Tarea 3: Medición experimental de la complejidad asintótica

María Gabriela Sandoval Esquivel

Abril 2018

1 Introducción

El objetivo de esta tarea es analizar el desempeño de dos algoritmos que tienen diferente complejidad asintótica: el algoritmo Floyd-Warshall y el algoritmo Ford-Fulkerson. El primero encuentra el camino más corto entre todos los pares de nodos en un grafo y fue propuesto en 1962 por Robert Floyd [1] y Stephen Warshall [2]. El segundo encuentra el flujo máximo entre un par de nodos haciendo uso de caminos aumentantes y fue propuesto por Lester Ford y Dester Fulkerson en 1965.

La teoría de la complejidad computacional pretende medir la efectividad de los algoritmos. Existen varias herramientas de análisis para describir la efectividad de los algoritmos y éstas se basan en las operaciones elementales que se realizan en el algoritmo. Una notación comúnmente utilizada en la literatura para describir la complejidad computacional de un algoritmo es la notación de la O grande (en inglés: Big O Notation)[3]. Se dice que un algoritmo corre en tiempo $O(f(n))$ si para ciertas constantes c y n_0, el tiempo que le toma al algoritmo es a lo más $cf(n)$ para todo $n >= n_0$[3]. En esta definición se considera que n es un parámetro que indica el tamaño del problema (por ejemplo: el número de nodos del grafo).

Para interés de esta práctica, se sabe que el algoritmo de Floyd-Warshall se ejecuta en tiempo $O(n^3)$[3] y que el algoritmo de Ford-Fulkerson se ejecuta en tiempo $O(mF)$ donde m es el número de aristas del grafo y F es el flujo máximo del grafo.

En esta práctica se hace una medición experimental de la complejidad de estos algoritmos. Primero se añadieron las funciones de estos algoritmos a la clase de Grafo que se realizó para la práctica pasada. Con estas funciones se hicieron pruebas para dos tipos de grafos: dirigidos y no dirigidos, ambos con pesos en sus aristas que indican el flujo. En este reporte se incluyen las comparaciones de los tiempos de ejecución de ambas funciones para grafos de diferente tamaño. Se pretende verificar que estos resultados coincidan con la complejidad computacional de los algoritmos.

2 Programación de algoritmos

Estas funciones están basadas en las que se encuentran en la página del curso de Matemáticas Discretas [4] y adaptadas a la estructura de grafos de esta clase. La función del algoritmo de Floyd-Warshall da como resultado un diccionario que relaciona a cada par de nodos del grafo con el camino más corto entre ellos. La función del algoritmo de Ford-Fulkerson tiene como resultado el flujo máximo entre el par de nodos que recibe la función. Además, se añadieron contadores de tiempo para medir el tiempo de ejecución de cada función. De este modo, cada función tiene como segundo resultado su tiempo de ejecución.

3 Pruebas experimentales

En esta sección se presentan los resultados para los diferentes tipos de grafos: dirigidos y no dirigidos. Para cada tipo se generan grafos de diferentes tamaños de manera aleatoria: cada par de nodos se conecta con una probabilidad del 30% y con un peso aleatorio. Para cada grafo se generaron 500 instancias para hacer las pruevas de los algoritmos. A continuación se presentan gráficas de caja y bigotes para cada algoritmo y tipo de grafo. En el eje x se muestran el número de nodos que tenían las instancias y en el eje y el tiempo de ejecución en segundos.

