

# Tarea 5: Experimentos con el algoritmo de Ford-Fulkerson

María Gabriela Sandoval Esquivel

Mayo 2018

## 1 Introducción

La función del algoritmo de Ford-Fulkerson que se desarrolló en tareas anteriores da como resultado el flujo máximo que existe entre un par de nodos dado y el tiempo de ejecución del algoritmo. Este algoritmo encuentra todos los caminos que existen entre el nodo de inicio y el final y calcula el flujo máximo que se puede enviar a través de ellos. En esta tarea se hicieron pruebas de este algoritmo con un tipo especial de grafos para evaluar su desempeño conforme se eliminan nodos y aristas del grafo. La idea es analizar cómo se afecta el flujo máximo y el tiempo de ejecución del algoritmo conforme se quitan nodos y aristas del grafo hasta que el nodo inicial y el final quedan desconectados.

El tipo de grafos que se usaron para esta tarea pretenden asemejar una red social donde existen varias conexiones locales y algunas pocas conexiones a larga distancia. Para simular esta relación en el espacio se partió de un arreglo de nodos en forma de matriz cuadrada. Se definió como  $k$  el número de nodos por fila y de este modo el número total de nodos es  $k^2$ . Se definió otro parámetro  $l$  como el número de conexiones locales de cada nodo del grafo. De este modo se conectaron todo par de nodos que tuvieran una distancia Manhattan [1] entre ellos menor o igual al parámetro  $l$ . Además, a estas conexiones locales se les asignaron capacidades de flujo aleatoriamente conforme a una distribución normal. A los pares de nodos con distancia Manhattan mayor a  $l$  se les asignó una conexión de acuerdo con una probabilidad muy pequeña  $p$  para que hubiera pocas conexiones de larga distancia. A estas conexiones se les asignó una capacidad de flujo de acuerdo con una distribución exponencial.

Una vez que se construyeron los grafos se hicieron las pruebas con el algoritmo de Ford-Fulkerson teniendo como nodo inicial el de la esquina superior izquierda y nodo final el de la esquina inferior derecha. El hecho de que existan conexiones locales entre los nodos asegura que exista al menos un camino entre el nodo final y el inicial.

Las pruebas que se realizaron son de dos tipos: una donde se quitaron al azar aristas del grafo y otra donde se quitaron los nodos. La variabilidad de las pruebas depende de los parámetros  $k$ ,  $l$  y  $p$  con los que se generaron varios tipos de grafos. Para cada tipo de grafo se generaron algunas muestras con las que

se analizó el desempeño del algoritmo Ford-Fulkerson conforme se percolaba el grafo hasta que el nodo inicial y final quedaran desconectados.

El motivo de estas pruebas es analizar qué tan cruciales son ciertos nodos para mantener el flujo en el grafo y cómo se comporta el flujo y los tiempos de ejecución del algoritmo de Ford-Fulkerson conforme se reduce el grafo. Otro resultado de estas pruebas es que una vez que se desconectan los nodos inicial y final se obtiene un conjunto de corte (ya sea de nodos o de aristas) con los elementos del grafo que se han quitado. La cardinalidad de este conjunto es una cota superior de la cardinalidad del mínimo conjunto de corte para los nodos inicial y final [2].

## 2 Programación de funciones

Para esta tarea primero se creó la función que generaba el tipo de grafos que se describió en la introducción. Esta función recibía como datos de entrada los parámetros  $k$ ,  $l$  y  $p$  y asignaba las conexiones y las capacidades de flujo como se describió antes.

Después se crearon funciones para quitar aristas y nodos del grafo. Al quitar una arista, no solo se eliminaba de la lista de aristas del grafo, también era necesario quitar de la lista de vecinos del nodo de salida el nodo de llegada (y viceversa si se trataba de una arista no-dirigida). Al quitar un nodo, se eliminaba de la lista de nodos del grafo y para todos los vecinos de tal nodo se eliminaban sus aristas con la función de quitar arista. Se quitaban también las aristas dirigidas que llegaban al nodo por eliminar.

## 3 Pruebas experimentales

A continuación, se presentan gráficas que muestran el comportamiento del algoritmo de Ford Fulkerson conforme se removían aristas y nodos para los diferentes tipos de grafos. Para cada prueba se midió la evolución del flujo máximo del grafo y el tiempo que llevó la ejecución del algoritmo en cada iteración. Las graficas muestran las diferentes muestras que se consideraron para cada tipo de grafo con colores distintos. Para el flujo máximo se consideraron medidas absolutas con respecto al flujo obtenido inicialmente. En el eje horizontal se miden los nodos o aristas que se quitaron en cada iteración del algoritmo, en algunos casos se quitaron dos o cinco objetos de una vez para agilizar el análisis.

Las figuras 1 a 16 muestran los resultados sobre la eliminación de nodos para distintos tipos de grafos y el resto de las figuras muestran resultados para eliminación de aristas.

Las figuras 1 y 2 muestran los resultados para los grafos con  $10^2$  nodos, conexiones locales de 1 y probabilidad de conexiones a larga distancia de  $10^{-3}$ .

Estas gráficas muestran los resultados sobre cinco grafos de este tipo. El flujo máximo permanece estable con algunos cambios abruptos y el tiempo de ejecución es consistentemente decreciente. El número máximo de nodos que se quitaron para tener un flujo de cero es 180.

El efecto de la probabilidad de conexiones a larga distancia se aprecia en las diferencias de estas gráficas con las que se encuentran en las figuras 3 y 4 donde esta probabilidad es de  $10^{-3}$ . Ahí se aprecia un caso muy estable en cuanto al flujo máximo y el máximo número de nodos que se quitan para alcanzar un flujo máximo de cero es cercano a 35.

## 4 Conclusiones

Las gráficas que se obtuvieron de las pruebas muestran que el tiempo de ejecución del algoritmo de Ford Fulkerson siempre fue decreciente conforme se eliminaban objetos del grafo. En cuanto al flujo máximo es común encontrar gráficas escalonadas que indican que el flujo se mantiene estable hasta que se quitan ciertas aristas cruciales que muy probablemente pertenecen a un conjunto de corte de los nodos inicial y final del grafo.

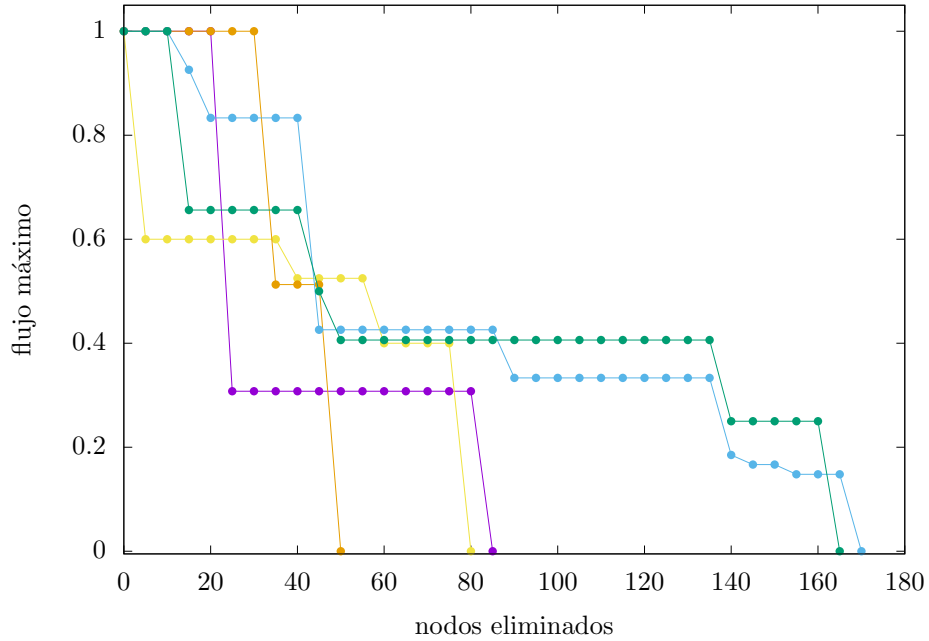


Figura 1: Flujo máximo quitando nodos para grafo con  $k = 10$ ,  $l = 1$  y  $p = 10^{-3}$

