Algoritmia y complejidad

Dominating-Set es NP Completo vía Vertex-Cover $\leq_p$ Dominating-Set

Daniel Sánchez Triviño

15 de mayo de 2024

Resumen

En este ejercicio tenemos que demostrar que el problema $Dominating - Set(DS) \in NPC$.
Para ello, usamos la reducibilidad con un problema que ya conocemos que es NP completo, el problema de $Vertex - Cover(VC)$.

Índice

1 Definición del problema

1.1 Definición

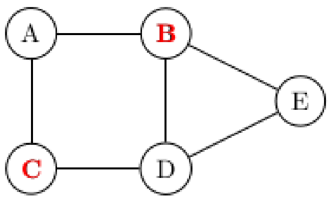
El problema del conjunto dominante o dominating set se define como:

> $DOMINATING-SET$ = {<G,k> | G tiene un conjunto dominante con k nodos}.

Dada la codificación de un grafo G y un número k, el problema **DOMINATING-SET** se define como el lenguaje formado por un conjunto de grafos G que tienen un conjunto dominante con k nodos. Un **conjunto dominante**[2] se define como un **subconjunto de nodos de G**, que para todo nodo del grafo G, este es **vecino de algun elemento del conjunto dominante o bien pertenece al conjunto dominante**.

1.2 Ejemplo válido

Un ejemplo de un grafo G y un número k=2 que pertenezcan al lenguaje del problema DOMINATING-SET sería el siguiente:



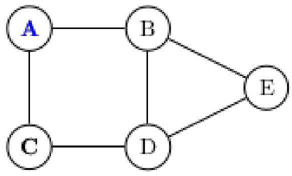
En este caso, se ha seleccionado **los nodos B y C** marcados en rojo como aquellos que **forman el conjunto dominante** DS de tamaño 2, y se comprueba que **es válido** ya que desde estos nodos, **todos los nodos de G son vecinos de algun nodo del conjunto DS o bien está en este**.

En nuestro ejemplo, se ve como el conjunto DS = { B,C } para los nodos de G:

1. A $\rightarrow$ Es vecino de B y C que pertenecen a DS
2. B $\rightarrow$ Pertenece a DS
3. C $\rightarrow$ Pertenece a DS
4. D $\rightarrow$ Es vecino de B y C que pertenecen a DS
5. E $\rightarrow$ Es vecino de B que pertenece a DS

1.3 Ejemplo no válido

En cambio, un ejemplo de grafo G y número k=1 que no pertenecerían al lenguaje DOMINATING-SET, sería:



En este caso, se han seleccionado el **nodo A** como el que forma el conjunto dominante, pero este **no es suficiente** ya que sea cual sea la combinación y elección de conjunto dominante que hagamos, siempre **habrá algún nodo que no sea vecino del conjunto dominante o pertenezca a este**, por lo cual este grafo G y este número k **no pertenecen al lenguaje DOMINATING-SET**

1.4 Campo de aplicación

Este problema, además de ser muy interesante desde el punto de vista teórico, también puede ser útil para distintos campos de aplicación como:

- **Redes de comunicación**, para obtener los puntos donde colocar routers/repetidores de forma que todos los nodos queden cubiertos por al menos un dispositivo
- **Redes de sensores**, para detectar aquellas zonas donde debe haber un sensor obligatoriamente para tener la zona de estudio/interés cubierta
- **Videovigilancia**, ya que se debe cubrir todos los puntos sin dejar ningún punto ciego, con el número mínimo de cámaras para controlar la zona

2 Demostración DOMINATING-SET $\in$ NPC

Queremos demostrar que el problema **DOMINATING-SET** que hemos definido previamente es **NP completo**. Para ello, primero necesitamos realizar algunas comprobaciones dado que vamos a usar la reducibilidad[1] de **Vertex-Cover** $\leq_p$ **Dominating-Set**.

2.1 DOMINATING-SET $\in$ NP

Para demostrar que un problema pertenece a la clase NP[3], puede hacerse por dos vías:

- Con **verificador polinómico**
- Con una **máquina de turing no determinista** que opera en tiempo polinómico

Usando un verificador polinómico V tendría que:

- El **certificado C** para <G,k>sería el conjunto dominante DS con k nodos
- El **algoritmo** sobre la entrada <G,k>sería:
  1. Generar un **conjunto D** de manera que se elija de forma no determinista k nodos distintos del grafo G $\approx \mathcal{O}(n)$
  2. Para cada elemento del conjunto D, **comprobar los nodos que son vecinos de este y apuntarlo** $\approx \mathcal{O}(n^2)$
  3. Consultar los **nodos apuntados**, y si estos más los elementos del conjunto D son todos los nodos del grafo G, aceptar, sino rechazar $\approx \mathcal{O}(n^2)$

  Finalmente, el algoritmo tendría una complejidad $\boldsymbol{\mathcal{O}(n^2)}$

Como se ha visto que el verificador opera en tiempo polinómico, podemos concluir que el problema **DOMINATING-SET pertenece a la clase NP**.

