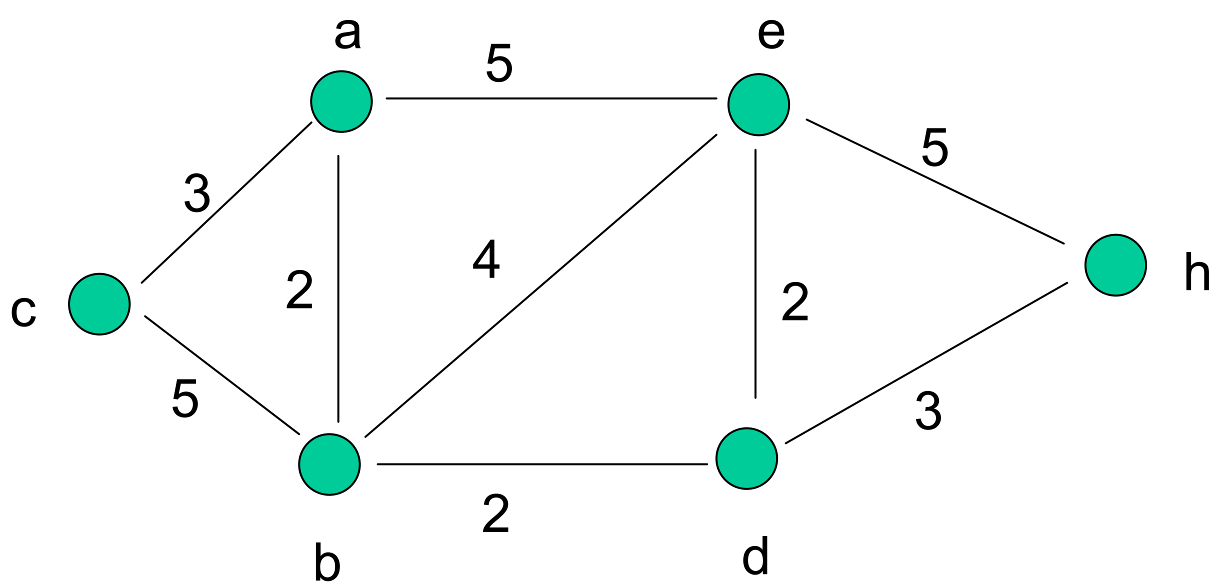
Universidad Politécnica
de Madrid

FLUJOS EN GRAFOS
GOMORY-HU Trees

Gregorio Hernández
UPM

Optimización Combinatoria

Red de transporte no dirigida R = (G,c)



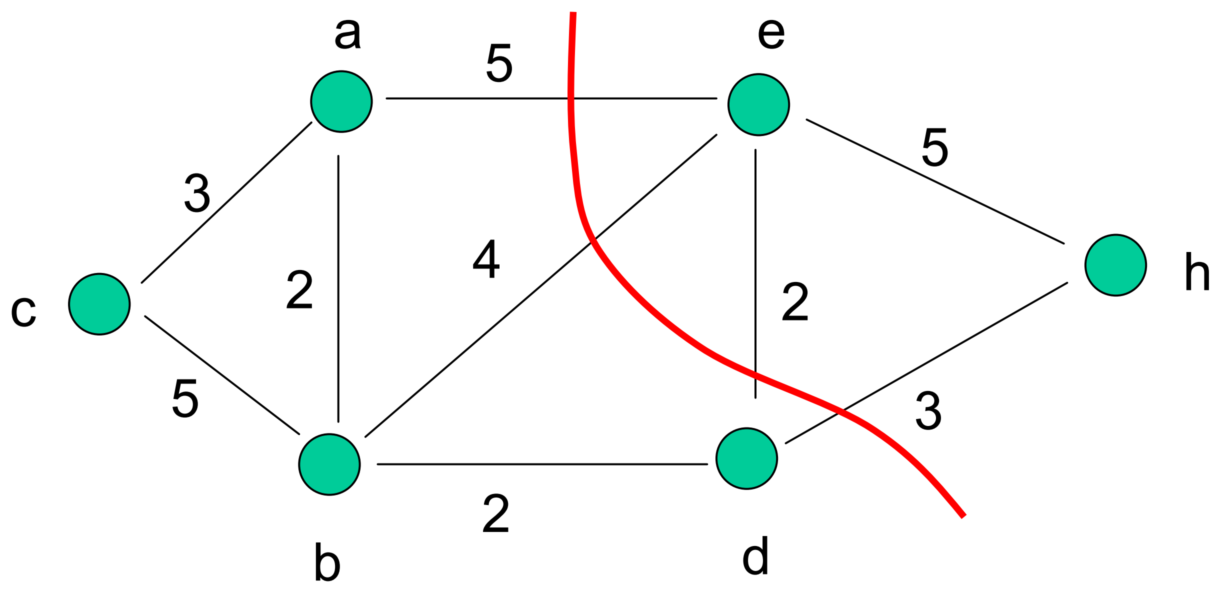
FLUJO en una red R $\quad f : A \rightarrow Z^{+}$

Hallar un flujo de valor máximo entre todos los pares de vértices de la red

Aplicar el algoritmo de Edmonds-Karp para cada par

$O(n^2)$ flujos !!

Red de transporte no dirigida R = (G,c)



CORTE en una red R es una partición de V en dos conjuntos S y T

Capacidad de un corte $\text{cap}(S,T) = \sum_{x \in S, y \in T} c(x,y)$

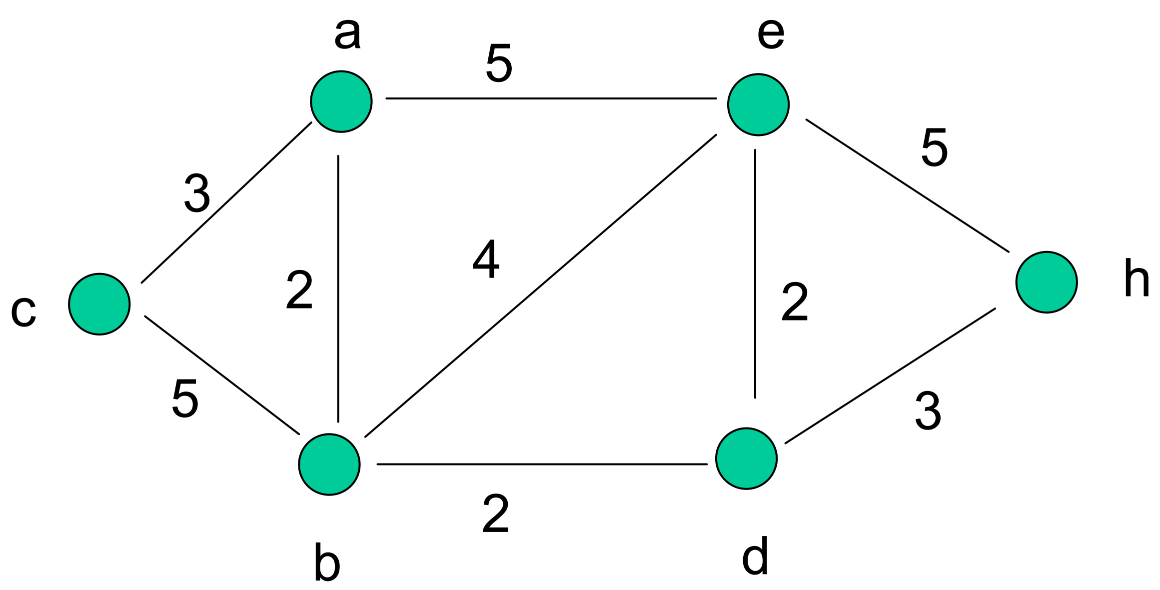
Cap(rojo) = 14

Objetivo:
Hallar un flujo de valor máximo entre cada par de vértices

Teorema de Ford-Fulkerson

El valor máximo de un flujo en una red N es igual a la mínima capacidad de los cortes de N