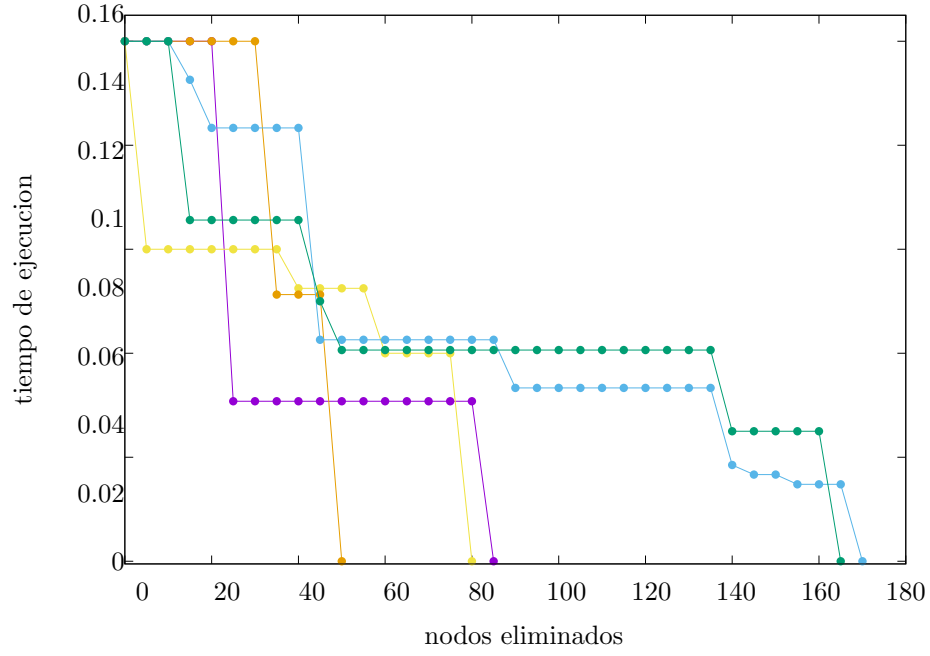


Figura 2: Tiempo de ejecución quitando nodos para grafo con  $k = 10$ ,  $l = 1$  y  $p = 10^{-3}$

## References

- [1] Eric W Weisstein et al. Taxicab metric. *From MathWorld—A Wolfram Web Resource*, 2010.
- [2] Xiao Chen, Zhe-Ming Lu, Hong-Wa Yang, and Gao-Feng Pan. A new approach to constructing a minimum edge cut set based on maximum flow.

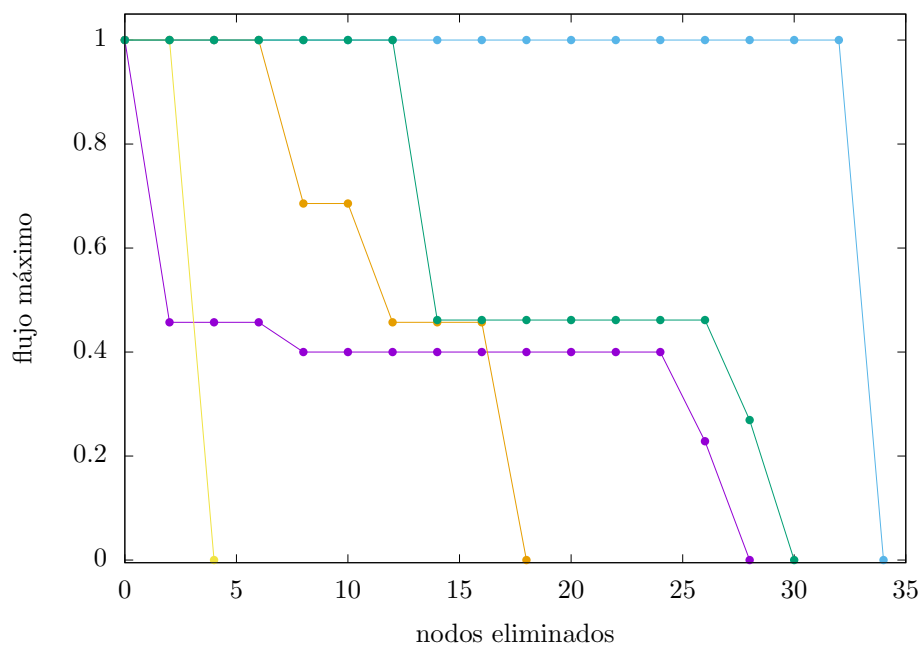


Figura 3: Flujo máximo quitando nodos para grafo con  $k = 10$ ,  $l = 1$  y  $p = 10^{-4}$

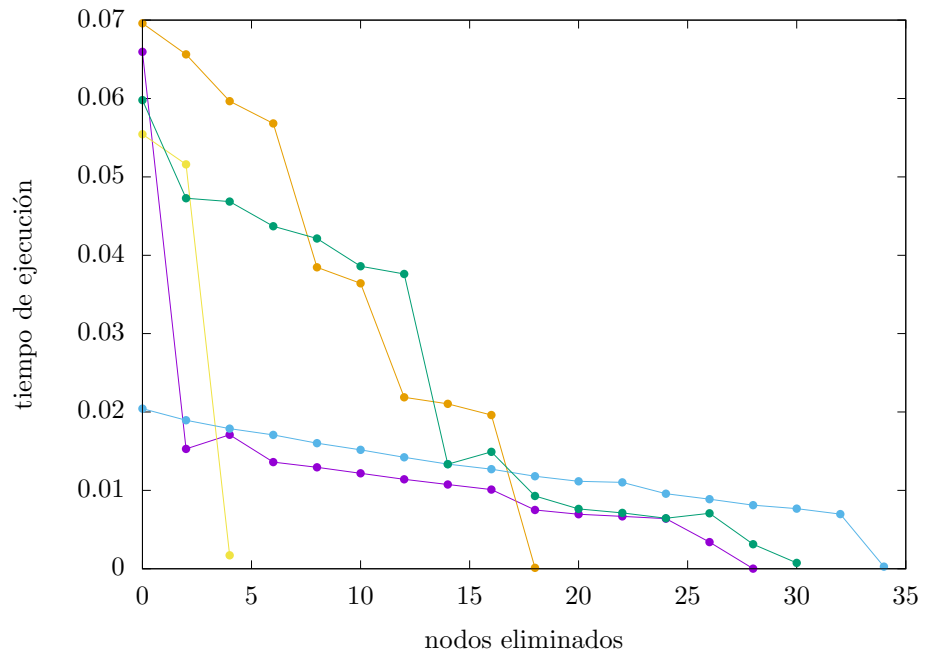


Figura 4: Tiempo de ejecución quitando nodos para grafo con  $k = 10$ ,  $l = 1$  y  $p = 10^{-4}$

