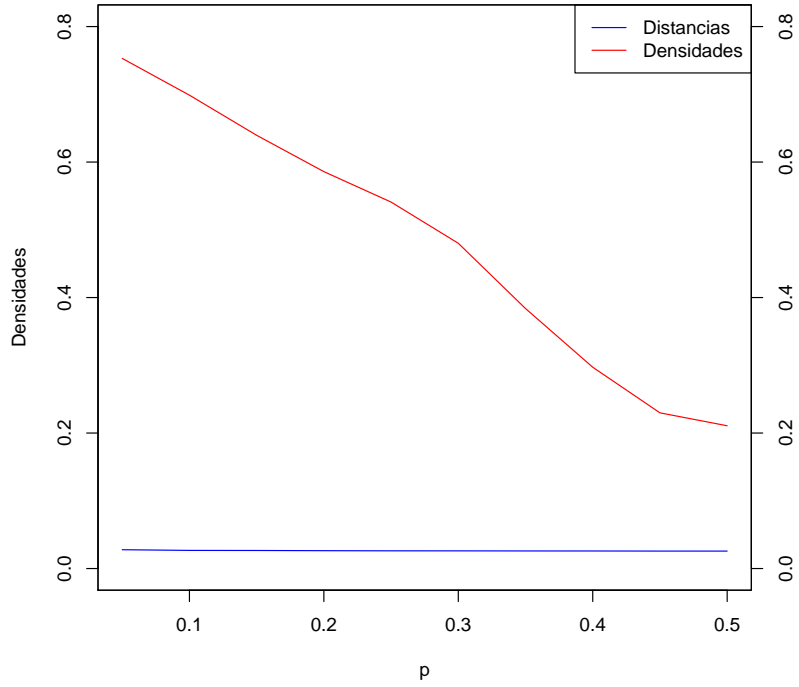


Figura 3: Comparación entre las densidades y las distancias.

4. Medición de tiempos

Para medir los tiempos, se fijó la probabilidad $p = 0,2$ y se variaron los números de nodos desde 10 hasta 100, donde para cada tamaño de grafo se consideró una $k = \lfloor n/4 \rfloor$.

La figura 4 muestra cómo fue que el tiempo de ejecución creció a medida que el tamaño de grafo fue incrementado.

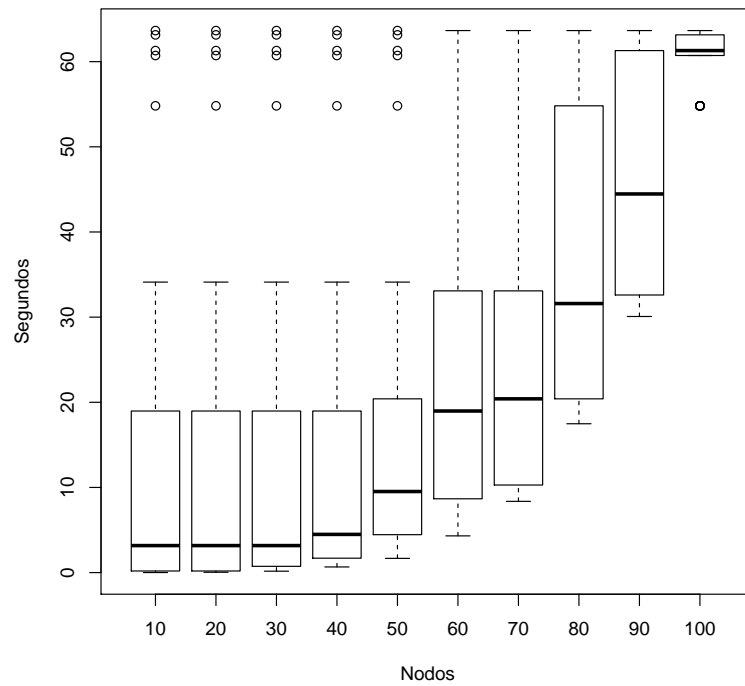


Figura 4: Incremento en los tiempos de ejecución.

Referencias

- [1] Francisco Meza. *FlujoEnRedes*
<https://github.com/Cicciofano/FlujoEnRedes>, 2018.