Vertex Cover

Felipe Andrés Álvarez Avaria
Javier Alberto Martín
Alejandro Hernández Padrón
Abel Delgado Falcón

Diciembre 2018

Índice

Introducción

Vertex Cover

Conjunto de vértices tales que cada arista del grafo incide en al menos un vértice del conjunto.

NP-Completitud

El Vertex Cover es un problema NP-completo.
Sabemos que es NP-completo por que hay otros problemas que son NP-Completo y son polinaomialmente reducibles a él.

Problemas

3-SAT

Instancia: Se establece un conjunto U de variables y una colección $C = \{c_1, c_2, \ldots, c_m\}$ de cláusulas sobre U, de manera que cada cláusula $|c_i| = 3$, para $1 \leq i \leq m$.
Pregunta: ¿Hay una asignación de verdad para U que cumpla con la colección de cláusulas C?

Vertex Cover

Instancia: Un grafo $G = (V, E)$ no dirigido y un entero K.
Pregunta: ¿Existe una cobertura de vértices de tamaño K o menos para el grafo G, es decir, un subconjunto V' de V con tamaño menor que K tal que cada arista tiene al menos un extremo en V'?

Transformación de 3-SAT a VC

Sea $U = \{u_1, u_2, \ldots, u_n\}$ y $C = \{c_1, c_2, \ldots, c_m\}$ una instancia de 3-SAT .

Debemos construir un grafo $G = (V, E)$ y un entero positivo $K \leq |V|$ tal que G posee un recubrimiento de vértices de tamaño K o menos si y solo si C es satisfactible.

Lo primero que haremos es forzar una opción para cada variable a $verdadero$ o $falso$ al tener un par de vértices para cada literal y su negación. Así que cada $x_i \in U$ producirá dos vértices $V_i = \{x_i, \bar{x_i}\}$ y una arista que una los dos ejes $E_i = \{\{x_i, \bar{x_i}\}\}$.

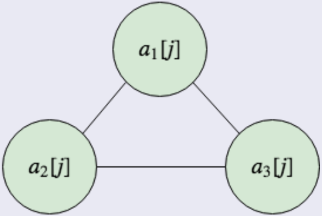
Transformación de 3-SAT a VC II

Paso 2

A continuación, creamos un 'gadget' para representar las cláusulas. Lo que haremos será que para cada cláusula $c_j \in C$ crearemos un triángulo de tres vértices donde cada vértice esté conectado a modo de triángulo:

$$V'_j = \{a_1[j], a_2[j], a_3[j]\}$$

$$E'_j = \{\{a_1[j], a_2[j]\}, \{a_1[j], a_3[j]\}, \{a_2[j], a_3[j]\}\}$$



Transformación de 3-SAT a VC III

Paso 3

Por último para cada cláusula $c_j \in C$ sean los tres literales en c_j denotados como x_j, y_j y z_j. Se crearán ejes de comunicación que partirán desde los vértices de nuestros 'gadgets' con sus literales correspondientes:

$$E''_j = \{\{a_1[j], x_j\}, \{a_2[j], y_j\}, \{a_3[j], z_j\}\}$$

Paso 4

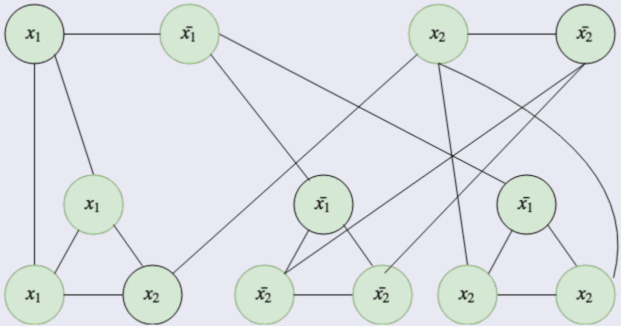
Por lo tanto terminamos que la construcción del grafo $G = (V, E)$ es:

$$V = (\bigcup_{i=1}^{n} V_i) \cup (\bigcup_{j=1}^{m} V'_j)$$

y

$$E = (\bigcup_{i=1}^{n} E_i) \cup (\bigcup_{j=1}^{m} E'_j) \cup (\bigcup_{j=1}^{m} E''_j)$$

Transformación de 3-SAT a VC IIII



Demostración de NP-completitud

Finalmente, debemos elegir un K apropiado para la instancia de Vertex Cover. Elegiremos $K = n + 2m$ donde n es el número de literales y m es el número de cláusulas.