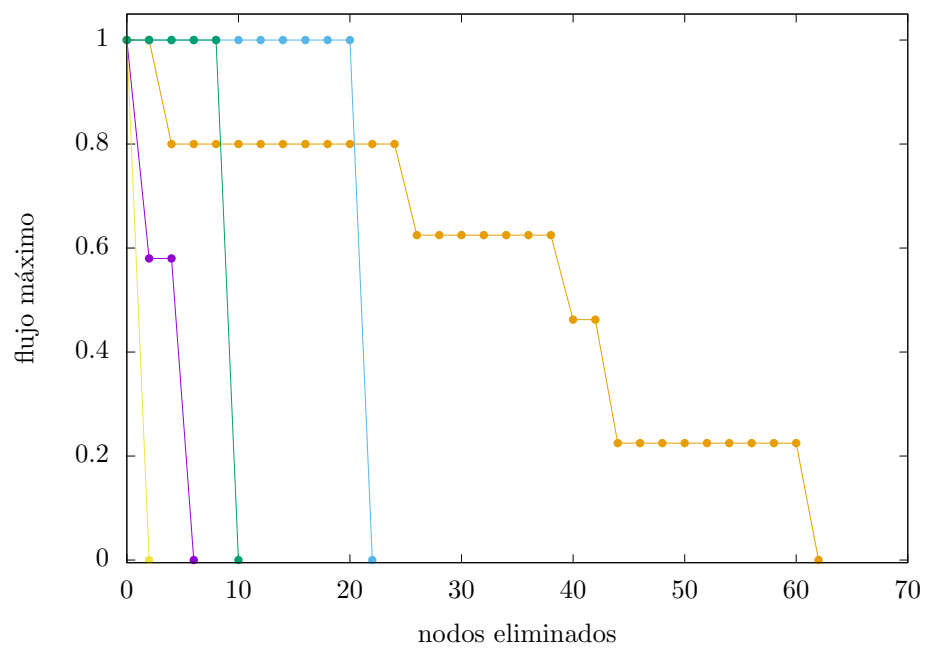


Figura 5: Flujo máximo quitando nodos para grafo con  $k = 10$ ,  $l = 2$  y  $p = 10^{-4}$

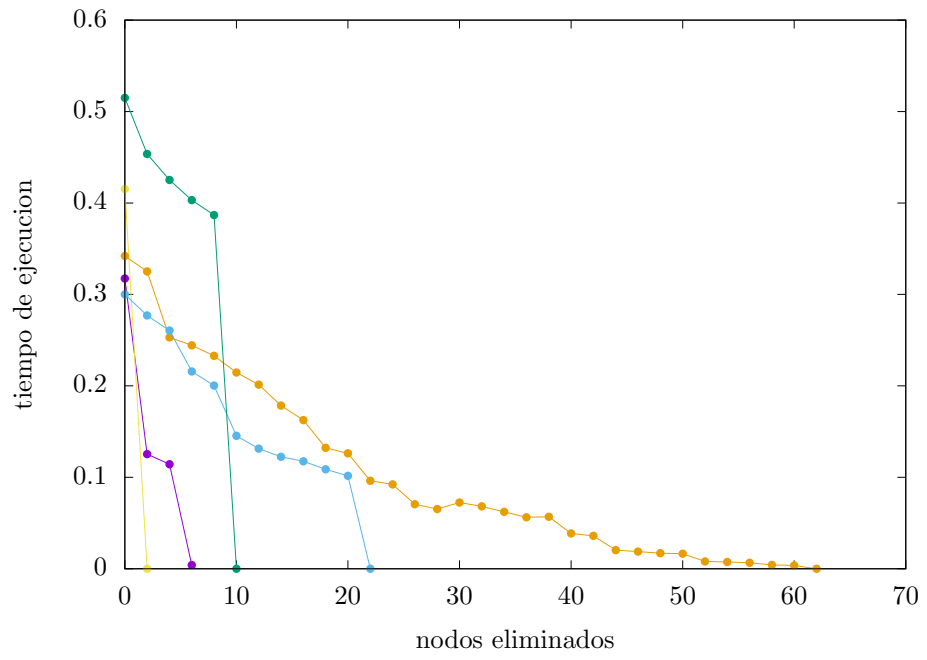


Figura 6: Tiempo de ejecución quitando nodos para grafo con  $k = 10$ ,  $l = 2$  y  $p = 10^{-4}$



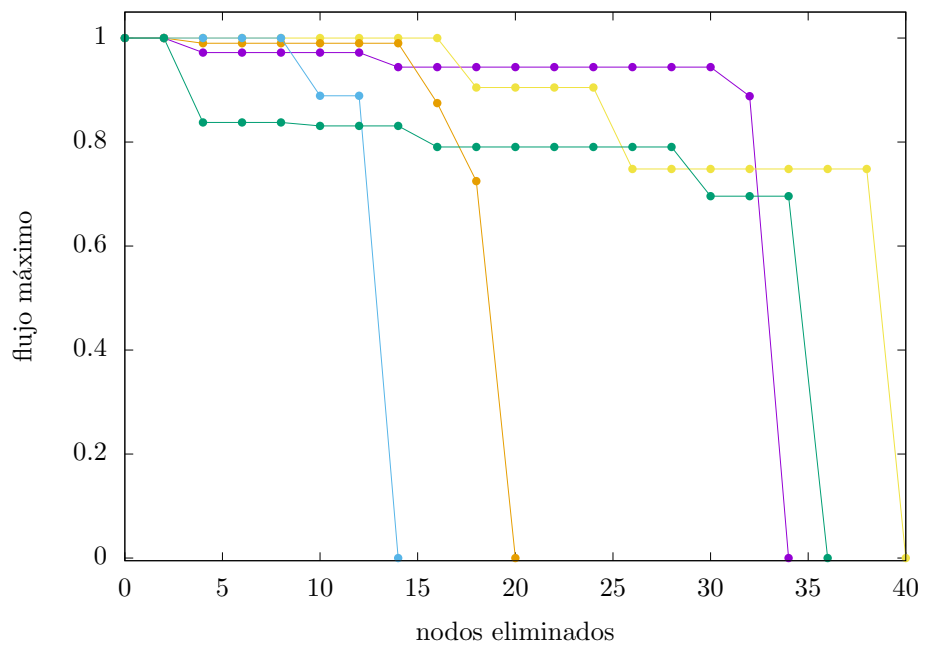


Figura 7: Flujo máximo quitando nodos para grafo con  $k = 10$ ,  $l = 3$  y  $p = 10^{-4}$

