

Vertex cover

Teoría de Algoritmos I (75.29 / 95.06)

Ing. Víctor Daniel Podberezski

✉ vpodberezski@fi.uba.ar

Problema Vertex cover

Sea

Grafo $G=(V,E)$

Cada vértice i tiene un peso $w_i \geq 0$

Queremos

encontrar el Set $S \subseteq V$ donde cada arista E del grafo pertenezca a algún vértice de S .

Minimizando el costo de los vértices seleccionados.

Costo pagado

Existen

Diferentes subset S de V que conforman un vertex cover

Llamaremos

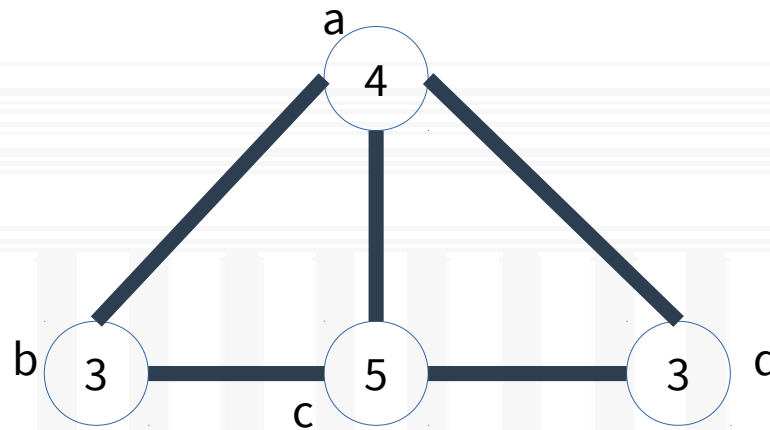
$w(S)$ como el costo del vertex cover formado por $S \subseteq V$

La solución optima S^*

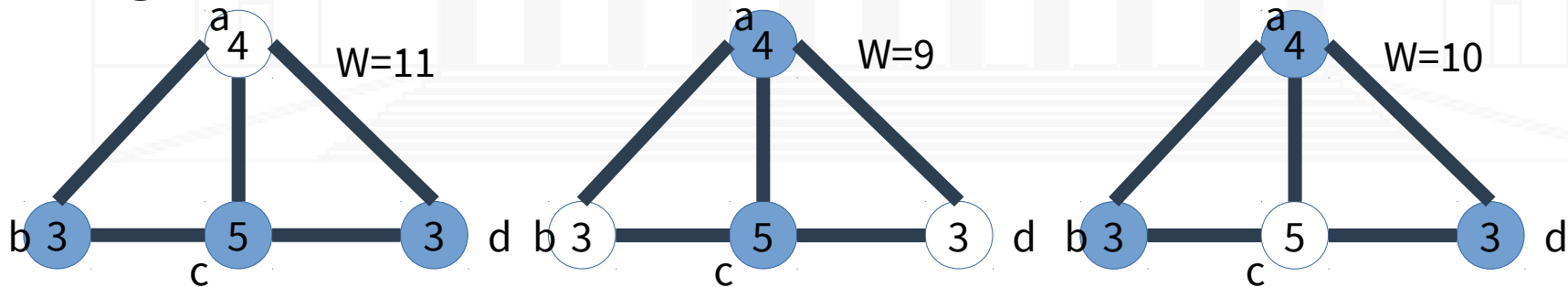
Es aquella para la que $w(S^*) \leq w(S)$ para todo S

Ejemplo

Sea el siguiente gráfico



Las siguientes son coberturas de vértices



Pricing method (A.K.A “primal-dual method”)

Basados en una perspectiva económica

Podemos pensar cada peso de los vértices como un “costo”

A cada vértice se le debe pagar por pertenecer a la solución

Cada eje es un “agente” dispuesto a pagar algo al vértice que lo cubre.

Diremos que un vértices esta pagado

si la suma de lo pagado por sus ejes es igual al costo del vértice.