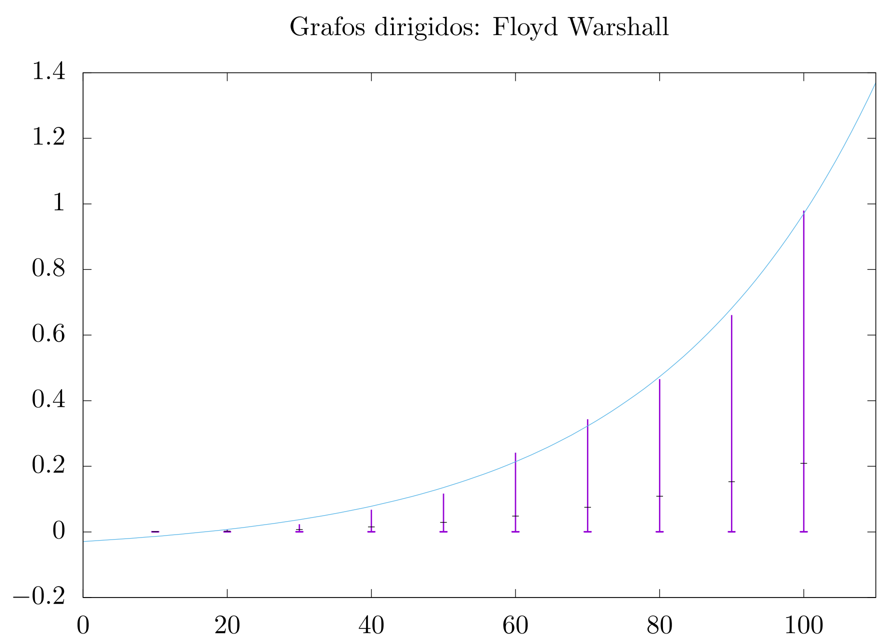

Figure 1: Diagramas de cajas y bigotes para el algoritmo de Floyd-Warshall y función exponencial ajustada a los tiempos máximos de ejecución

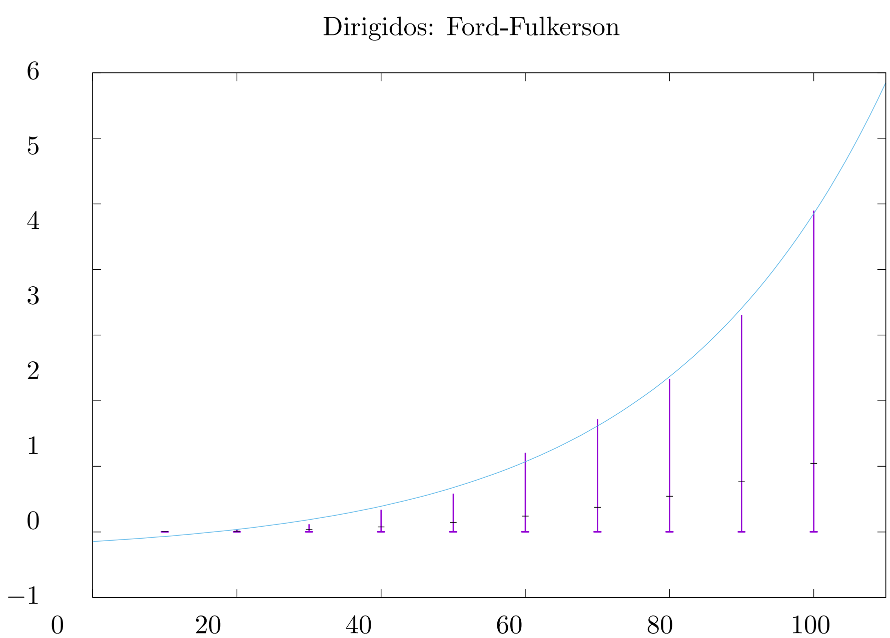

Figure 2: Diagramas de cajas y bigotes para el algoritmo de Ford-Fulkerson y función exponencial ajustada a los tiempos máximos de ejecución