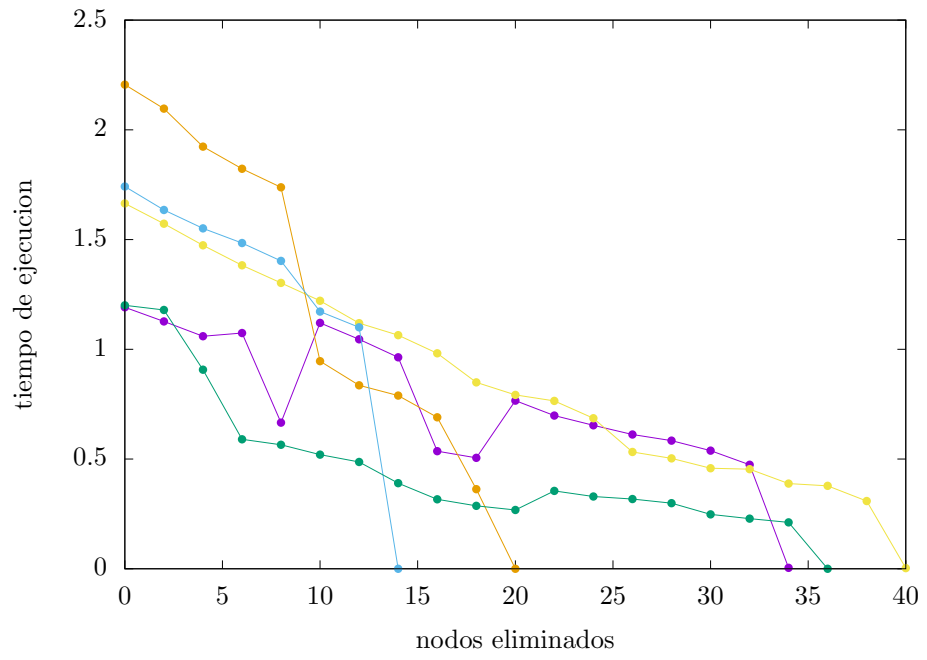


Figura 8: Tiempo de ejecución quitando nodos para grafo con  $k = 10$ ,  $l = 3$  y  $p = 10^{-4}$

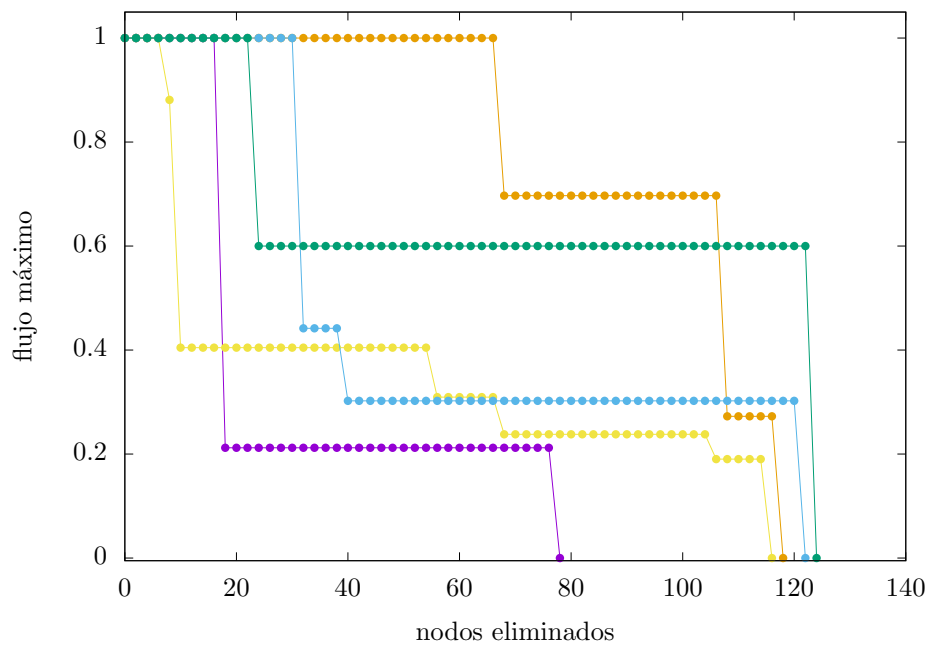


Figura 9: Flujo máximo quitando nodos para grafo con  $k = 20$ ,  $l = 1$  y  $p = 10^{-4}$

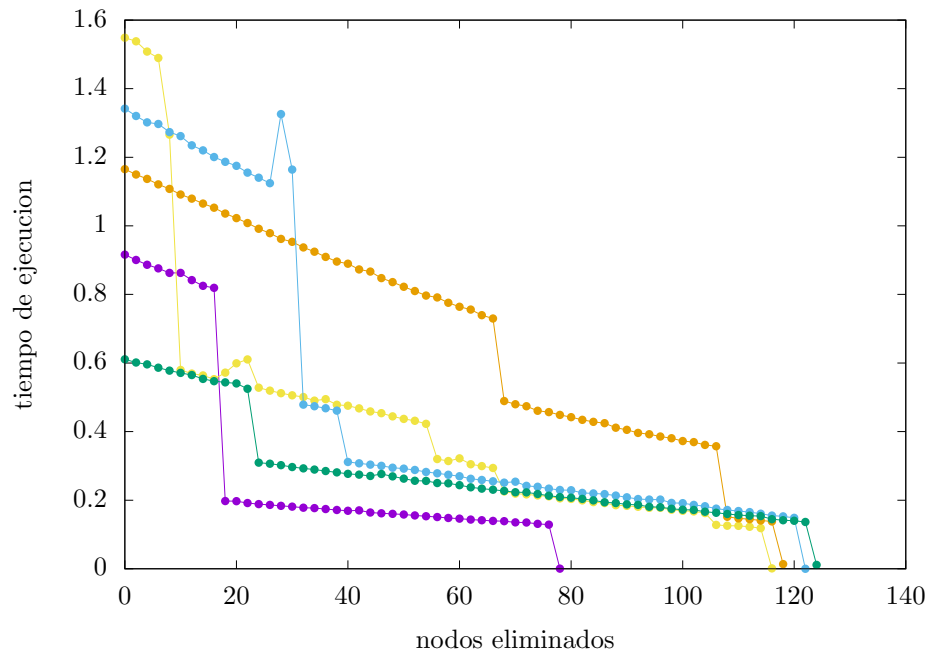


Figura 10: Tiempo de ejecución quitando nodos para grafo con  $k = 20$ ,  $l = 1$  y  $p = 10^{-4}$

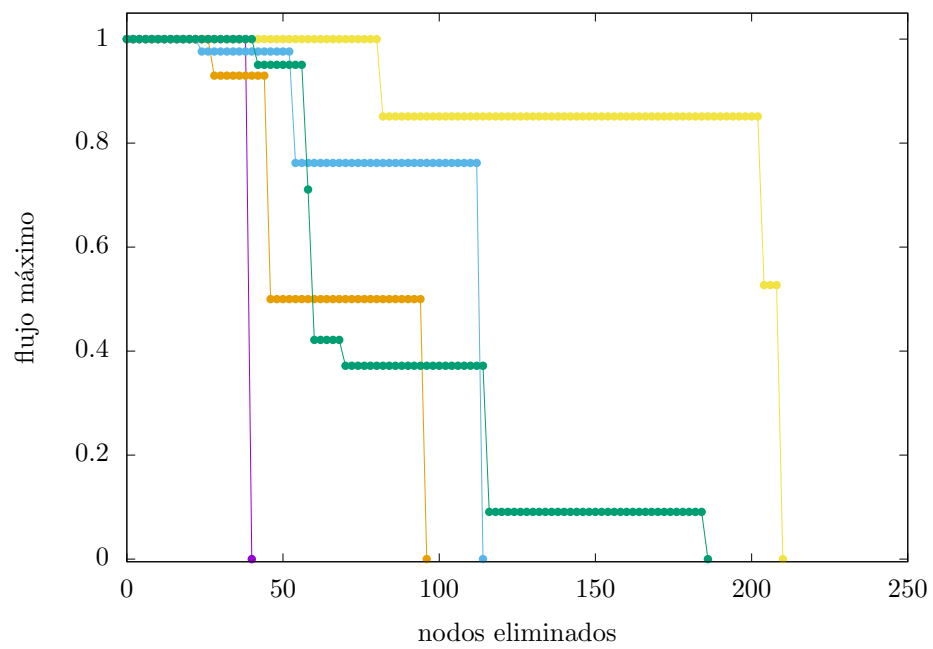


Figura 11: Flujo máximo quitando nodos para grafo con  $k = 20$ ,  $l = 2$  y  $p = 10^{-4}$

