Dominating Sets - Aquiles Fernández Gambero

Ejercicio 7.32:

A subset of the nodes of a graph G is a dominating set if every other node of G is adjacent to some node in the subset. Let:

$DOMINATING-SET = \{< G, k > \mid \text{G has a dominating set with k nodes}\}$

Show that it is NP-complete by giving a reduction from VERTEX-COVER

Definición de Dominating-Set

Contando con la definición de Dominating-Set, definida formalmente como:

$\text{DOMINATING-SET} = \{< G, k > \mid \text{G has a dominating set with k nodes}\}$

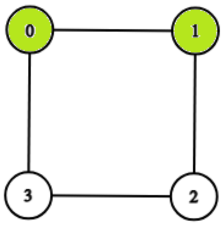
Entendemos entonces que un Grafo G=(V,E) (Siendo V los vértices y E los aristas) tendrá un conjunto dominante *(dominating set* en inglés) si tiene un subconjunto de vértices D de tal forma que cualquier vértice que NO está en D es al menos adyacente a uno que sí está incluido en el subconjunto. Por otro lado, k es el tamaño del conjunto dominante.

Como problema de decisión, lo enfocaremos como:

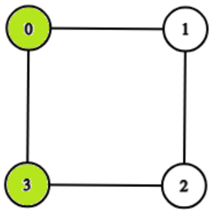
¿Existe algún conjunto de vértices C en G tal que cada arista de G tiene al menos un extremo en D y el tamaño de D es menor o igual a k?

Ejemplos válidos

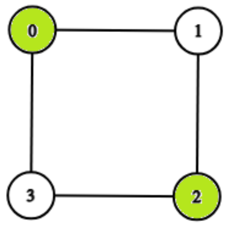
Un ejemplo muy sencillo de un conjunto dominante es un grafo de 4 vértices y 4 aristas. Cualquier par de vértices podrá ser aquí un conjunto dominante, puesto que de esa forma los otros dos vértices restantes forzosamente tendrán aristas que lo hagan adyacente al Conjunto Dominante. En éste ejemplo, k=2.



D={0,1}

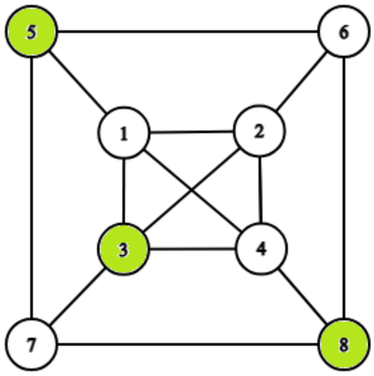


D={0,3}



D={0,2}

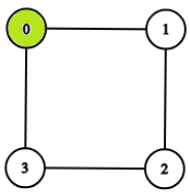
El siguiente grafo aunque un poco más complejo ilustra igualmente bien el problema de los conjuntos dominantes. Cuenta con 8 vértices y 14 aristas, y cada vértice verde (5, 3, 8) pertenecerá al conjunto dominante. (D={3,5,8}) Para cada vértice negro hay al menos un arista que lo hace adyacente a uno que sea verde. En éste caso, k=3.



D={3,5,8}

Ejemplo no válido

Por otro lado, un ejemplo muy claro de un grafo que no tiene conjunto dominante es el primer grafo que vimos, pero con sólo un vértice incluido en k. En éste ejemplo, sólo 0 forma parte del subconjunto de vértices señalados, y tanto 1 como 2 son adyacentes a él. Sin embargo, 3 no lo es y por ello el grafo no contará con un conjunto dominante.



D={0}

Campos de aplicación

Los conjuntos dominantes cuentan con muchas aplicaciones. Una muy notable es usarlos en los protocolos de enrutamiento para las redes inalámbricas ad hoc, buscando encontrar el conjunto dominante en una red de dispositivos y usar los nodos dominantes para encaminar los mensajes.

$\text{DOMINATING-SET} \in NP.$

Recordemos que la definición de NP es:

$NP \equiv \cup_k NTIME[n^k]$

Que como ya sabemos quiere decir que un problema pertenecerá a NP si se puede solucionar en tiempo polinómico con una máquina de Turing no determinista.

Podemos demostrar entonces que DOMINATING-SET está en NP de la siguiente forma:

Teniendo un grafo G y su conjunto de nodos dominantes C podemos comprobar fácilmente en tiempo polinómico que éste conjunto es dominante asegurándonos de que cada nodo dentro de G es adyacente a C o un nodo dentro de C.

