Trabajo Práctico Especial

Programación 3

TUDAI

Ciencias Exactas

Unicen

02/07/2023

Agustin Montes

montesagustin99@gmail.com

Backtracking

Estrategia

La estrategia que llevó a cabo para resolver el problema fue explorar todas las combinaciones posibles de túneles con la intención de construir una red subterránea con un costo mínimo de metros de túneles.
Cada vez que se accede a backtracking verificar si la red se encuentra completa, en caso de ser así verificar si la red actual era menor en cantidad de metros que la mejor red hasta el momento.
Si la red todavía no está completa, se continúa recorriendo los túneles posibles, por cada túnel lo agrego a la red actual y verificar si es posible podar. La poda se realiza si la cantidad de metros de la red actual es mayor a la cantidad de metros de la mejor red hasta el momento.
En caso de no ser posible podar, continuó la exploración de los posibles túneles de forma recursiva hasta encontrar la red que conlleve a la menor cantidad de metros de túnel.
En el momento que vuelve de la recursión se elimina el túnel seleccionado en ese momento, de forma que se pueda explorar todas las soluciones posibles.

Costo operacional

El costo operacional de explorar todas las soluciones posibles, en el peor de los casos, va a ser o(n!). Esto se debe a que, en el peor de los casos, va a tener que recorrer todas las combinaciones posibles que se generen con los n tuneles.

Greedy

Estrategia

La estrategia greedy que se llevó a cabo fue, ir recorriendo las estaciones y por cada estación obtener los túneles que parten desde la misma. De este conjunto de túneles se selecciona el túnel de menor cantidad de metros y que a su vez te lleve a una estación que no haya sido visitada. De esta forma se obtendrá una red conectada, de donde se pueda llegar a una estación de forma directa o indirecta.
Con este algoritmo estaremos optando por seleccionar la mejor decisión localmente, pero no nos asegura que vayamos a encontrar la mejor solución global, por la red final puede no ser la mejor solución.