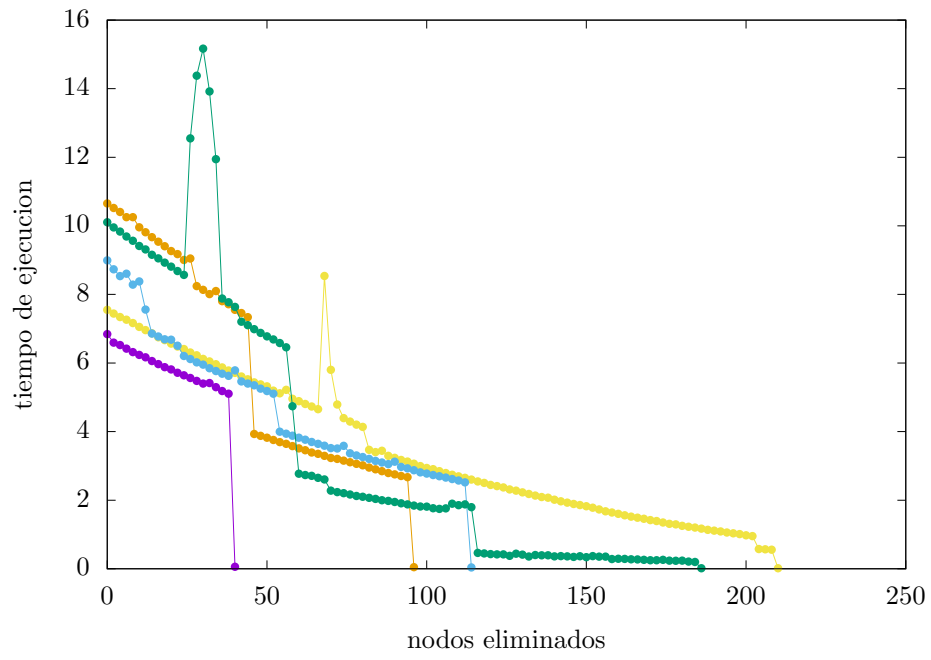


Figura 12: Tiempo de ejecución quitando nodos para grafo con  $k = 20$ ,  $l = 2$  y  $p = 10^{-4}$

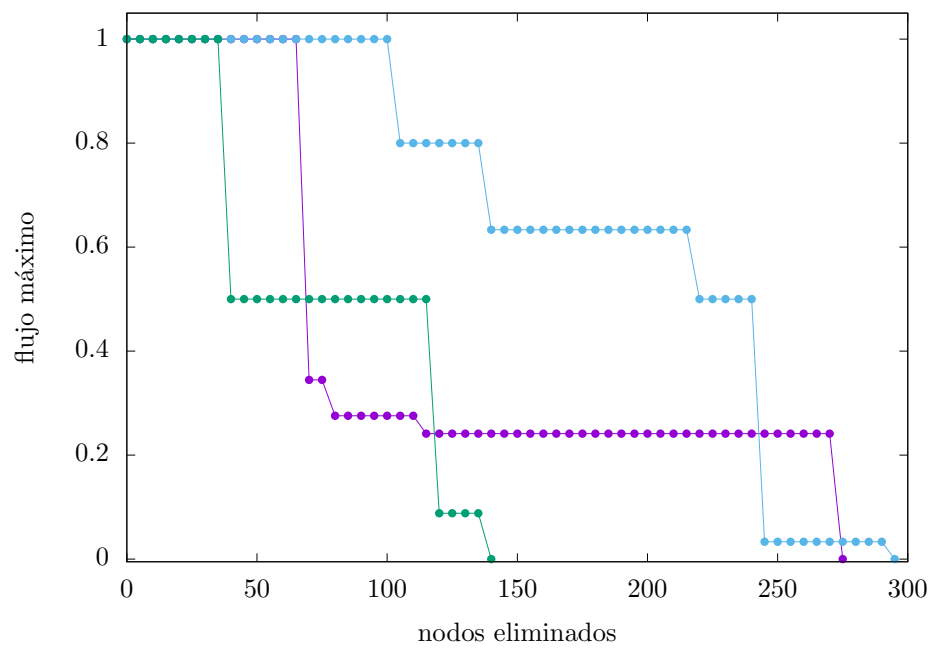


Figura 13: Flujo máximo quitando nodos para grafo con  $k = 30$ ,  $l = 1$  y  $p = 10^{-4}$

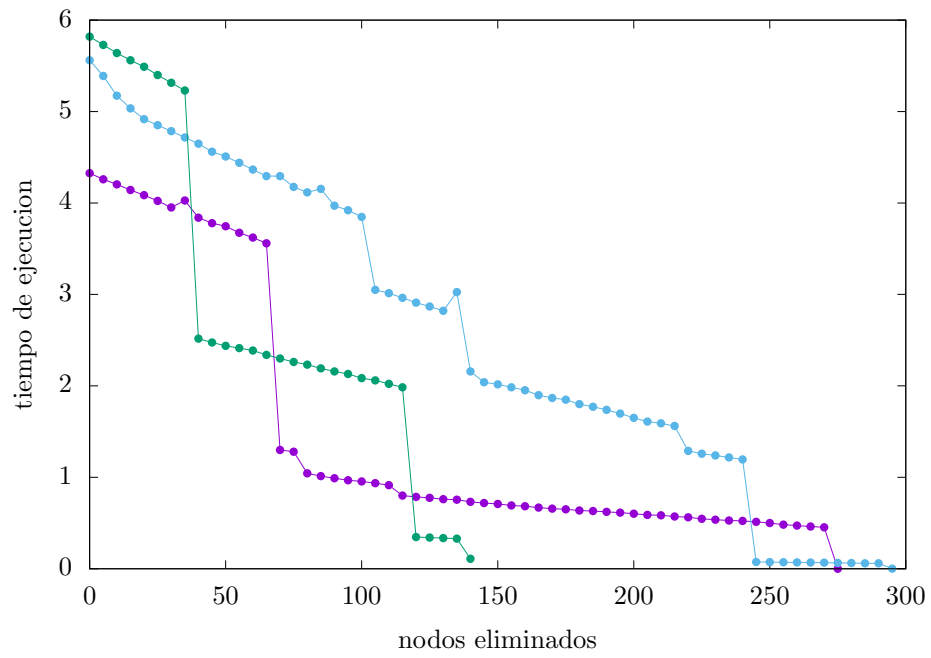


Figura 14: Tiempo de ejecución quitando nodos para grafo con  $k = 30$ ,  $l = 1$  y  $p = 10^{-4}$