El algoritmo sería, entonces:

Entrada: $< G, k >$
$\text{siendo G un grafo no orientado tal que G=(V,E) con V siendo sus vértices y E siendo sus aristas, y k siendo un}$

1. Escogemos k nodos de G. A éste conjunto lo llamaremos C.
2. Para cada nodo n en V:
   a. Comprobamos si está en C.
   b. Si no lo está, comprobamos que sea adyacente.
3. Si todos los nodos cumplen la condición del segundo paso, se devuelve $true$.
4. En cualquier otro caso, se devuelve $false$.

Ésta comprobación se puede hacer en $\mathrm{O}(n^2)$, siendo n el número de nodos.

Por ello, $\text{DOMINATING-SET} \in NP$.

Paso 2: $\text{VERTEX-COVER} \leq_p \text{DOMINATING-SET}$

Podemos encontrar un parecido bastante razonable entre el problema de los conjuntos dominantes y el de la cobertura de vértices.

Definición:

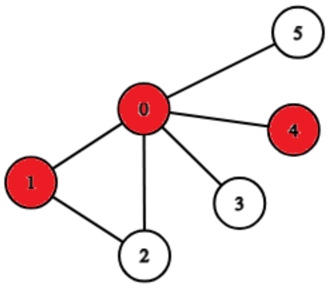
Formalmente, una cobertura de vértices C de un grafo no dirigido $G = (V, E)$ es un subconjunto de V tal que $uv \in E \implies u \in C \vee v \in C$ donde cada arista es adyacente al menos a un vértice dentro de C.

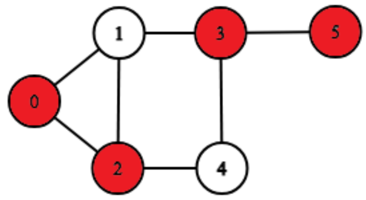
Como ya sabemos, se trata de un problema NP-complejo, que recordemos que significa que no puede ser resuelto por un algoritmo en tiempo polinómico si $P \neq NP$. Siendo uno de los problemas más estudiados de la informática teórica cuenta con una versión de decisión y una de optimización.

En su versión de decisión es NP-Completo, lo que quiere decir que es improbable encontrar un algoritmo eficiente que lo resuelva para grafos arbitrarios. Es demostrable por la reducción de 3-SAT o desde la reducción de clique, como lo hizo Karp. (Lo que lo convierte además en en uno de los 21 problemas NP-Completos de Karp.)

Ejemplos válidos

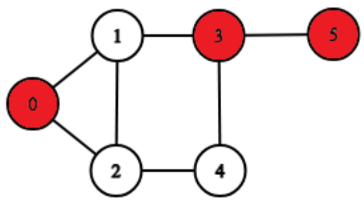
Éstos grafos son instancias válidas de cobertura de vértices. Todo arista incide en un vértice dentro de C (representados en rojo) :





Ejemplos no válidos

Éste grafo NO es válido. El arista de 2 a 4 no es incidente a ningún vértice del subconjunto.



Campos de Aplicación

La cobertura de vértices sirve como modelo para muchos problemas, tanto teóricos como aplicables al mundo real. Un ejemplo puede ser instalar el menor número de cámaras de seguridad para cubrir todos los pasillos (siendo éstos los aristas) que unen todas las habitaciones. (que serían los vértices)

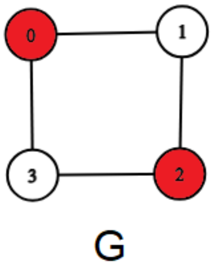
Entre otras aplicaciones encontramos la eliminación de secuencias de ADN repetitivas en la biología sintética o la ingeniería metabólica.

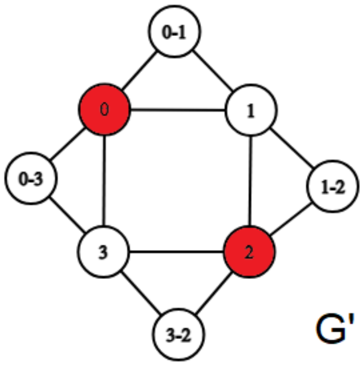
Gadget

Para $\text{VERTEX-COVER} \leq_p \text{DOMINATING-SET}$ usaremos como gadget una instancia del problema de cobertura de vértices que convertiremos en una del conjunto dominante de la siguiente forma:

Teniendo como entrada $< G, k >$ que como hemos visto son un grafo no orientado $G = (V, E)$ y un número entero k que indica el tamaño de la cobertura de vértices C dentro de éste mismo grafo, creamos un nuevo grafo $G' = (V', E')$ a partir de G.