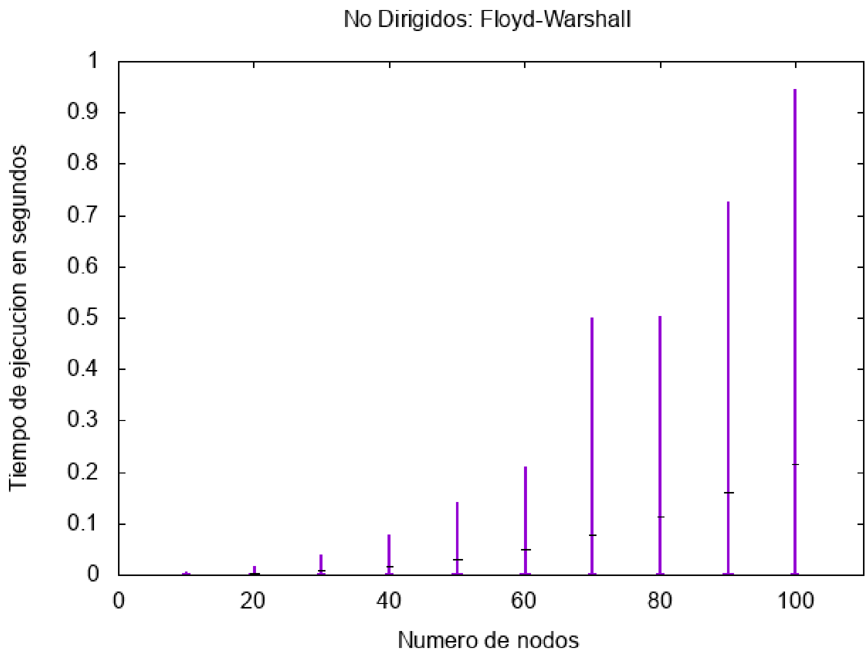

Figure 3: Diagramas de cajas y bigotes para el algoritmo de Floyd-Warshall y función exponencial ajustada a los tiempos máximos de ejecución

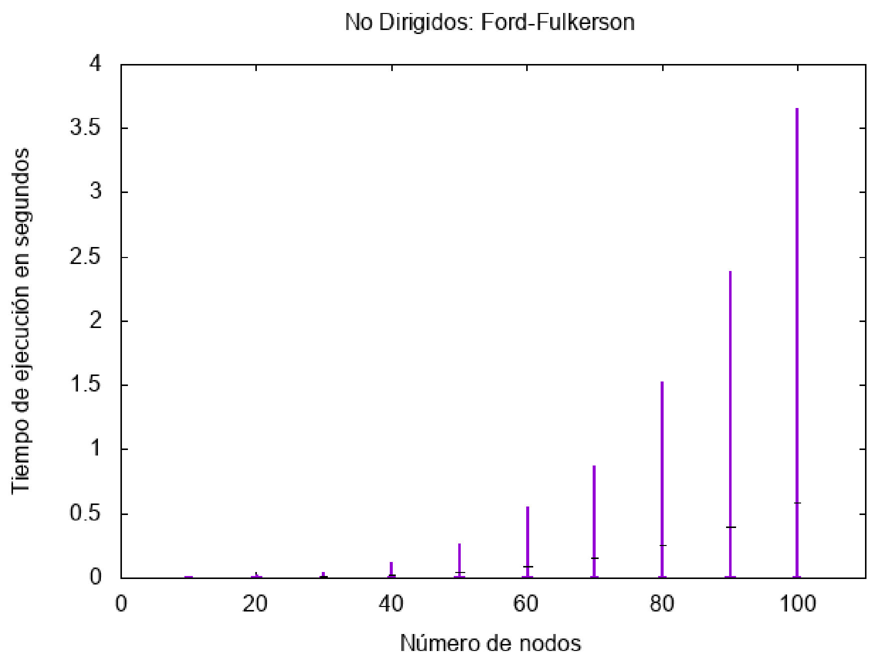

Figure 4: Diagramas de cajas y bigotes para el algoritmo de Ford-Fulkerson y función exponencial ajustada a los tiempos máximos de ejecución

4 Conclusiones

Es evidente en las cuatro gráficas anteriores que los resultados para los dos algoritmos y cada tipo de gráfico crece de manera exponencial de acuerdo al número de nodos (y por consecuencia de aristas) que tienen las instancias generadas. En cuanto a la diferencia entre los algoritmos, se puede ver que para ambos tipos de grafos los tiempos de ejecución crecen más rápido para el algoritmo de Ford-Fulkerson. Esta observaión concuerda con la teoría. Por otro lado, en cuanto a los tipos de grafos los tiempos de ejecución son menores para los grafos no dirigidos en ambos algoritmos. La diferencia es más notoria en el algoritmo de Ford-Fulkerson y puede justificarse por el hecho de que hay un menor número de aristas en los grafos no dirigidos por la manera en la que fueron generados.

References

[1] Robert W Floyd. Algorithm 97: shortest path. *Communications of the ACM*, 5(6):345, 1962.

[2] Stephen Warshall. A theorem on boolean matrices. *Journal of the ACM (JACM)*, 9(1):11–12, 1962.

[3] Ravindra K Ahuja, Thomas L Magnanti, and James B Orlin. *Network flows.* Elsevier, 2014.

[4] Satu Elisa Schaeffer. Matemáticas discretascurso en línea.