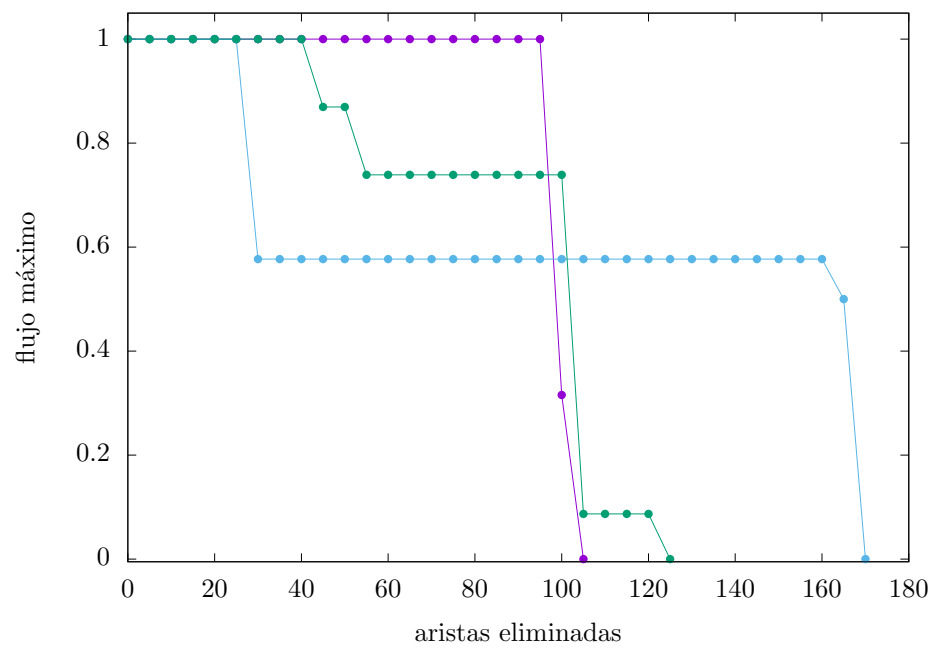


Figura 15: Flujo máximo quitando aristas para grafo con  $k = 10$ ,  $l = 1$  y  $p = 10^{-4}$

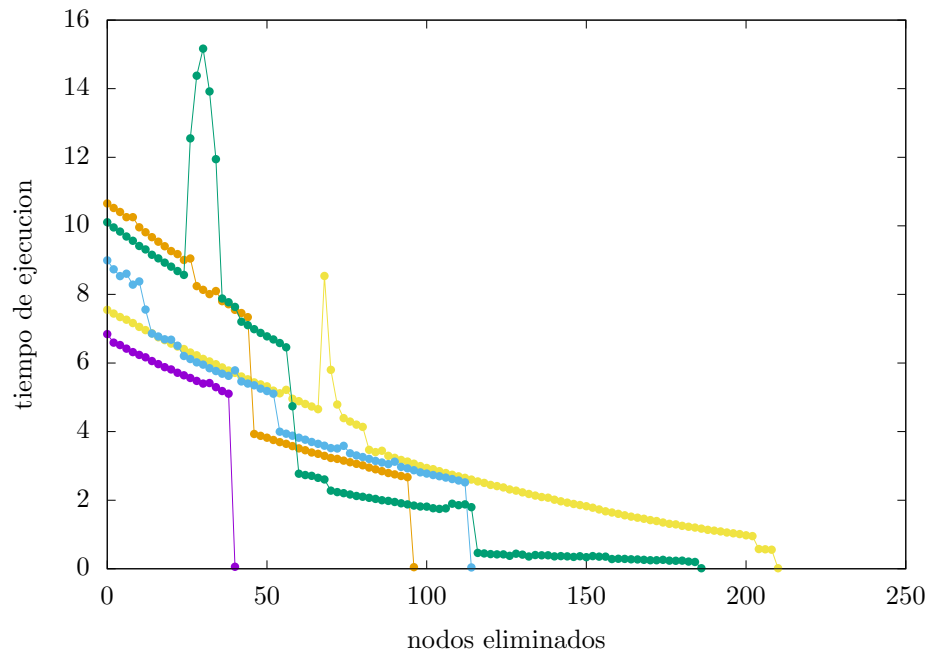


Figura 16: Tiempo de ejecución quitando aristas para grafo con  $k = 10$ ,  $l = 1$  y  $p = 10^{-4}$

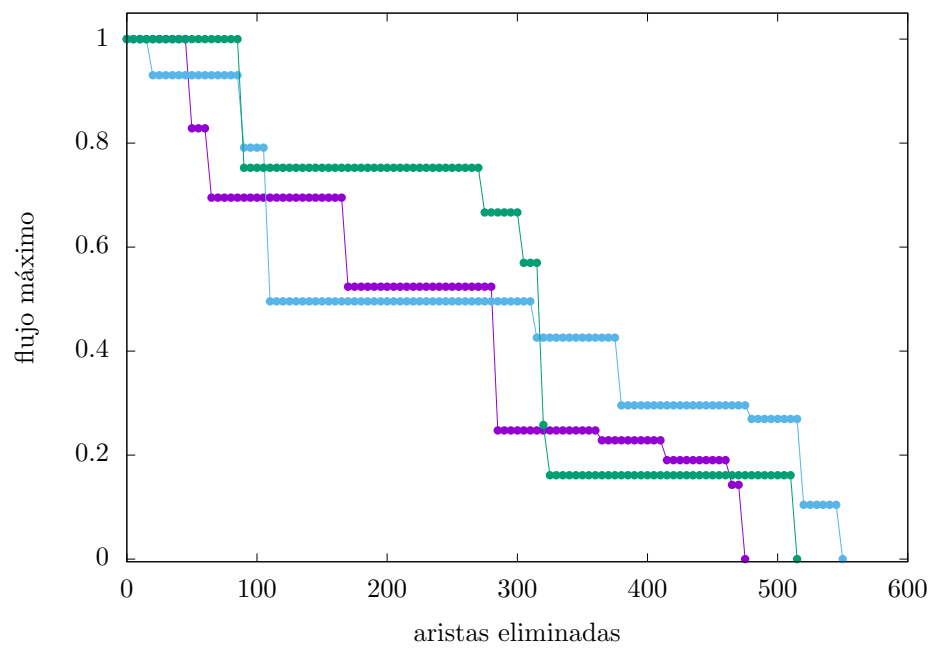


Figura 17: Flujo máximo quitando aristas para grafo con  $k = 10$ ,  $l = 2$  y  $p = 10^{-4}$