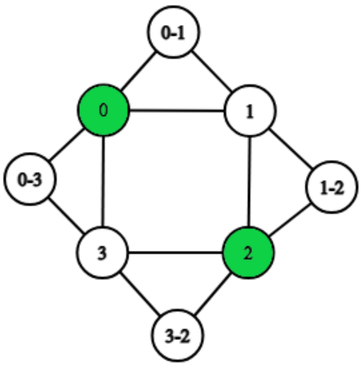
Nuestro objetivo para crear G' será hacer triángulos. G' tendrá los mismos vértices y aristas que G, pero les añadiremos además nuevos vértices, en concreto un nuevo vértice por cada arista de G de tal forma que para el arista $E = \{u, v\}$ (con extremos u y v) añadiremos un nuevo vértice u-v, que será conectado tanto con u como con v tal y como muestra la imagen.





Como asumimos que G tiene una cobertura de vértices C de tamaño k, por la propia definición de VERTEX-COVER toda arista del grafo debe ser adyacente a un vértice perteneciente a C. Esto se traduce en que para cada arista $E = \{u, v\}$ uno de los dos extremos será o parte de la cobertura de vértices o adyacente a ella. Lo interesante es que por ésta misma razón al haber "triangulado" G' añadiendo a cada arista un nuevo vértice u-v que está conectado tanto con su extremo u como con v, sí o sí éste también estará cubierto por C, demostrando que todos los vértices adicionales son también cubiertos por C.

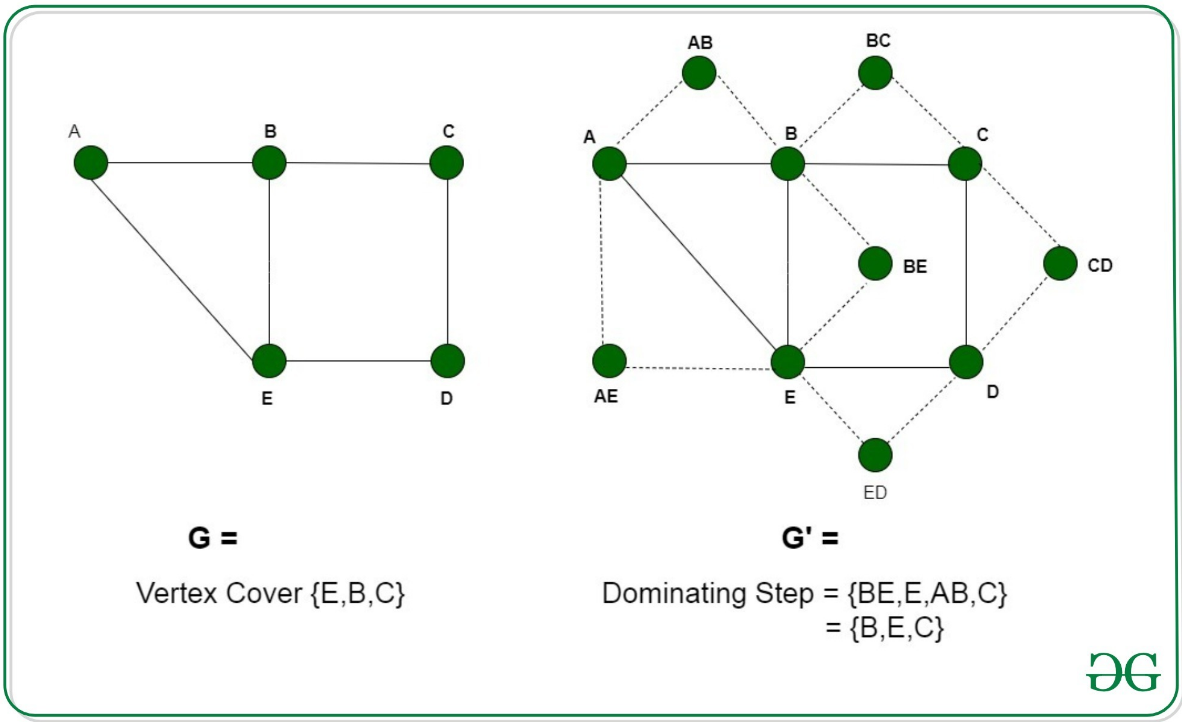
De ésta forma vemos que si G tiene una cobertura de vértices C de tamaño k, sí o sí G' tendrá un conjunto dominante del mismo tamaño.



Por otro lado, asumimos que G' tiene un conjunto dominante D de tamaño k. Pueden ocurrir entonces dos situaciones: Que los nodos dominantes sean parte de los vértices originales de G o parte de los vértices adicionales que surgieron de triangular las aristas. En cualquier caso podremos eliminar todos los vértices nuevamente añadidos, ya sea considerándolos redundantes porque no son nodos dominantes o en el caso de que sí lo sean reemplazando uno de los tres vértices del triángulo por el que sí que pertenece a D, manteniendo la dominancia.

