Informe TPE programación III

Agustin Carretto
Lucas Quiroga

Introducción

A continuación se presentaran la estrategia, costo computacional y comparación entre estrategias de los distintos métodos empleados para calcular el camino con la distancia mínima para cumplir con la siguiente consigna:

“Las autoridades de una ciudad deciden construir una red de subterráneos para resolver los constantes problemas de tráfico. La ciudad ya cuenta con N estaciones construidas, pero todavía no tienen ningún túnel que conecte ningún par de estaciones entre sí. La red de subterráneos que se construya debe incluir a todas las estaciones (es decir, que de cualquier estación H pueda llegar a cualquier otra estación J, ya sea de manera directa o atravesando otras estaciones). Sin embargo, debido al acotado presupuesto, las autoridades desean construir la menor cantidad de metros de túnel posibles. Para esto han calculado cuantos metros de túnel serían necesarios para conectar de manera directa cada par de estaciones existentes. ”

Backtracking

¿Cuál fue la estrategia Backtracking llevada adelante para resolver el problema? ¿Cuál es el costo computacional de dicha estrategia?

La estrategia de Backtracking utilizada fue la siguiente:

- Obtengo todos los Nodos del grafo
- Para cada uno de estos nodos, comienzo una búsqueda en profundidad
- Mientras ejecuto la búsqueda, armo un camino y verifico que no sea un camino inviable y por otro lado, si es un camino solución
- En caso de ser inviable corto la iteración actual para seguir buscando por otra rama
- En caso de ser solución (solo llega si es una mejor que la anterior) guardo el camino actual y corto la iteración para continuar por otro camino

Esta estrategia recorrerá todos los arcos del grafo,aprovechando distintas estructuras de almacenamiento para llevar registro de las estaciones y los arcos visitados, así como de los túneles que ya fueron construidos utilizando una clase auxiliar (IntegerPair) para asegurarse de no contabilizar distancias de túneles ya construidos y llamando a la misma función recursivamente para actualizar dichas estructuras hasta llegar a una solución.

La complejidad temporal de este algoritmo es ϵ O(N!) donde N es el número de arcos del grafo, ya que las iteraciones son sobre ellos y se visitan todos los arcos por cada conjunto generado, y cada conjunto tendrá entre 1 y N elementos.

GREEDY

¿Cuál fue la estrategia Greedy llevada adelante para resolver el problema? ¿Cuál es el costo computacional de dicha estrategia?

La estrategia greedy elegida fue la siguiente:

Se obtienen todos los arcos del grafo ordenados por su peso, el primero se agrega indiscriminadamente y luego de eso se agrega para cada estación el mejor arco siguiente que esté disponible. Alternativamente hay una porción de código que permite reducir la cantidad de iteraciones pero reduciendo también la eficacia del camino encontrado, la cual consta de un solo loop que itera los arcos conseguidos y los agrega solamente si es posible hacerlo (existe la estación origen de dicho arco).

La complejidad temporal de este algoritmo es ϵ O(N) si se elige usar el algoritmo eficiente pero poco eficaz y ϵ O(N^2) si se utiliza el otro loop.

Mostrar una tabla comparativa, para las distintas entradas posibles, de los resultados obtenidos por ambas técnicas. Por resultados no sólo nos referimos a la calidad de la solución sino también a la métrica que determina el costo de obtener dicha solución.

Algoritmo	Dataset 1	Dataset 2	Dataset 3
Backtracking	M:152 R:55	M:4503 R:135	***
Greedy O(N^2)	M:130 R:55	M:812 R:135	M:22830 R:440
Greedy O(N)	M:12 R:55	M:30 R:260	M:156 R:670

M -> Métrica

R-> Resultado

***: El algoritmo tardó 35 minutos en ejecutarse y lo detuve, no logre obtener una métrica.

Para calcular la métrica, se colocó un contador de iteraciones dentro de cada algoritmo el cual se consulta al finalizar la ejecución.

Conclusión:

En cuanto a la elección de las técnicas algorítmicas utilizadas, tanto el Backtracking como el Greedy son empleados para determinar la cantidad de metros de túnel necesarios para conectar directamente cada par de estaciones existentes en la ciudad.

El Backtracking es un enfoque exhaustivo que explora todas las posibles combinaciones de conexiones entre estaciones, lo que garantiza que se encuentre la solución óptima en términos de la cantidad mínima de metros de túnel necesarios. Sin embargo, este enfoque puede resultar costoso en términos de tiempo de ejecución, especialmente cuando el número de estaciones es grande.

Por otro lado, el enfoque Greedy se basa en tomar decisiones óptimas en cada paso del proceso de construcción de la red de subterráneos. Se propuso 2 formas de elegir la mejor opción, siendo una la mejor opción entre todas las posibles, y otra tomando la mejor opción entre todas las restantes, generando asi una diferencia entre los resultados y el tiempo de ejecución de este algoritmo. Este enfoque tiende a ser más eficiente en términos de tiempo de ejecución, ya que no explora todas las posibles combinaciones, sino que se centra en las conexiones más prometedoras en cada momento. Sin embargo, debido a su naturaleza ávida, puede no garantizar siempre la solución óptima en términos de la mínima cantidad de metros de túnel necesarios.

Al comparar los resultados obtenidos con ambas técnicas, es importante considerar diversas métricas que evalúen tanto la calidad de la solución como el costo asociado. Estas métricas incluyen la cantidad total de metros de túnel construidos y las iteraciones del algoritmo

En conclusión, la elección de la técnica algorítmica más adecuada para este problema debe basarse en un análisis cuidadoso de las necesidades y restricciones específicas ,considerando la calidad de la solución deseada y el costo asociado.

Como ejemplo de la necesidad de elegir bien el algoritmo a utilizar, al probar el algoritmo con los 3 Datasets proveídos, en el tercero el backtracking tomo tiempos muy extensos mientras que el greedy terminó rápidamente variando su resultado y eficiencia final segun que version utilice