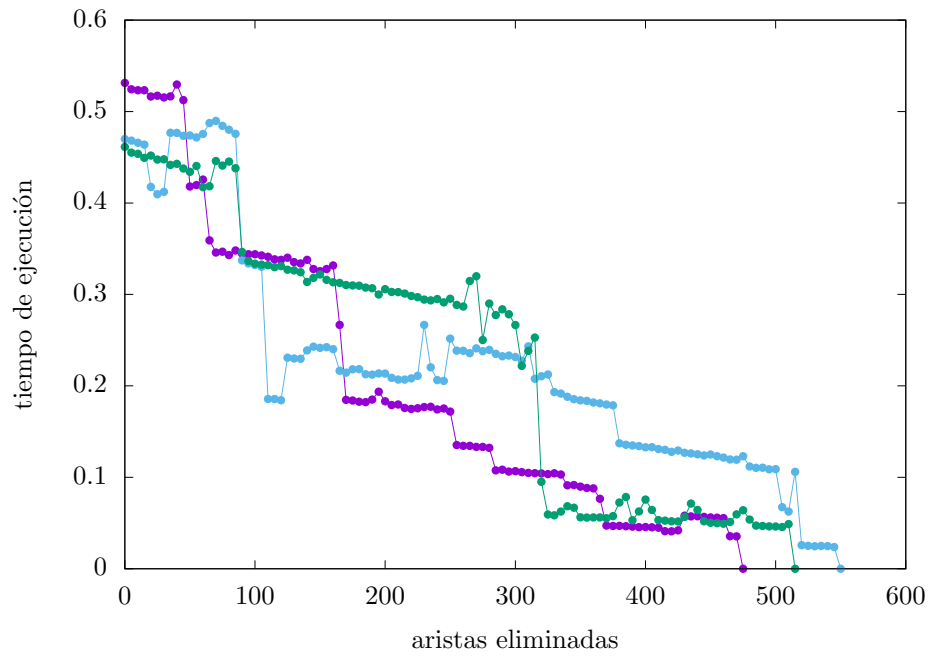


Figura 18: Tiempo de ejecución quitando aristas para grafo con  $k = 10$ ,  $l = 2$  y  $p = 10^{-4}$

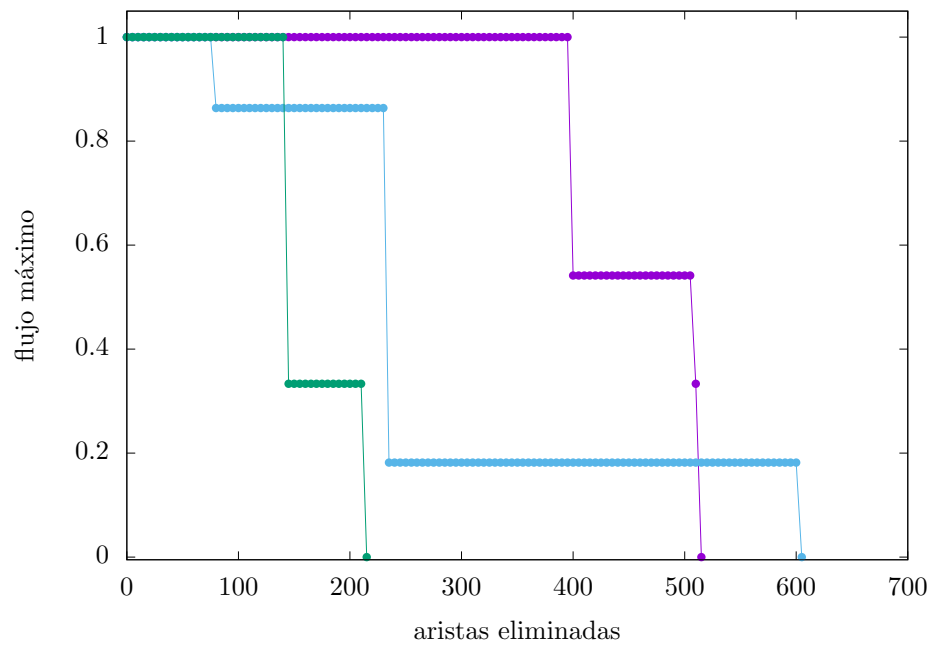


Figura 19: Flujo máximo quitando aristas para grafo con  $k = 20$ ,  $l = 1$  y  $p = 10^{-4}$

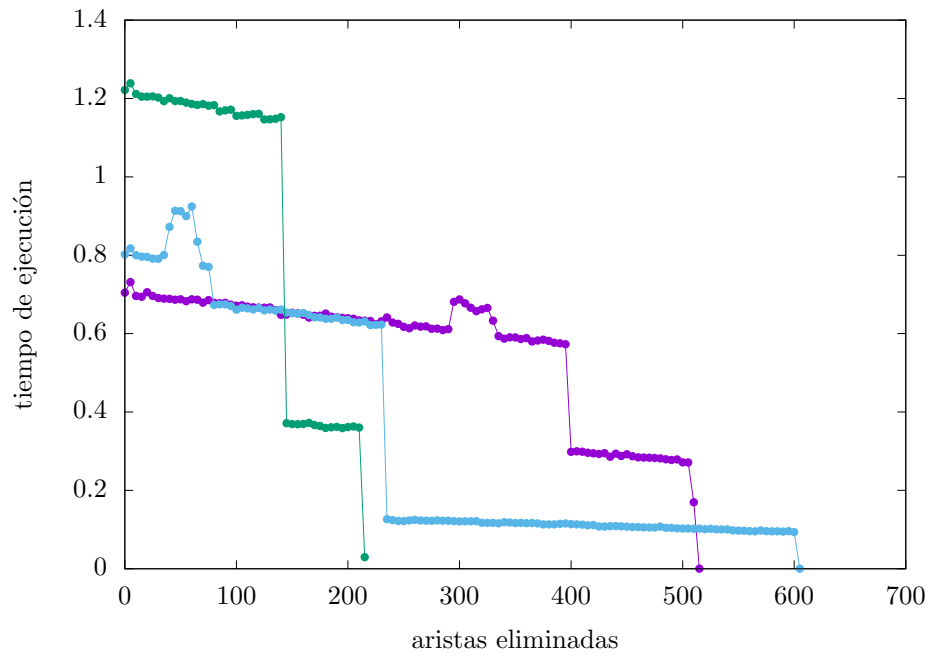


Figura 20: Tiempo de ejecución quitando aristas para grafo con  $k = 20$ ,  $l = 1$  y  $p = 10^{-4}$

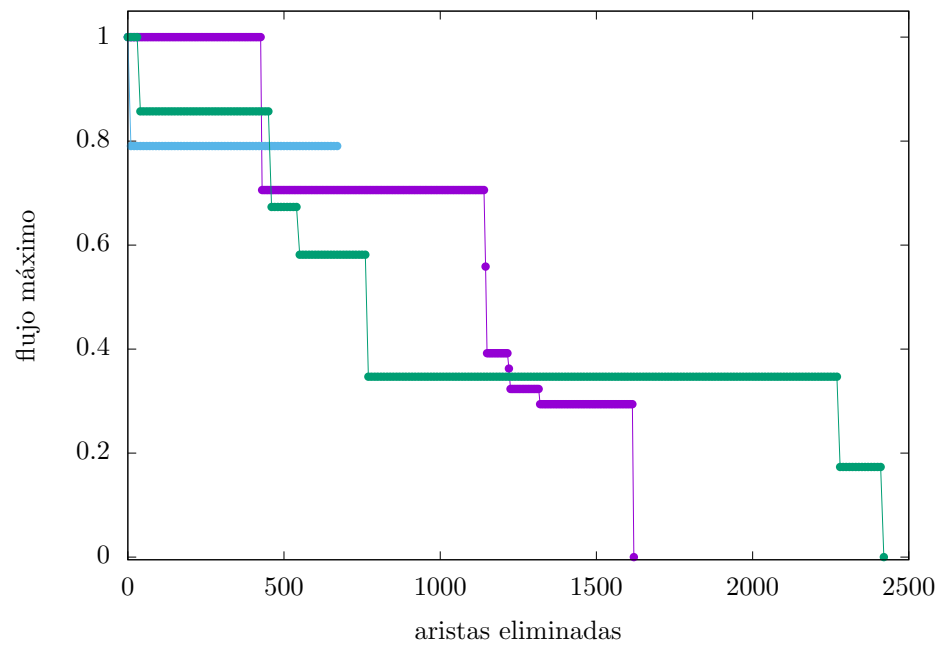


Figura 21: Flujo máximo quitando aristas para grafo con  $k = 20$ ,  $l = 2$  y  $p = 10^{-4}$