Por lo tanto, si G' tiene un conjunto dominante D de tamaño k, G tendrá una cobertura de vértices C del mismo tamaño.

En la siguiente imagen el vértice B domina sobre AB y BE, haciéndolos redundantes.



Análisis de Complejidad

Puesto que la transformación consiste en hacer un nuevo grafo, la complejidad temporal se puede desglosar como:

Teniendo la entrada $< G, k > \text{con } G = (V, E)$, consideraremos que $|V| = n \text{ y } |E| = m$.

Para la creación creación de $G' = (V', E')$:

$|V'| = n + m$, (vértices originales más nuevos vértices por cada arista) es decir, $\mathrm{O}(n + m)$

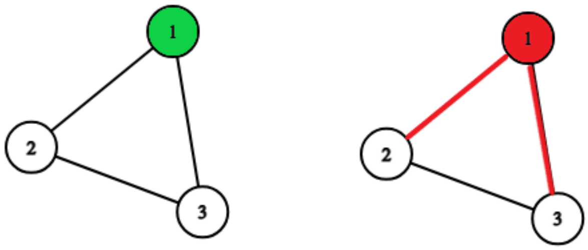
$|E'| = m + 2m$, (aristas originales más las dos aristas que conectan el nuevo vértice) lo que implica $\mathrm{O}(3m) = \mathrm{O}(m)$

En conclusión, puesto que la transformación se realiza en función del número de vértices y de aristas, la complejidad temporal será de $\mathrm{O}(n + m)$, que es efectivamente complejidad en tiempo polinómico.

Finalmente, tras haber demostrado que $\text{VERTEX-COVER} \leq_p \text{DOMINATING-SET}$ se hace en tiempo polinómico, concluimos que $\text{DOMINANT-SET} \in \text{NPC}$

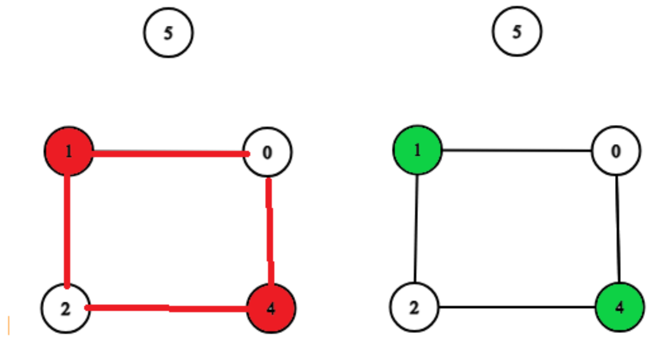
Preguntas

1. ¿Es siempre cierto que toda instancia de VERTEX-COVER es también una de DOMINANT-SET?
    a. Sí, por la transformación casi inmediata que hemos demostrado. Si todos los aristas están cubiertos, entonces todos los vértices son o parte de la cobertura o adyacentes a ellas.
2. ¿Es entonces cierta la situación inversa? Es decir, ¿Es una instancia de DOMINANT-SET siempre una VERTEX-COVER?
    a. No necesariamente. Un contraejemplo es el siguiente grafo, que irónicamente es muy sencillo. Se puede dar un conjunto dominante de k=1 en cualquiera de los vértices. Sin embargo si ese es el caso siempre va a haber un arista que no esté cubierto por la cobertura.



Dominant-Set {1} Vertex-Cover {1}

3. ¿Cuál es el grado mínimo que deben tener todos los vértices?
    a. Uno. El grafo debe de ser conexo. Los vértices aislados no tienen aristas y quedarían fuera de la transformación, y por lo tanto del conjunto dominante. Deberían ser forzados a ser nodos dominantes a posteriori. Sin embargo, el grafo sí que podrá tener una cobertura de vértices, puesto que en ese caso lo que nos interesan son los aristas, no los vértices.



Referencias

- Dominant Set of a Graph - GeeksforGeeks
- Dominating set - Wikipedia
- Vertex cover - Wikipedia
- Introduction and Approximate Solution for Vertex Cover Problem - GeeksforGeeks
- Proof that Dominant Set of a Graph is NP-Complete - GeeksforGeeks
- Proof that vertex cover is NP complete - GeeksforGeeks
- Vertex Cover - Álvaro García-Faure Torres (Universidad de Málaga)
- Dominating Sets and Domination Numbers in Graph Theory | Baeldung on Computer Science
- np hard - Reduction from Vertex Cover to Dominating Set - Computer Science Stack Exchange
- people.cs.umass.edu/~barring/cs611/exams/finpracsol.html
- Introduction to the Theory of Computation, Third Edition - Michael Sipser