Diremos que un precio a pagar P_e por el vértice $e=(i,j)$ es “justo”

si P_e mas la suma de los otros pagos de los ejes incidentes a i no superan w_i
(idem para j)

Algoritmo propuesto

Definir $p_e = 0$ para todo $e \in E$

Mientras exista un eje $e=(i,j)$ tal que i o j no este “pagado”

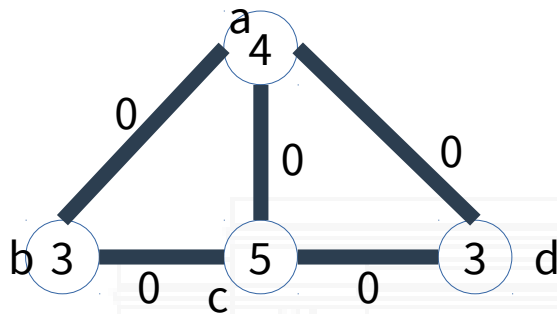
 Seleccionar el eje e

 Incrementar p_e sin violar la integridad

Sea S el set de todos los nodos pagados

Retornar S

Ejemplo

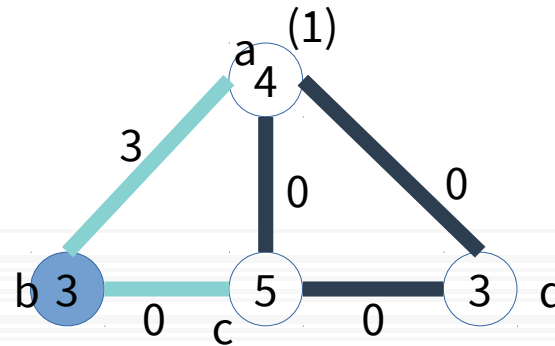


Inicialmente:
 $p(e)=0$ para todo $e \in E$
 Nodos pagados: \emptyset

Selecciono
 Eje $e=(a,b)$

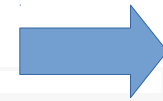


Puedo
 pagar
 hasta 3
 $\min(4, 3)$

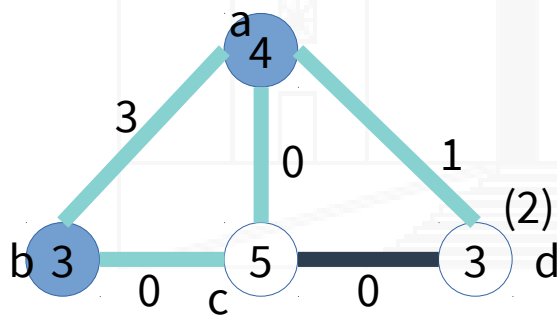


Nodos pagados: b

Selecciono
 Eje $e=(a,d)$



Puedo
 pagar
 hasta 1
 $\min(1, 4)$

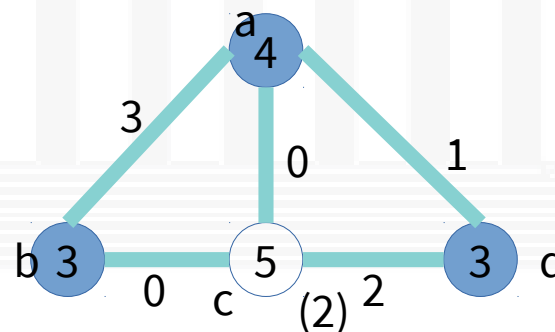


Nodos pagados: b,a

Selecciono
 Eje $e=(c,d)$



Puedo pagar
 hasta 2
 $\min(5, 2)$



Nodos pagados: b,a,d

Finalizacion:
 No quedan ejes sin
 cubrir
 Vertex cover: a,b,d
 $W(s): 10$

Análisis del algoritmo

Para

cualquier vertex cover S^* ,
cualquier precio justo P_e , tenemos que

Podemos calcular

$$w(S) = \sum_{i \in S} w_i$$

Por definición de integridad

tenemos que para todos los nodos $i \in S^*$

$$\sum_{e=(i,j)} P_e \leq w_i$$

(!)El algoritmo –
dependiendo la
manera de elegir,
puede construir
cualquier vertex
cover del Grafo (!)