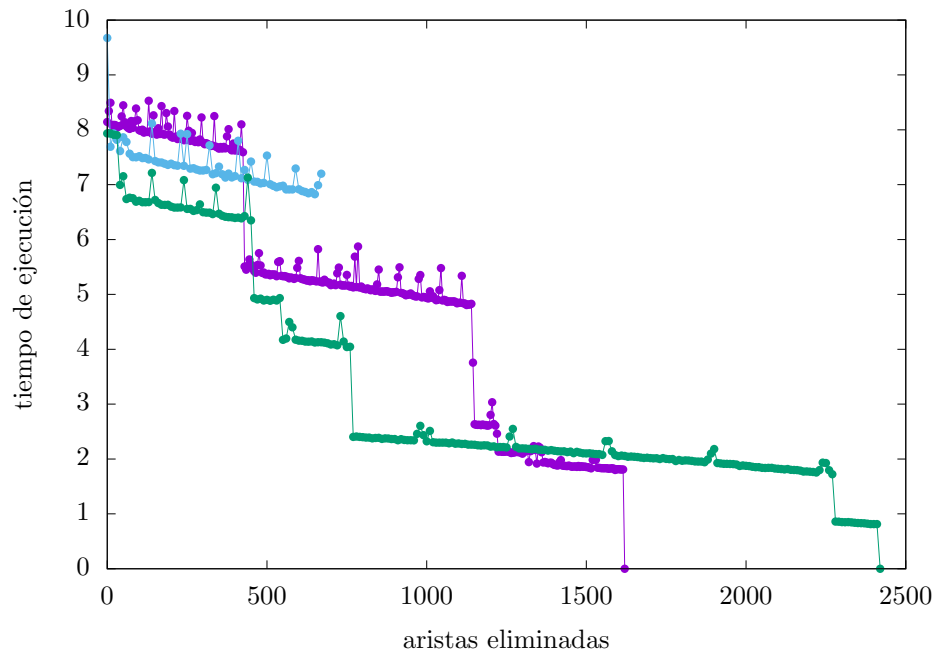


Figura 22: Tiempo de ejecución quitando aristas para grafo con  $k = 20$ ,  $l = 2$  y  $p = 10^{-4}$



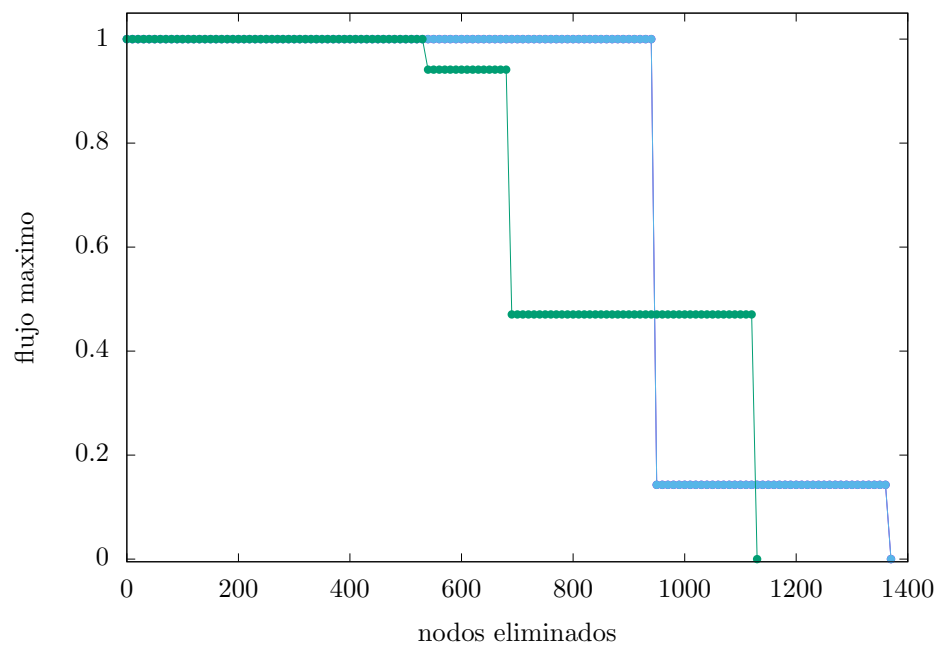


Figura 23: Flujo máximo quitando aristas para grafo con  $k = 30$ ,  $l = 1$  y  $p = 10^{-4}$

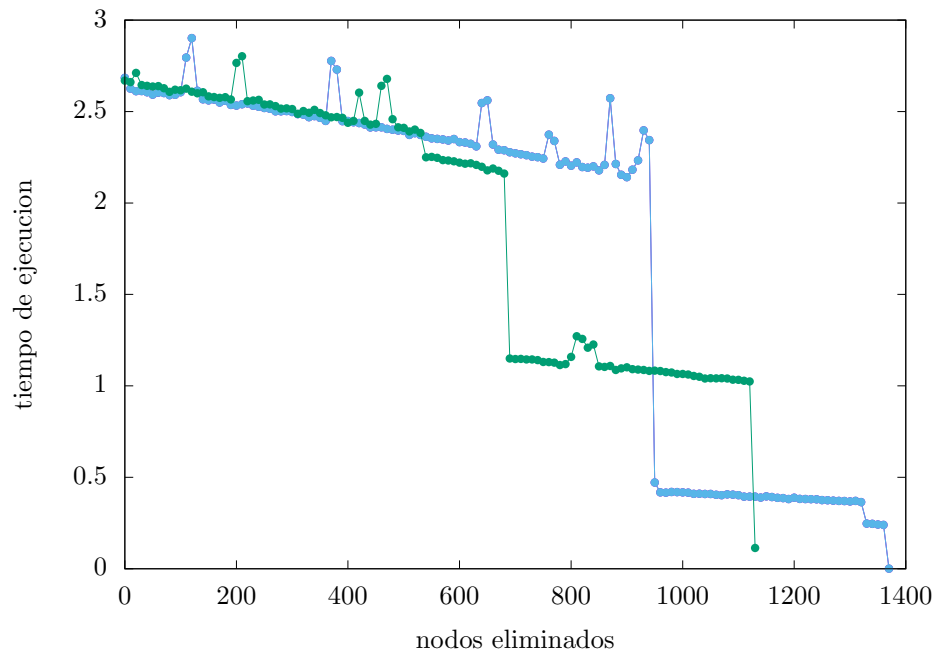


Figura 24: Tiempo de ejecución quitando aristas para grafo con  $k = 30$ ,  $l = 1$  y  $p = 10^{-4}$