Red de transporte no dirigida R = (G,c)

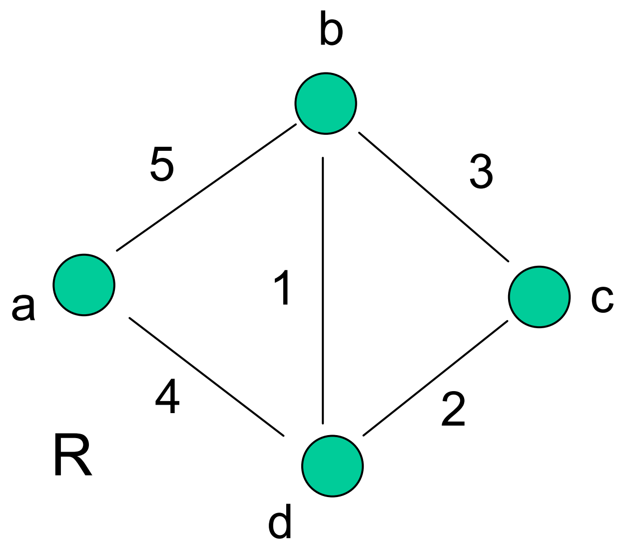


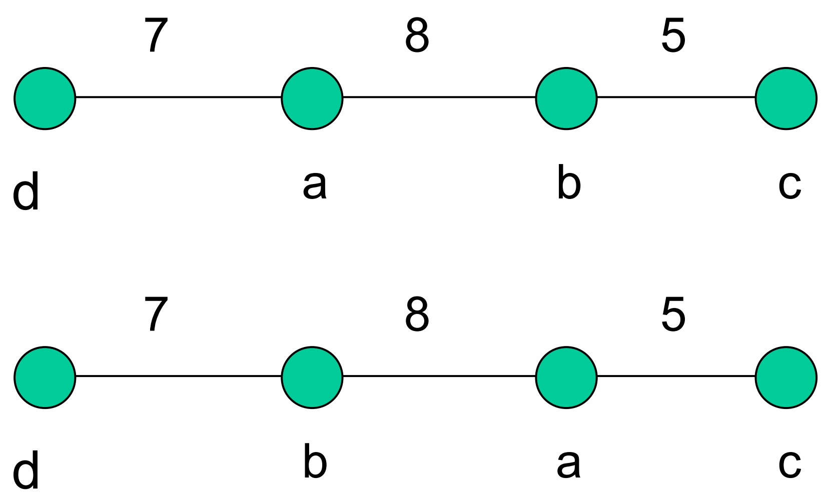
Hallar un flujo de valor máximo entre todos los pares de vértices de la red
Hallar el corte de capacidad mínima entre todos los pares de vértices de la red

¡ Sólo se necesitan n – 1 cálculos !

Redes equivalentes en flujos

Las redes R = (V, A, c) y R' = (V, A', c') son equivalentes en flujos si para cada par de vértices u, v de V, la mínima capacidad de un corte u-v (flujo máximo de u a v) en R es la misma que en R'

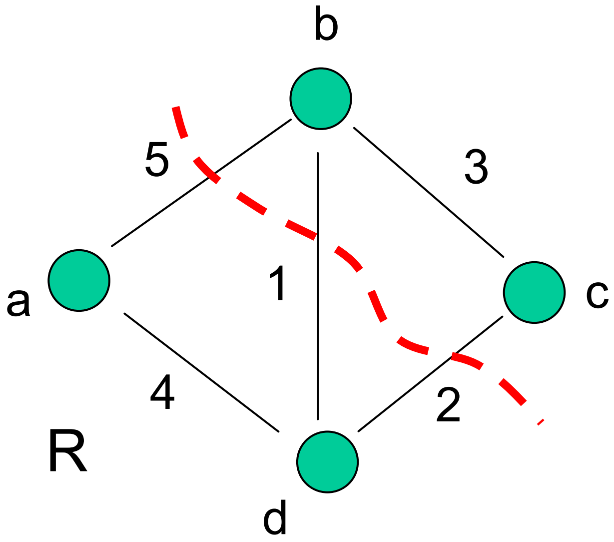




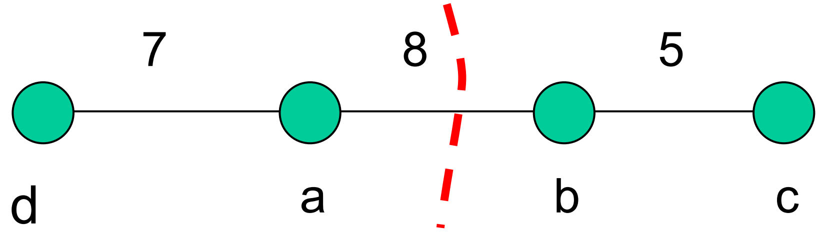
Dos árboles equivalentes en flujos a R

Redes equivalentes en flujos

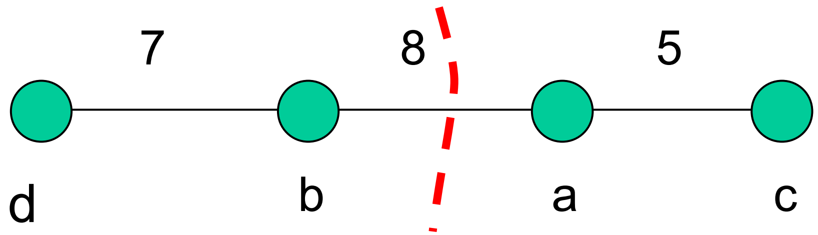
Las redes R = (V, A, c) y R’ = (V, A’, c’) son equivalentes en flujos si para cada par de vértices u, v de V, la mínima capacidad de un corte u-v (flujo máximo de u a v) en R es la misma que en R’



Corte mínimo a-b
separa V en {a,d} y {b,c}



Corte mínimo a-b separa V en dos partes {a,d} y {b,c}



Corte mínimo a-b separa V en dos partes {b,d} y {a,c}

¡No coincide el corte en R y en el árbol!

