

1. Algoritmo de Solución

En este caso se escogio un ejercicio de fuerza bruta. Cuando se mira este problema es esencialmente el problema de Clique Cover. Este es un problema muy conocido que es NPC. Por lo tanto habia que hacer una decisión entre correctitud y velocidad. Para iniciar, se hizo intento de montar un algogoritmo greedy. Este algoritmo esencialmente lo que hacia era randomizar los nodos y probar meterlos en diferentes cliques repitiéndolo un numero de veces. Este era un algoritmo aproximado sin embargo, en ningún momento conseguí que este algoritmo funcionara pues aunque daba respuestas aproximadas correctas (En algunas ocasiones incluso coincidía con el numero de cliques que habia en los resultados) pero la manera en la que lo repartia era sumamente aleatoria (probablemente por el principio de que acomodaba los nodos de manera aleatoria, quizás hubiera dado mejores resultados si los combinaba de manera mas inteligente). Por lo tanto decidi irme por correctitud y programar un decorador de modo tal que si se demoraba mas de 10 segundos lo terminara y diera una respuesta por default. De ese modo sacrifique velocidad puesto que la manera en la que se califica esto es suficientemente rígida como para que no me sintiera cómodo confiando en la suerte y prefiriendo quedarme con aquellos problemas que pudieran ser solucionados en menos de 10 segundos.

2. Explicación del Algoritmo

Este algoritmo esencialmente lo que hace es revisar absolutamente todas las posibles combinaciones de subgrafos. En caso de que una combinación concreta tenga todos los nodos revisa si esta es una solución con todos cliques. Es decir, revisa si todos los subgrafos son cliques. En caso de que sea asi revisa que haya menos que en el ultimo que se encontro. Si es asi entonces lo escoge y retorna best. Si alguna de estas ultimas dos condiciones no se cumple simplemente se queda con el anterior mejor. En caso de que el grupo de subgrafos no tenga todos los nodos prueba con todas las posibles combinaciones de subgrafos que contengan este subgrafo concreto.

Nota, es importante aclarar que los subgrafos estan representados en una lista en donde el indice n es el identificador y el valor es el subgrafo al que pertenece.

3. Análisis

3.1. Complejidad Temporal

La complejidad de este algoritmo es increíblemente alta puesto que es a fuerza bruta. Para iniciar, tiene que pasar por absolutamente todos los posibles conjuntos de subgrafos. En este caso existen $(n + 1)^n$ posibles conjuntos de subgrafos. Sin embargo, ademas, este algoritmo debe verificar que el resultado es un conjunto de cliques que cubren todo. Por lo tanto, en el peor

de los casos debe verificar para cada uno de los subgrafos esto tiene entonces una complejidad de $O((n+1)^n \cdot n^2)$ lo cual es impensablemente grande y por lo tanto quizás habría sido mejor probarlo con otro tipo de algoritmo aproximado.

3.2. Complejidad Espacial

Vamos a ignorar en este caso el espacio ocupado por la pila de recursión. Ahora tomando eso en cuenta note que se consume la matriz de adjacencia (que tiene un costo de $O(n^2)$) además de tomar el costo de la lista de cliques que es donde se guardan todos los posibles subgrafos que tiene entonces un peso de $O(n)$ por lo tanto la complejidad espacial es de $O(n)$

4. Preguntas

4.1. Pregunta 1

En este caso para el algoritmo que implemente lo único que cambiaría sería la revisión de que todos son cliques. Lo único que se necesitaría es tener un booleano extra que muestre si existen dos células que no se comuniquen y si ese es el caso y se encuentra otro par entonces se considera como una solución no válida y se continúa con el siguiente conjunto de subgrafos.

4.2. Pregunta 2

Este punto, nos daría un problema y una ventaja. Para iniciar cambia lo que consideramos un conjunto de subgrafos válido como respuesta pues si existe un subgrafo de mayor tamaño que T entonces este sería inválido. La ventaja que tiene es que nos permitiría cortar cantidades inmensas de posibles conjuntos de subgrafos pues si ya existe uno cuyo grado supere T entonces ninguno de los que sigan podrán ser válidos.