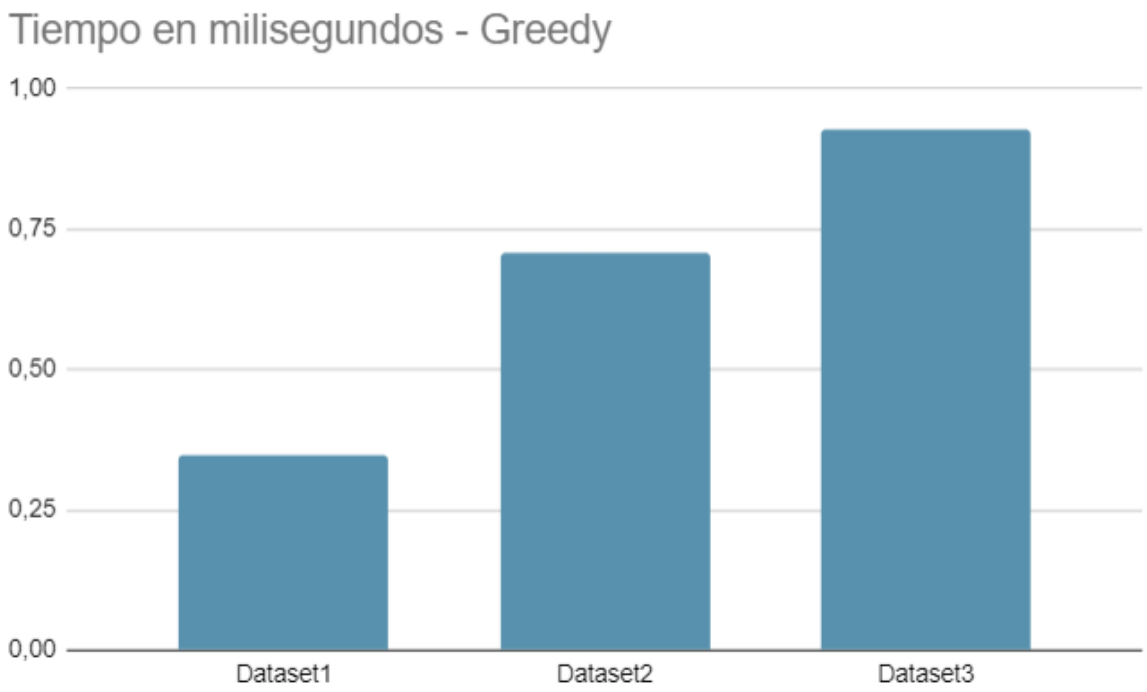
Costo operacional

El costo operacional de este algoritmo es o(n^2), donde n es la cantidad de estaciones. Esto se debe a que se recorre todas las estaciones (n) y por cada estación se recorre por todos los túneles de la misma.
Esto nos llevará a que en el peor de los casos tengamos que recorrer todos los túneles de cada estación.

Resultados obtenidos

Tiempo en milisegundos

El siguiente gráfico refleja el tiempo en milisegundos al ejecutar el algoritmo greedy con los distintos datasets.



Se puede ver como el tiempo aumenta de manera progresiva en relación con la cantidad de estaciones que contiene cada dataset.

A continuación se muestra el gráfico que plasma los resultados obtenidos mediante backtracking midiendo los diferentes datasets en milisegundos.
Se puede notar el claro impacto en el aumento de tiempo para los diferentes datasets. Esto se debe a que la cantidad de combinaciones posibles es n!. A medida que n crece, las combinaciones también lo hacen con un gran impacto.

Se puede ver como el dataset 3 sobrepasa los límites del gráfico, fue realizado de esta forma para que se pueda notar el tiempo de los demás datasets. De otra manera habria que cambiar las escalas y los tiempos del dataset 1 y 2 no se podrían apreciar.