ÁRBOLES CORTE (cut-tree)

Dada una red R = (V, A, c), un **árbol-corte** T = (V, F) es un árbol construido a partir de R con una función capacidad c’ verificando

1. Equivalencia de flujos

Para cada par de vértices s, t en V el valor máximo de un flujo entre s y t en R coincide con dicho valor en T
(Y pasando a capacidad mínima de un corte, esto significa que ese valor es la menor capacidad de las aristas en el único camino entre s y t en T)
(Los valores numéricos de las capacidades mínimas coinciden)

2. Propiedad de corte

Cada corte de capacidad mínima en T también lo es en R
(Coinciden también las particiones definidas por los cortes mínimos)

ÁRBOLES GOMORY-HU

Dada una red R siempre existe un árbol T que es equivalente en flujo a R y que, además, es un árbol-corte (cut-tree)

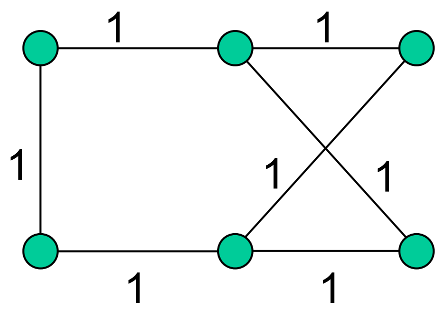
GOMORY-HU Tree

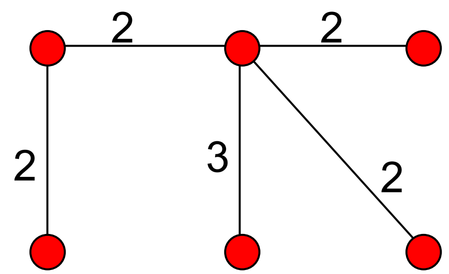
Las n – 1 aristas del árbol corresponden a los n – 1 cortes de capacidad mínima en R, es decir, sólo hay O(n) cortes de capacidad mínima en una red R

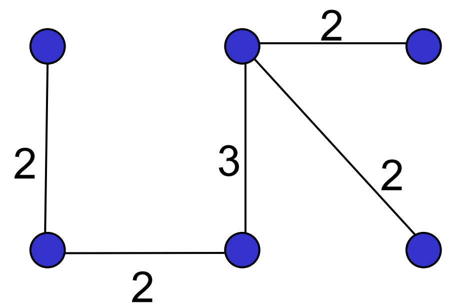
ÁRBOLES GOMORY-HU

Dada una red R siempre existe un árbol T que es equivalente en flujo a R y que, además, es un árbol-corte (cut-tree)

Una red puede tener más de un árbol-corte







ÁRBOLES GOMORY-HU

Algoritmo de construcción

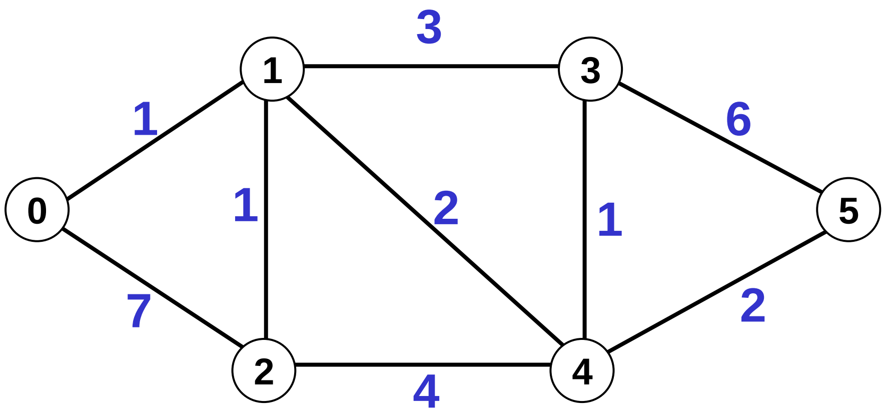
Ideas clave:

1. Mantener una partición de V, $(S_1, S_2, \dots, S_k)$, y un árbol generador T sobre el conjunto de vértices $\{S_1, S_2, \dots, S_k\}$

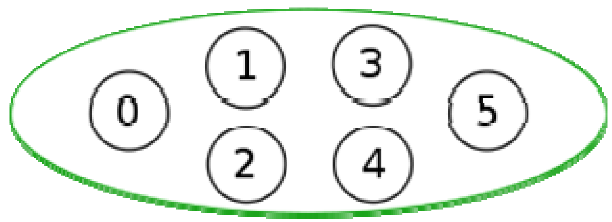
2. Asignar pesos c' a las aristas de T de forma que en cada iteración se satisfaga el siguiente invariante:
Para cada arista (S_i, S_j) de T hay vértices $a \in S_i$, $b \in S_j$
tales que $c'(S_i, S_j) = f(a,b)$ y el corte definido por (S_i, S_j) es un corte de capacidad mínima en R entre a y b

ÁRBOLES GOMORY-HU

Algoritmo de construcción Ejemplo



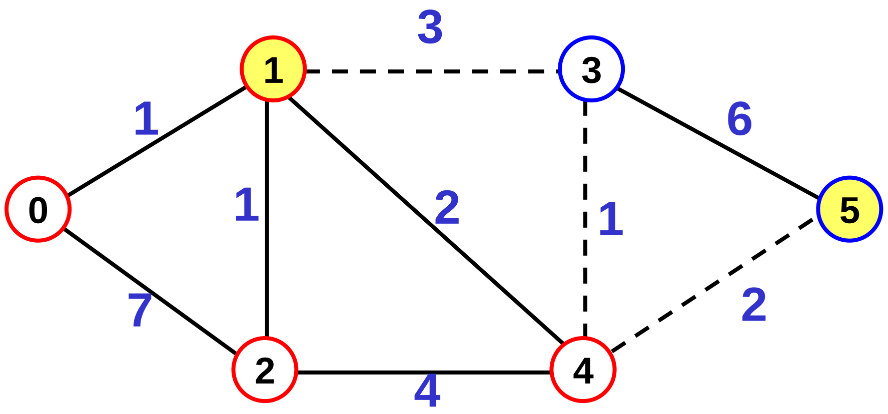
T

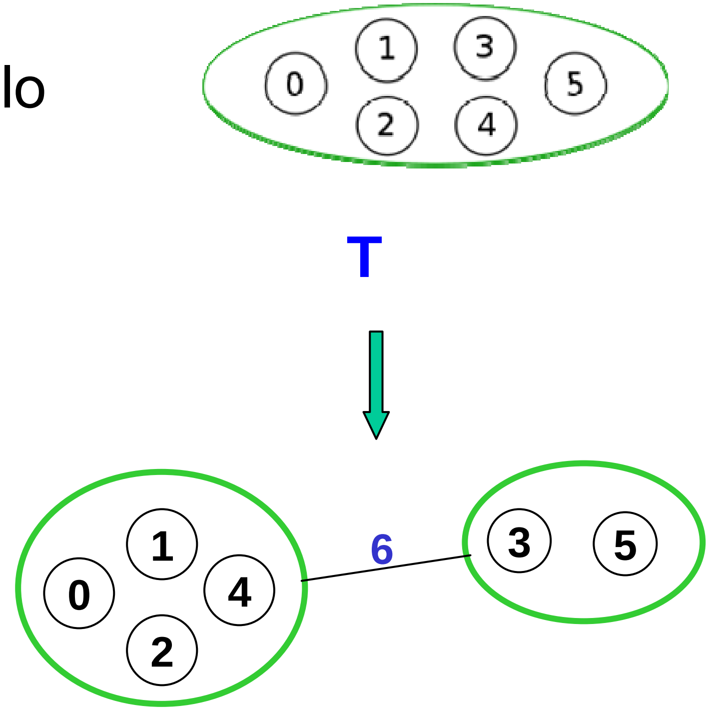


Elegimos dos vértices 1 y 5 en un vértice de T,
calculamos el corte de capacidad mínima entre 1 y 5