Análisis del algoritmo (cont.)

Sumando las desigualdades:

$$\sum_{i \in S^*} \sum_{e=(i,j)} p_e \leq \sum_{i \in S^*} w_i = w(S^*).$$

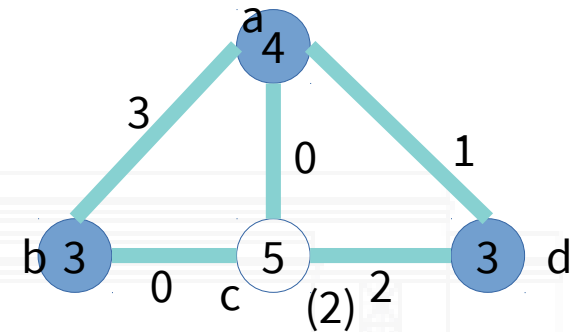
Por otro lado sabemos que

$$\sum_{e \in E} p_e \leq \sum_{i \in S^*} \sum_{e=(i,j)} p_e.$$

Finalmente combinamos y obtenemos:

$$\sum_{e \in E} p_e \leq w(S^*),$$

La suma de lo pagado por los ejes es menor a igual al costo de la cobertura de vértices



$S^*: b, a, d$
 $w(S^*)=10$
pagado=6

Análisis del algoritmo (cont.)

Sea S el set retornado por el algoritmo

Todos los nodos en S están “pagados”

por lo que para todo i en S : $\sum_{e=(i,j)} p_e = w_i$

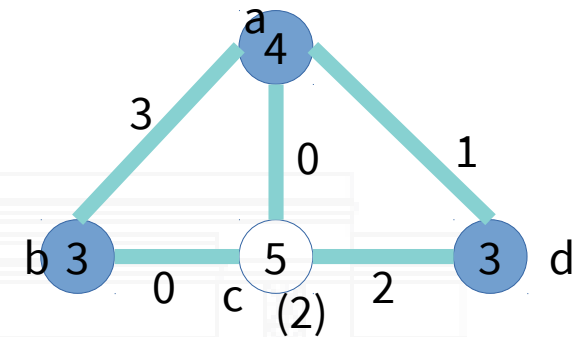
Podemos por lo tanto expresar el costo de S

como
$$w(S) = \sum_{i \in S} w_i = \sum_{i \in S} \sum_{e=(i,j)} p_e.$$

Un eje $e=(i,j)$ puede sumar peso a dos vértices (aun sin estar en S)

por lo que:
$$w(S) = \sum_{i \in S} \sum_{e=(i,j)} p_e \leq 2 \sum_{e \in E} p_e,$$

El costo de la cobertura esta acotado por 2 veces la suma de los precios pagados



$S: b, a, d$
 $w(S)=10$
 $2 * \sum(p_e)=12$

Análisis del algoritmo (cont.)

Por funcionamiento del algoritmo

El set S obtenido es un vertex cover (sino no termina el “mientras”).

Por los definiciones que probamos anteriormente:

$$\sum_{e \in E} p_e \leq w(S^*),$$

$$w(S) = \sum_{i \in S} \sum_{e=(i,j)} p_e \leq 2 \sum_{e \in E} p_e,$$

$$w(S) \leq 2 \sum_{e \in E} p_e \leq 2w(S^*).$$

El costo del set S retornado por el algoritmo es como mucho el doble de algún vertex cover posible.

El algoritmo es un 2-algoritmo de aproximación



Presentación realizada en Julio de 2020