2.2 Reducción Vertex-Cover $\leq_p$ Dominating-Set

2.2.1 Definición de Vertex-Cover

Antes de abordar la reducibilidad que necesitamos, vamos a **describir el problema** desde el cual se hace la reducibilidad.

> Vertex-Cover = {<G,k>| G tiene una cobertura de vértices C tal que |C| $\leq$ k }.

Dado un grafo G y un entero positivo k, el problema **Vertex-Cover** se define como el lenguaje formado por el conjunto de grafos G tal que tienen un cubrimiento de vértices con un **subconjunto de vértices C**, cuyo **cardinal es menor o igual que el entero k**.

El **cubrimiento de vértices** es un subconjunto de los vértices de G tal que **cada arista del grafo es incidente en al menos un vértice** del subconjunto.

De manera informal, sería decir algo como que: ”Toda arista de G esta recogida por algún vértice de un subconjunto de los nodos de G”.

Un **ejemplo** sería el siguiente:

En este caso, se tiene un grafo G con **6 vértices**, y un **k**=**3** como mínimo, ya que se usa un subconjunto C de tamaño 3 para el cubrimiento de vértices (los marcados en rojo).

El problema de cubrimiento de vértices, tiene **multitud de aplicaciones** como por ejemplo:

- **Construcción de carreteras**, de modo que se consiga el mínimo número de vías para conectar todas las ciudades
- **Redes de comunicación**, al igual que anteriormente dije con el problema Dominating-Set, Vertex-Cover es útil para detectar los puntos claves donde instalar repetidores o estaciones base, así como minimizar los costes, al poner los mínimos dispositivos.
- Etc.

2.2.2 Gadget

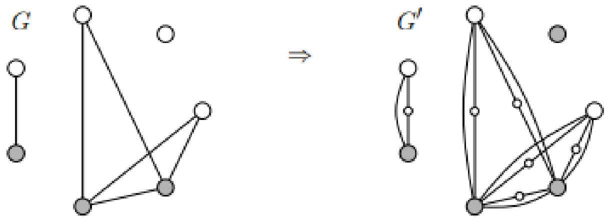
La principal diferencia entre los problemas Vertex-Cover y Dominating-Set es la **condición de cada problema**.

Sea V' un subconjunto de vértices de un grafo G: En VC, cada arista es incidente en un vértice de V', mientras que en DS, cada vértice está en V' o es adyacente a un vértice de V'.

Teniendo esto en cuenta, podemos construir nuestro **gadget** para partir de un **grafo G de VC y un número k**, y llegar a un **grafo G' de DS y un número k'**, de forma que mapee las aristas de G a vértices en G', de modo que **una arista incidente en G se mapea como un vértice adyacente en G'**. La idea por tanto quedaría como que:

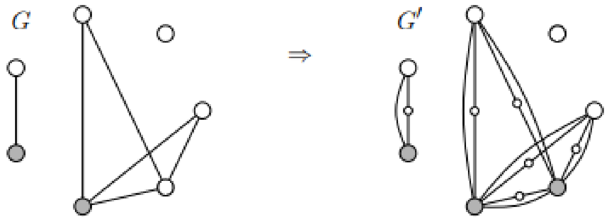
- Para cada arista $\{u, v\}$, crearemos un **vértice especial** w_{uv} y sustituimos la arista $\{u, v\}$ con dos aristas, $\{\{u, w_{uv}\}$ y $\{v, w_{uv}\}$. También dejamos la arista $\{u, v\}$.
- Teniendo en cuenta el número de **vértices aislados** I_v, definimos $k' = k + I_v$.

Un **ejemplo válido** del gadget sería el siguiente:



Aquí se ve como pasamos de un **grafo G** con un cubrimiento de vértices con **k** = 3, a un **grafo G'** con **k'** = 4 debido al nodo aislado.

En cambio, un **ejemplo no válido** sería por ejemplo el anterior pero con un **k** = 2, quedando por ejemplo:



En este caso, **es insuficiente un subconjunto de tamaño 2** para conseguir el cubrimiento de vértices de G, por lo que al aplicar el gadget también **es imposible** encontrar el conjunto dominante de tamaño k'= 3 en G'.

2.2.3 Demostración del gadget

Hemos visto como usar el gadget para reducir el problema VC al problema DS, pero debemos **comprobar que realmente el gadget funciona** para relacionar ambos problemas.