ÁRBOLES GOMORY-HU

Algoritmo de construcción Ejemplo



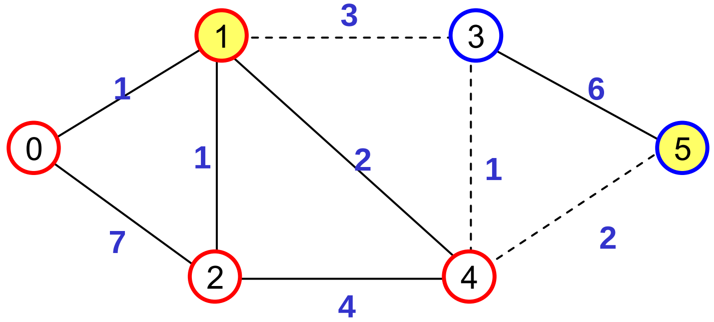


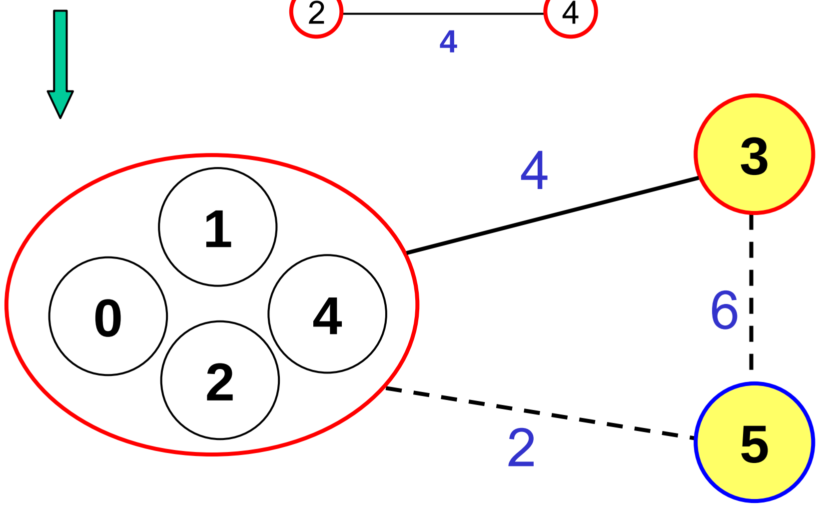
Elegimos dos vértices 1 y 5 en un vértice de T,
calculamos el corte de capacidad mínima entre 1 y 5
cap(1,5) = 6

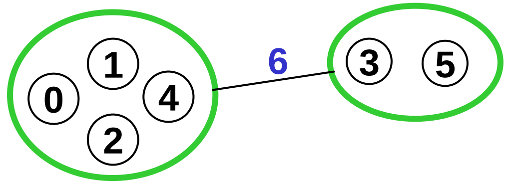
Rojo-Azul el corte 1-5

ÁRBOLES GOMORY-HU

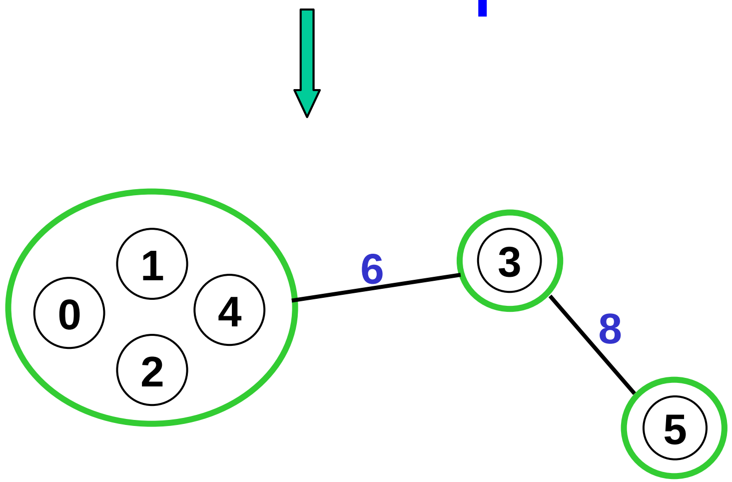
Algoritmo de construcción Ejemplo







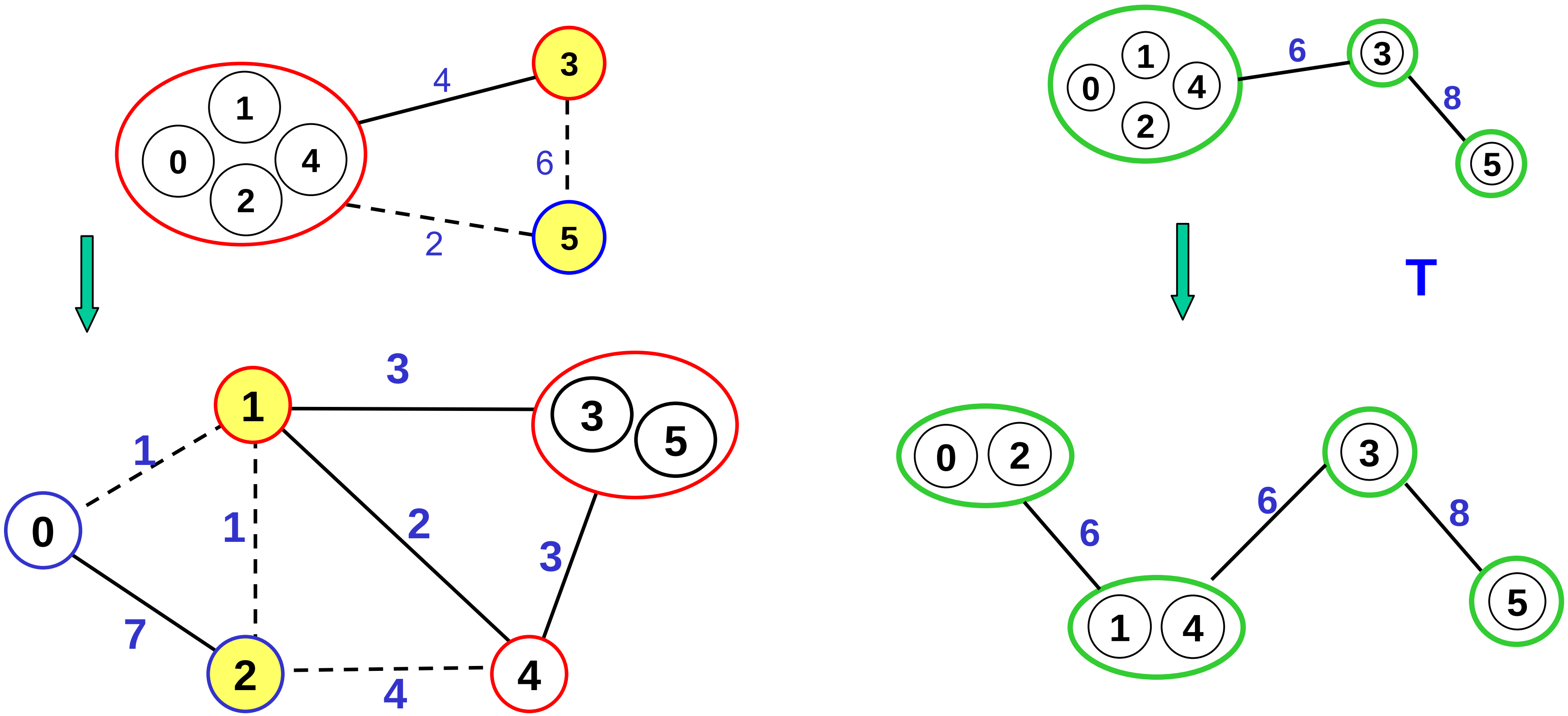
T



Elegimos dos vértices 3 y 5 en un vértice de T, calculamos el corte de capacidad mínima entre 3 y 5 cap(3,5) = 8

