ALGORITMO DE COBERTURA DE VÉRTICES

Vertex cover algorithm

JAVIER LÓPEZ WONG[1]

Recibido:16 de marzo de 2022. Aceptado:10 de junio de 2023

DOI: http://dx.doi.org/10.21017/rimci.2023.v10.n20.a146

Resumen

Problema a resolver P=NP, utilizando el problema de cobertura de un grafo que es NP y convertirlo a P.En la disciplina matemática de la teoría de grafos, una cobertura de vértices simplemente cobertura de un grafo, es un conjunto de vértices tales que cada arista del grafo es incidente a al menos un vértice del conjunto.El problema de encontrar la menor cobertura de vértices en un grafo se denomina problema de la cobertura de vértices. En teoría de la complejidad computacional se ha demostrado que este es un problema NP-completo.Un problema NP-completo es que no se sabe si tiene solución Polinomial.He encontrado un algoritmo que demuestra que es Polinomial.

Palabras clave: Diseño y análisis de algoritmo; matemática; optimización.

Abstract

Problem to solve P=NP, using the coverage problem of a graph that is NP and convert it to P. In the mathematical discipline of graph theory, a vertex cover, simply a graph cover, is a set of vertices such that each edge of the graph is incident to at least one vertex of the set. The problem of finding the smallest vertex coverage in a graph is called the vertex coverage problem. In computational complexity theory, it has been shown that this is an NP-complete problem. An NP-complete problem is that it is not known if it has a Polynomial solution. I have found an algorithm that proves that it is Polynomial.

Keywords: Algorithm design and analysis; Mathematics; Optimization.

I. Introducción

Dado el grafo G (V, E)

Si dado que encuentro un conjunto de vértices que cubran el grafo y extraigo los vértices que solamente están conectados a ellos dentro del mismo conjunto se convierte en minimal pues todos los vértices tuviesen una arista de E a otro vértice que no pertenece al conjunto escogido, no se cubre E. No existiera una cobertura de un grafo.

Llamémosle a ese conjunto de vértices minimal R entonces un minimal puede ser mínimo o no serlo. Ejemplo: En el grafo de la derecha, {1,3,5,6} es una cobertura de vértices de tamaño 4. Sin embargo, esta no es la cobertura mínima, porque existen las coberturas de vértices {2,4,5} y {1,2,4}, ambas de tamaño 3. Fig. 1.

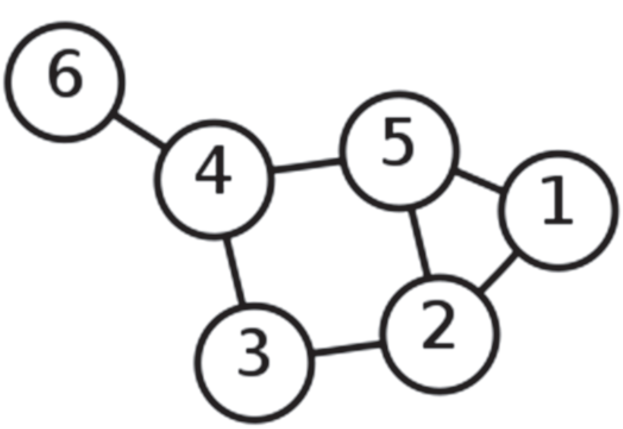

Fig. 1. Grafo de 6 vértices.

1 Licenciado en computación. Universidad de la Habana, la Habana, Cuba. ORCID: https://orcid.org/0009-0000-3368-6107 Correo electrónico: javierlopezwong@gmail.com

II. Materiales y métodos

Algoritmo:

Paso 1:

Encontrar un conjunto de vértices que cubran el grafo.

Paso 2:

Extraer los vértices que son redundantes con esto obtengo un conjunto de vértices minimal K.

Paso 3:

Encontrar el mínimo cubrimiento dado ese minimal K.

Si todos los subconjuntos K1,..Ki tal que los Ki estén desconectados y conectados a V-K, sean U1 ,..,Ui respectivamente.