Si hay una asignación booleana que satisface 3-SAT , podemos hacer que los literales verdaderos formen parte de nuestra cobertura de vértices en G. Eso es n nodos de nuestra cobertura. Además, al elegir un nodo para cada literal en V_i, hemos cubierto la arista E_i entre ellos. Observamos que ya que hemos cumplido con la fórmula 3-SAT , debemos haber elegido al menos uno de los literales en cada cláusula para que sea $verdadero$, esto asegura que al menos uno de los tres ejes de cada conjunto E''_j esté cubierto.

Demostración de NP-completitud II

Ahora, solo necesitamos dos vértices (o menos) de cada triángulo para cubrir el resto de las aristas. Esto produce como máximo $2m$ vértices más.

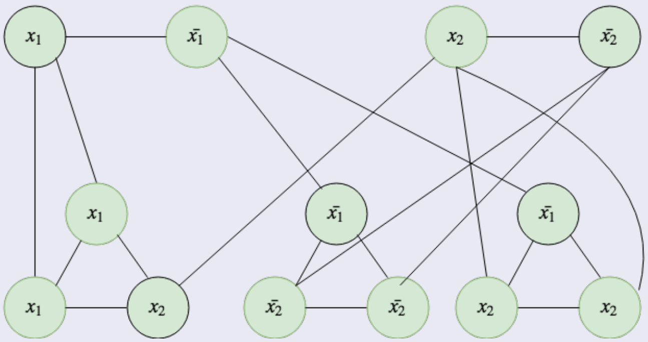
Necesitamos incluir en nuestra cobertura de vértices los puntos finales de cada triángulo de los otros dos vértices en E_j'' (el cual puede o no estar cubierto por otros vértices de los componentes de asignación) y esto nos da el recubrimiento de vértices deseado.

Demostración de NP-completitud III

Finalmente, esta construcción toma como entrada las cláusulas m y n literales y crea un gráfico con exactamente $2n + 3m$ nodos.

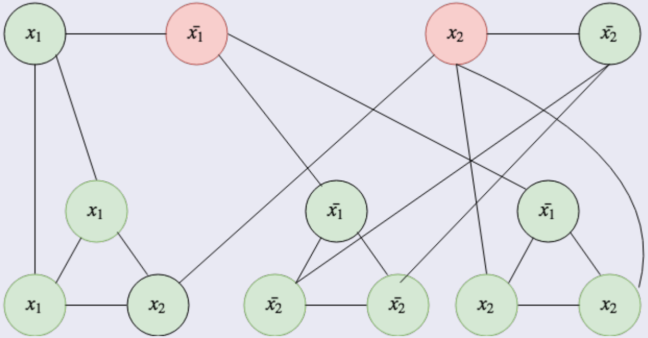
Esto se puede hacer fácilmente en tiempo polinomial. Por lo tanto concluimos que el Vertex Cover es $\mathcal{NP}$-completo

Ejemplo

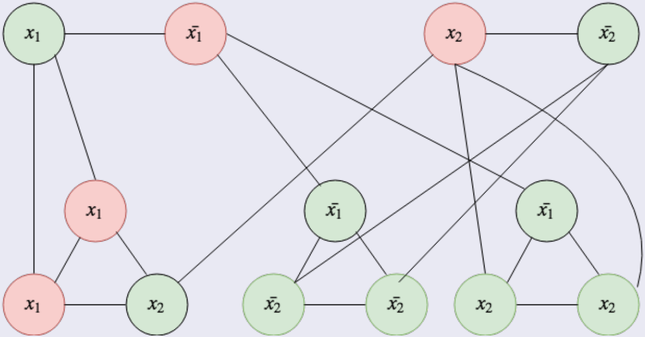


Ejemplo II

x1 = FALSE; x2 = TRUE

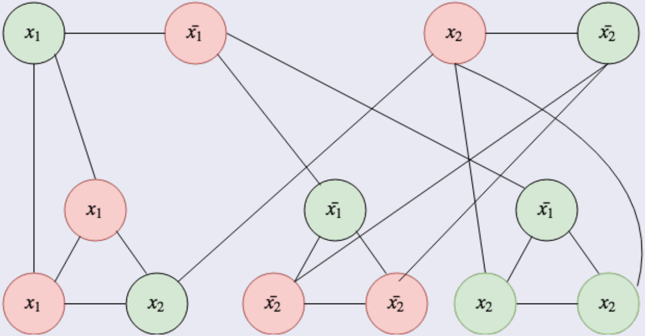


x1 = FALSE; x2 = TRUE



Ejemplo IV

x1 = FALSE; x2 = TRUE



Ejemplo V

x1 = FALSE; x2 = TRUE