ÁRBOLES GOMORY-HU

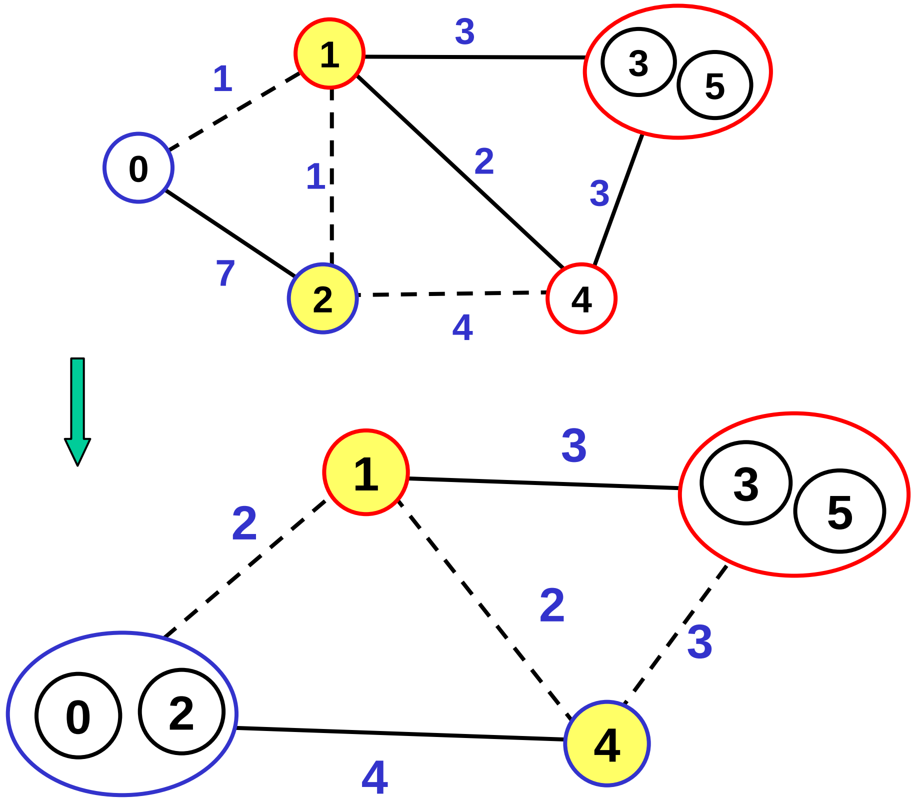
Algoritmo de construcción Ejemplo

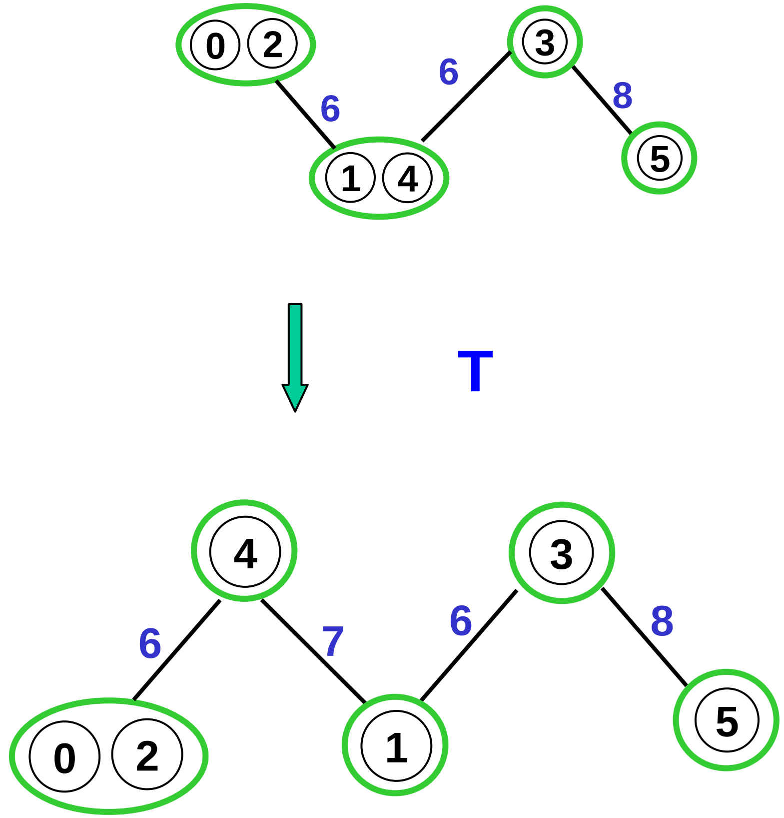


Elegimos dos vértices 1 y 2 en un vértice de T,
calculamos el corte de capacidad mínima entre 1 y 2
cap(1,2) = 6