Para esto, vamos a **demostrar** que:

1. Si **<G,k>** $\in$ **VC** $\rightarrow$ **f(<G,k>)** $\in$ **DS**
2. Si **f(<G,k>)** $\in$ **DS** $\rightarrow$ **<G,k>** $\in$ **DS**

1. ($\rightarrow$) Si <G,k>$\in$ VC, $\exists V'$ cubrimiento de vértices para G, entonces $V'' = V' \cup V_I$ es un conjunto dominante para G'.
   Se cumple que |V''| = $|V' \cup V_I| \leq k + I_v$ = k'
   Si **V'' es un conjunto dominante**, entonces todos los vértices aislados están en V'' y son dominantes

   Además, los **vértices especiales** w_{uv} de G' corresponden con una arista $\{u, v\}$ en G, por lo que u o $v \in$ V', por tanto w_{uv} es dominado por el mismo vértice en V''.

   Finalmente, cada **vértice no aislado** v es incidente al menos en una arista de G, por lo que **o bien está en V' o todos sus vecinos están en V'**, lo que implica que $v \in$ V'' o es adyacente a un vértice en V''.

2. ($\leftarrow$) Si f(<G,k>) $\in$ DS, $\exists V''$ conjunto dominante de tamaño k' = k + I_v, entonces $V''' = V'' \setminus$V$_I$ y para todo vértice especial $w_{uv} \in V'''$ lo sustituimos por u, que sigue dominando u, v y w_{uv} debido a las aristas que van hacia v y w_{uv} desde u, por lo que **dominamos los mismos vértices** que antes de sustituir.

   Por tanto, siendo V' el **conjunto resultante** tras esta sustitución, V' es un **cubrimiento de vértices** para G, ya que si por lo contraria había una arista $\{u, v\} \in G$ que no estaba cubierta entonces el vértice especial w_{uv} no sería adyacente a ningún vértice de V'' en G', llegando a un **absurdo** debido a que **V'' se suponía era un conjunto dominante para G'**.

De este modo, concluimos que nuestro gadget o función f **cumple con la propiedad de la reducibilidad y es válido**, dado que **se cumple la implicación en ambos sentidos**.

2.2.4 Cálculo del tiempo de la función

Para poder concluir que Dominating-Set $\in$ NPC, debemos ver que nuestro gadget o función f, **opera en tiempo polinómico**, por lo que voy a estudiar su complejidad de forma aproximada en base a un grafo G con tamaño **n** y un número **k**:

- **Recorrer G**, para **pintar todo vértice** de este en el nuevo grafo G' $\approx \mathcal{O}(n)$
- Para cada arista de G, **crear un vértice especial** en G' y **unirlo con dos aristas** $\approx \mathcal{O}(n^2)$ (ya que el número máximo de aristas de G es **n*(n-1)/2** y es lo que se consulta)

- Definir **k'** contando los **vértices aislados** de G' $\approx \mathcal{O}(n)$

Con esto, vemos que mi función f tendría una complejidad aproximada de $\mathcal{O}(n^2)$, por lo que opera en tiempo polínomico y podemos concluir que **Dominating-Set** $\in$**NPC.**

3 Preguntas sobre el problema

Las **preguntas** que se me han ocurrido que podrían ser más interesantes sobre este problema serían:

- **¿Qué relación existe entre el problema VC y DS?** $\rightarrow$ Sea V' un subconjunto de vértices del grafo G: En VC, cada arista es incidente en un vértice de V', mientras que en DS, cada vértice está en V' o es adyacente a un vértice de V'.
- **¿Cómo funciona el gadget relacionando el grafo G de VC y el grafo G' de DS?** $\rightarrow$ Para cada arista $\{u, v\}$, crearemos un vértice especial w_{uv} y sustituimos la arista $\{u, v\}$ con dos aristas, $\{\{u, w_{uv}\}$ y $\{v, w_{uv}\}$. También dejamos la arista $\{u, v\}$.
- **¿Cómo funciona el gadget sobre el número k de VC y el número k' de DS?** $\rightarrow$ Teniendo en cuenta el número de vértices aislados I_v, definimos $k' = k + I_v$.

4 Referencias

Referencias

[1] Alber, J., Fellows, M. R., and Niedermeier, R. (2004). Polynomial-time data reduction for dominating set. *Journal of the ACM*, 51(3):363–384.

[2] Allan, R. B. and Laskar, R. (1978). On domination and independent domination numbers of a graph. *Discrete Mathematics*, 23(2):73–76.

[3] Garey, M. R. and Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Series of Books in the Mathematical Sciences. W. H. Freeman and Company, New York, 1st edition. p. 190, problem GT2.