Y Li subconjunto de Ui entonces si tenemos todos los Li con su correspondiente Ni subconjunto del Ki conectados tal que cualquier vértice de Ni conectado a al menos un vértice de Li y |Ni| <=|Li| entonces quitar esos vértices de Ki.

Ki=Ki-Ni
Ui=Ui-Li.

Entonces me quedo con el menor K-Ki donde sea la mayor |Ki|-|Ui| se tiene el K-Ki mínimo.

K=K-Ki
K=K unión Ui

Se itera el paso 3 hasta que para cualquier |Li|>=|Ni|.

III. Resultados y discusión

Entonces dado un minimal R encontrar el conjunto mínimo, minimizando ese conjunto encontramos un minimal mínimo.

Demostración:

Si quisiéramos extraer m vértices del conjunto minimal y poner t vértices que no pertenecen a R sea {v1 ,…,vt} los vértices que voy a extraer si un vi y vj tal que la arista (vi,vj)=1 con 1<=i,j<=m entonces cuando se extrae queda esa arista fuera de R y no existe una cobertura de un grafo.

Entonces para todo vi,vj las aristas (vi,vj)=0 de los m vértices que voy a extraer (llamémosles vértices desconectados entre sí).

Luego podemos crear todos los conjuntos Ki máximos (los que más vértices tengan) desconectados entre sí, se puede crear hasta |R| cantidad de subconjuntos Ki empezando por cada uno de los vértices del conjunto R el máximo Ki si cumplen con la propiedad de desconectados entre sí.

Entonces si extraigo un subconjunto Ni de Ki que no pertenezca a otro Kj entonces Ni estaría conectado al cada vértice con al menos un vértice de Kj. Si no estuviese conectado un vértice u1 de Ni con ninguno de Kj, entonces Kj no fuese máximo porque le podría añadir u1 al conjunto Kj.

Entonces no puedo extraer de los demás Kj. Buscamos el conjunto Ki tal que minimice a R.

En la buscamos en todos los Ki, el que más vértices extraiga – vértices que entran al conjunto minimal R.

Los Ki si tomamos los Ni como los conjuntos conectado a Li subconjunto de V-R (y |Li|>=|Ni| y todos los vértices de Ni conectados a todos los vértices Li pues no minimiza a R, Ki=Ki-Ni).

Ahora quedándome con el conjunto M=R–Ki donde todos los Ni que no cumplen la propiedad anterior. El Ki que minimice a R con ese lo intercambio por Ui subconjunto de V-R tal que todo vértice Ki tenga al menos una arista a un vértice de Ui (el que me dé |Ki| - |Ui| mayor) da el minimal de menor tamaño M.

Ahora tenemos K y |K|=r :

1-No se puede crear un conjunto V-K más grande porque sí escojo k vértice entonces o hay una arista de los vértices que había cambiado hacia él (no existe un cubrimiento en K-{1,..,k}) o son un subconjunto de los vértices que hay que cambiar por lo menos con otro vértice más vértices que voy a agregar a U ya que de los vértices desconectados entre sí no mejoran a R entonces no maximizan a V-K (porque para cualquier Li subconjunto de U que tenga Ni subconjunto de algún Ki conectado a ellos |Li|>=|Ni| no puedo cambiar nada en ningún Ki). Por lo tanto, el mínimo sigue siendo K.

IV. Conclusiones

Se llega a la conclusión de que el algoritmo funciona y lo hace en tiempo polinomial.

Con un O(k^5) donde k<=n, n = |V|.

Los problemas NP son P.

V. Agradecimientos

Agradecimientos:A mi familia, profesores y amigos. También a mi centro de estudio Universidad de la Habana, Facultad Matemática y Computación.

Fig. 2. Logotipo institucional.

VI. Contribución

Algoritmo creado en Python.

https://drive.google.com/file/d/17ni7_HSY4uLppdJbW6IEj43Z16fEQwHS/view?usp=share_link