ÁRBOLES GOMORY-HU

Algoritmo de construcción Ejemplo

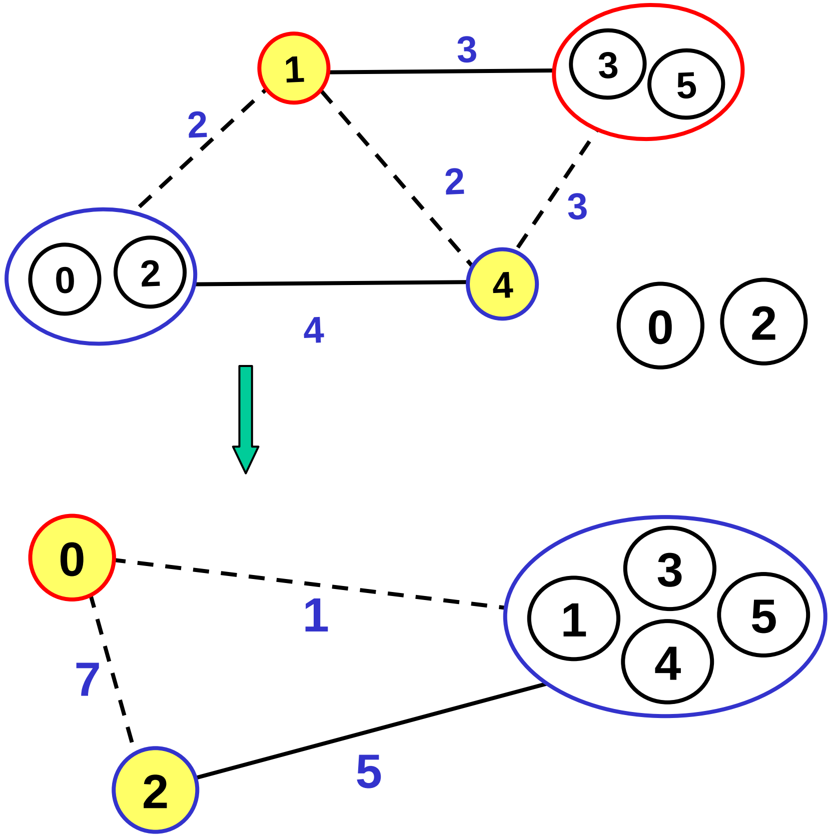


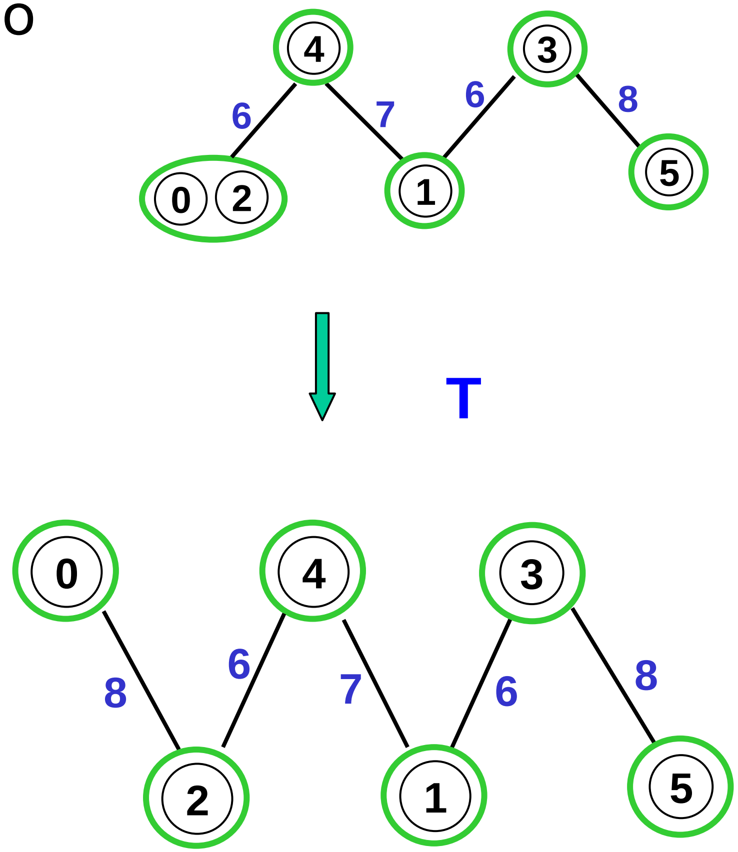


Elegimos dos vértices 1 y 4 en un vértice de T, calculamos el corte de capacidad mínima entre 1 y 4 cap(1,4) = 7

ÁRBOLES GOMORY-HU

Algoritmo de construcción Ejemplo

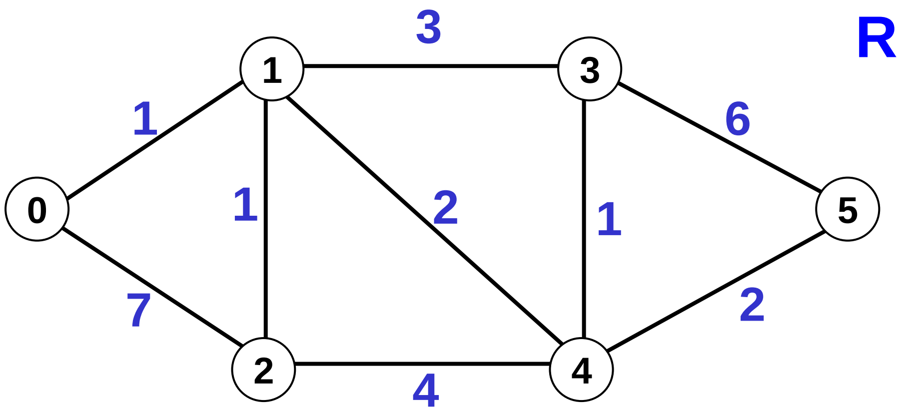


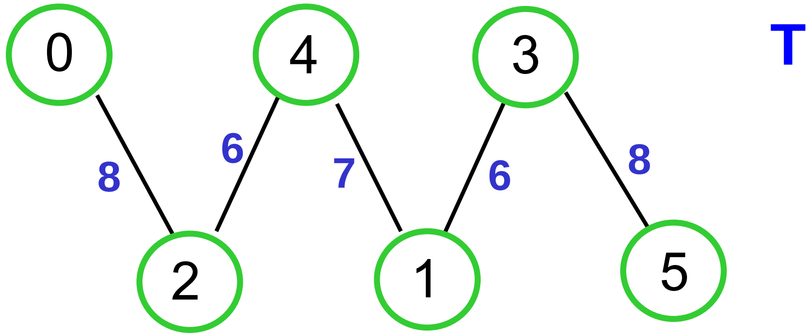


Elegimos dos vértices 0 y 2 en un vértice de T,
calculamos el corte de capacidad mínima entre 0 y 2
cap(0,2) = 8

ÁRBOLES GOMORY-HU

Algoritmo de construcción Ejemplo

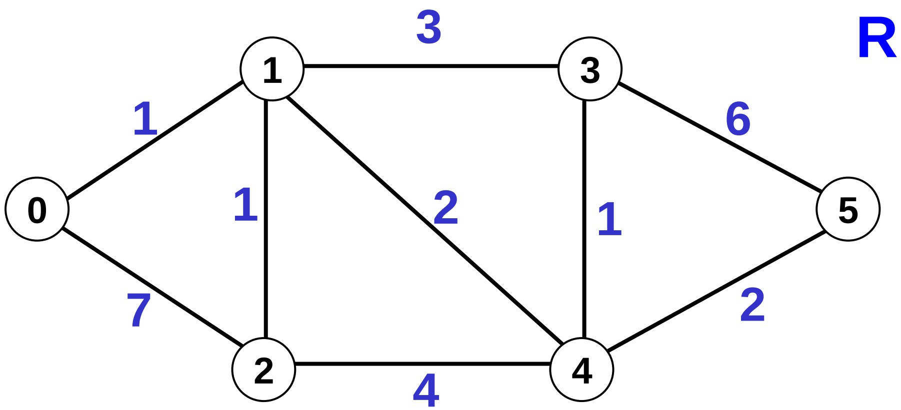




Si todos los pesos en T son distintos el corte de capacidad mínima en R entre cada par de vértices es único

ÁRBOLES GOMORY-HU

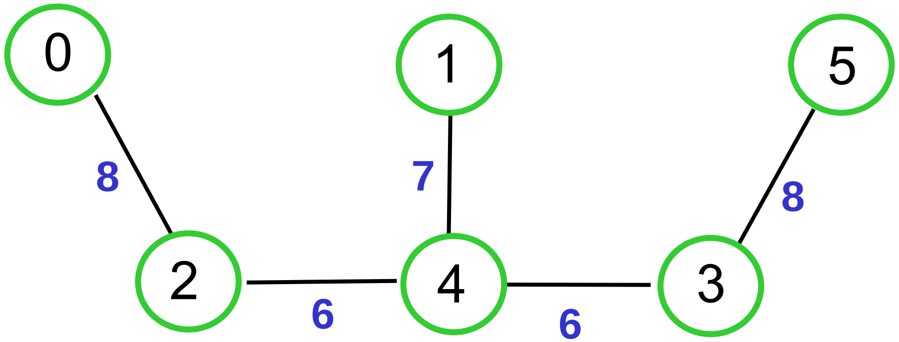
Algoritmo de construcción Ejemplo



R

Si se hubieran elegido otros vértices a separar en cada paso, el árbol obtenido puede ser distinto.