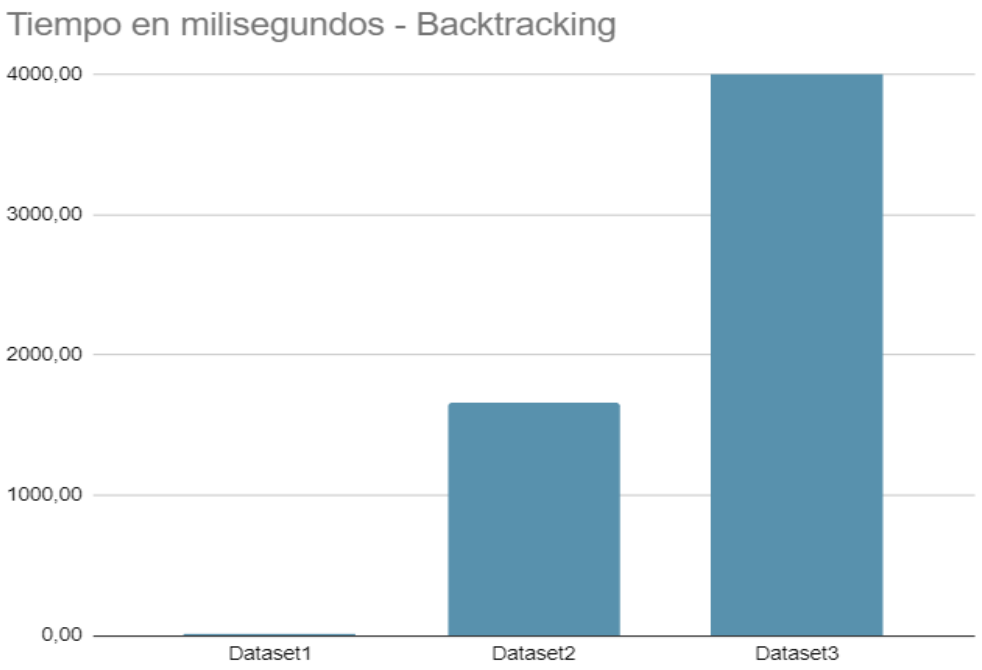
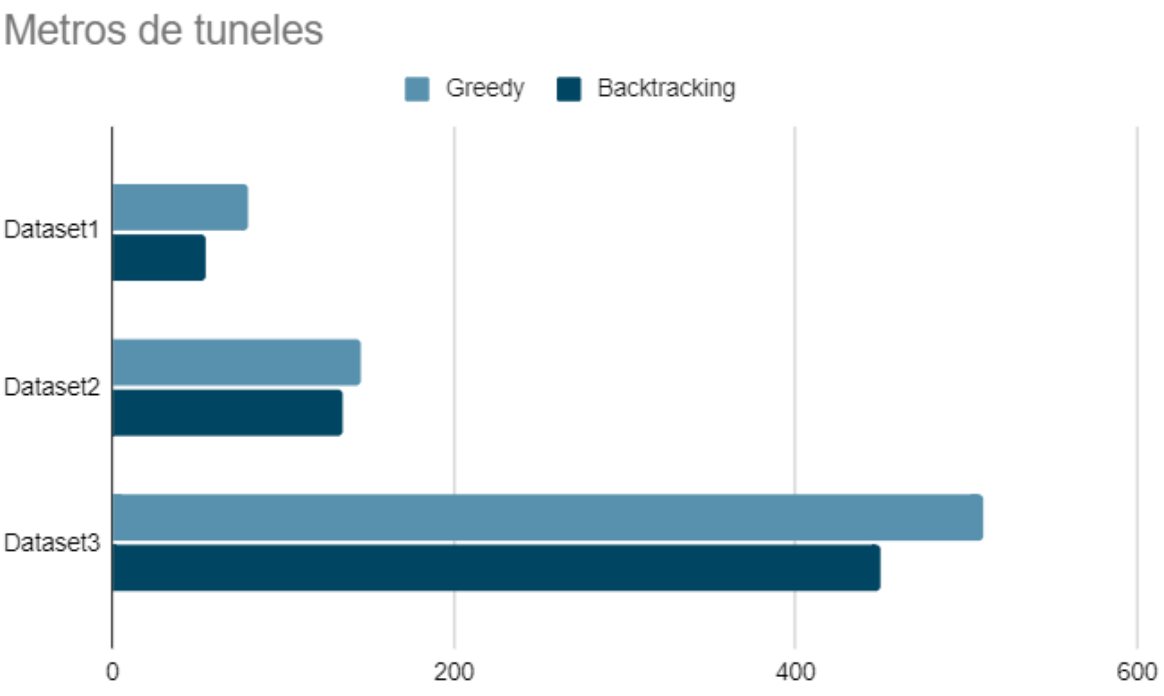


Tabla comparativa

A continuación se muestra una tabla comparando los algoritmos de greedy y backtracking midiéndolos por cantidad de metros de túneles.



Se puede observar que para los diferentes datasets, backtracking siempre obtiene la mejor solución posible ya que se encarga de recorrerlas todas. Mientras que en el caso de greedy obtiene solamente un aproximado.

Conclusión

La conclusión que podemos sacar de este trabajo es que el algoritmo de backtracking para realizar una búsqueda exhaustiva sobre todas las soluciones posibles, pero que no en todos los casos va a resultar del todo útil ya que esto tiene un costo muy alto para casos donde el espacio de búsqueda es grande.
Mientras que Greedy nos puede servir para obtener, en algunos casos, la mejor solución o una solución exacta. Este algoritmo tiene un costo menor pero no nos asegura siempre la mejor solución.

Por otra parte, podemos combinar ambas técnicas. Se puede utilizar greedy para obtener una solución aproximada y usarla como punta de partida para realizar un algoritmo de backtracking y poder tener una poda de manera inicial.