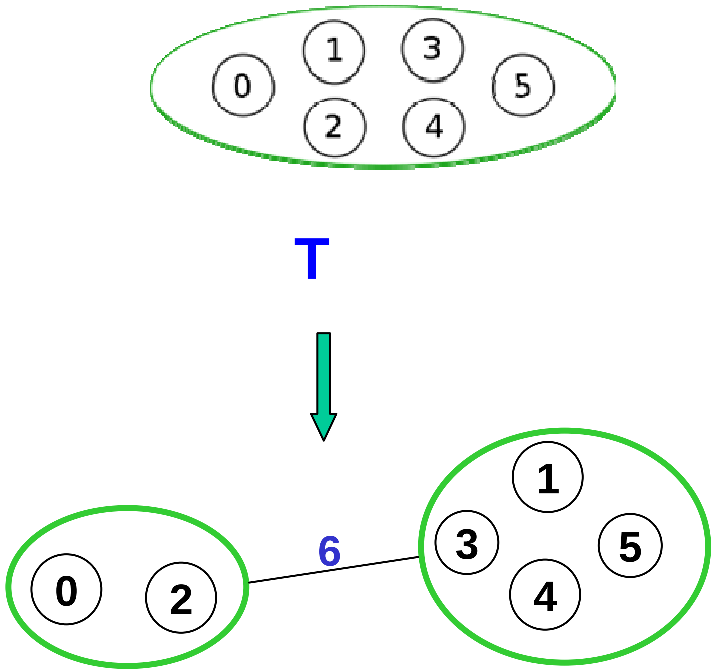
Eligiendo 0-4, 0-2, 1-4, 3-4 y 3-5 en este orden, resulta el árbol T*

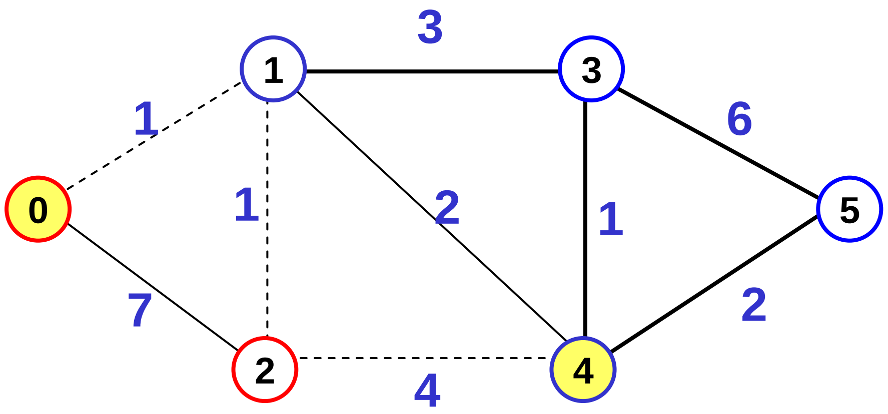


T*

ÁRBOLES GOMORY-HU

Algoritmo de construcción Ejemplo



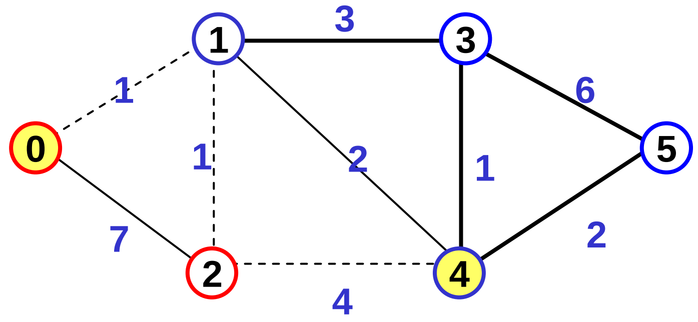


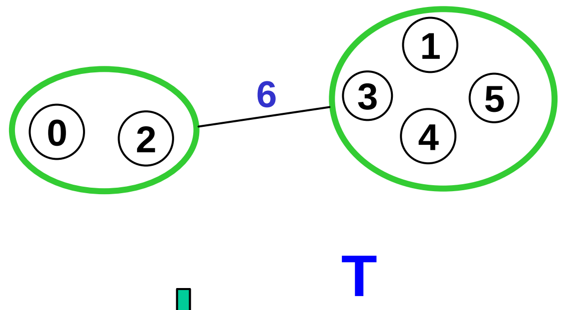
Elegimos dos vértices 0 y 4 en un vértice de T,
calculamos el corte de capacidad mínima entre 0 y 4
cap(0,4) = 6

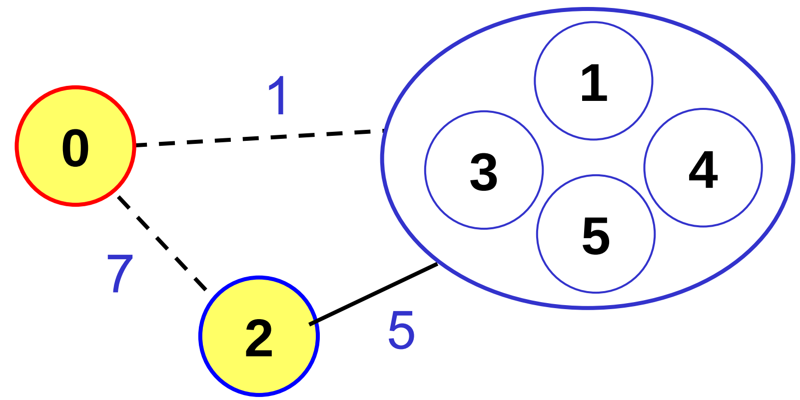
Rojo-Azul el corte 1-5

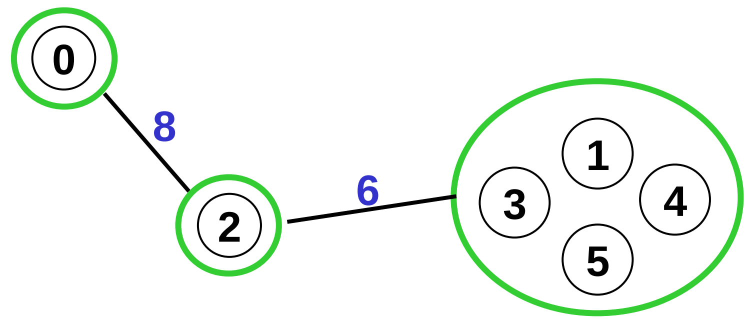
ÁRBOLES GOMORY-HU

Algoritmo de construcción   Ejemplo





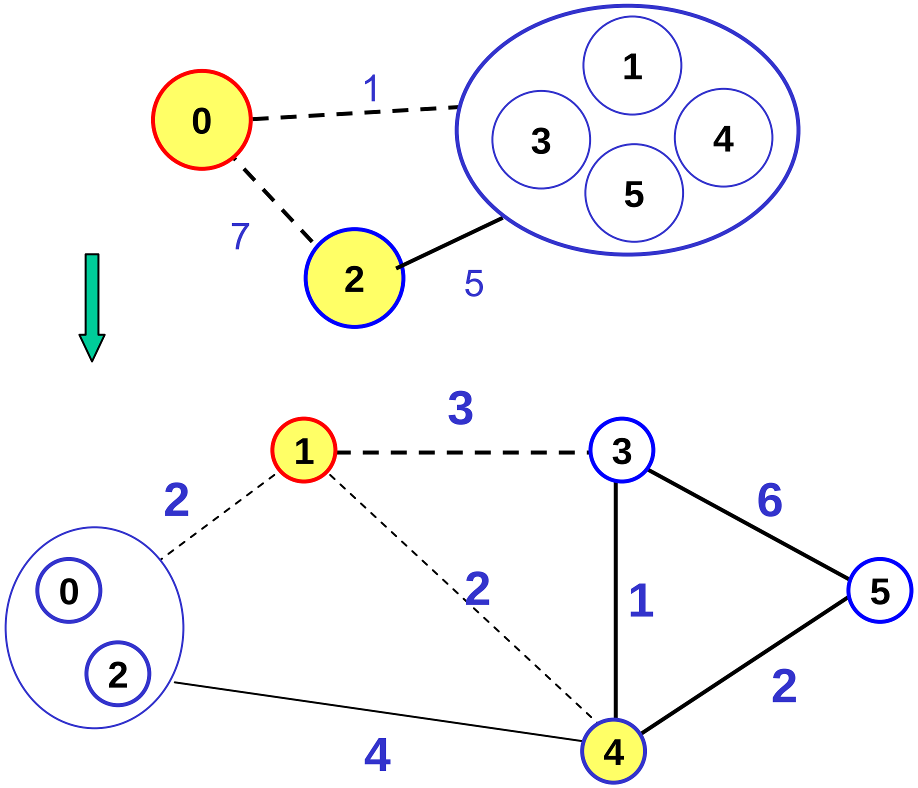


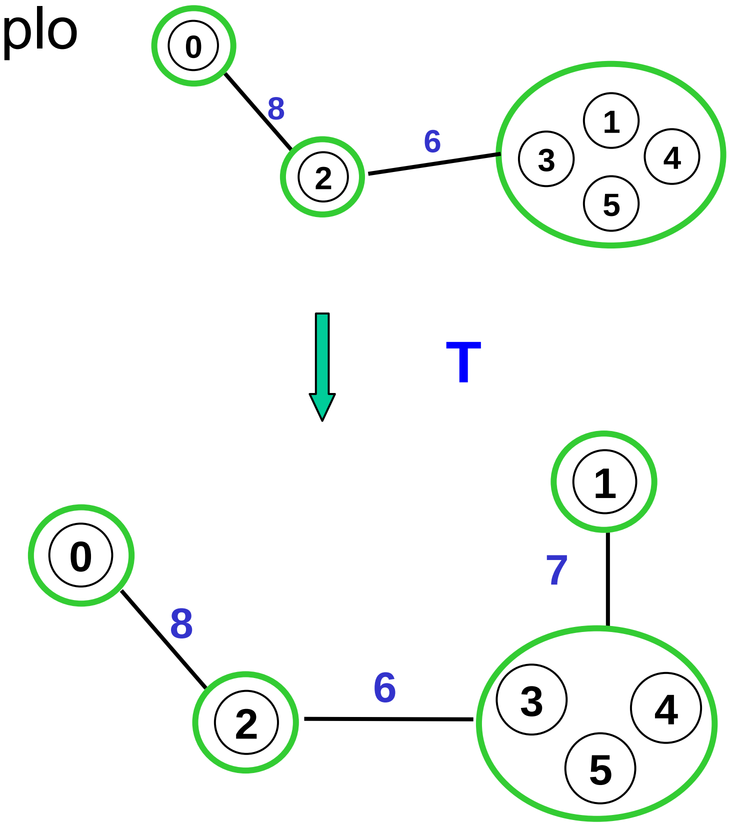


Elegimos dos vértices 0 y 2 en un vértice de T,
calculamos el corte de capacidad mínima entre 0 y 2
cap(0,2) = 8

ÁRBOLES GOMORY-HU

Algoritmo de construcción Ejemplo

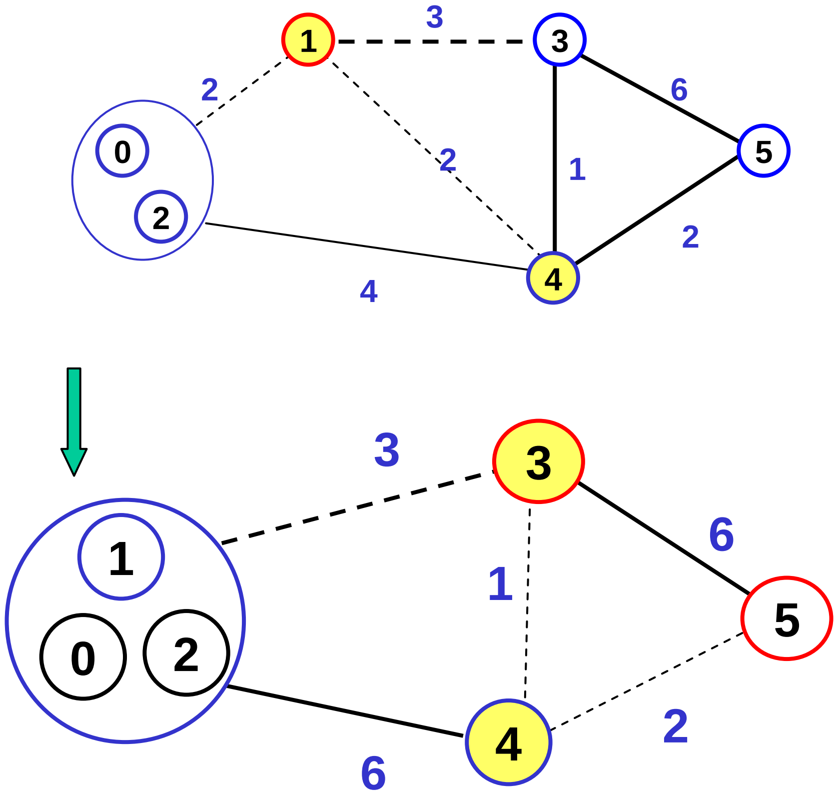


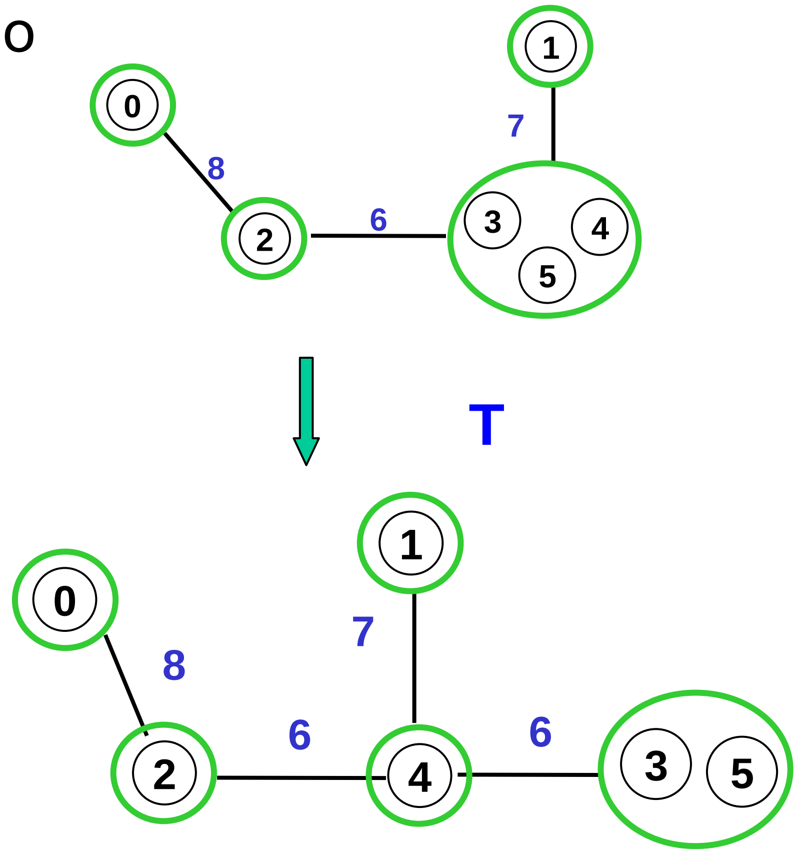


Elegimos dos vértices 1 y 4 en un vértice de T, calculamos un corte de capacidad mínima entre 1 y 4 cap(1,4) = 7

ÁRBOLES GOMORY-HU

Algoritmo de construcción   Ejemplo

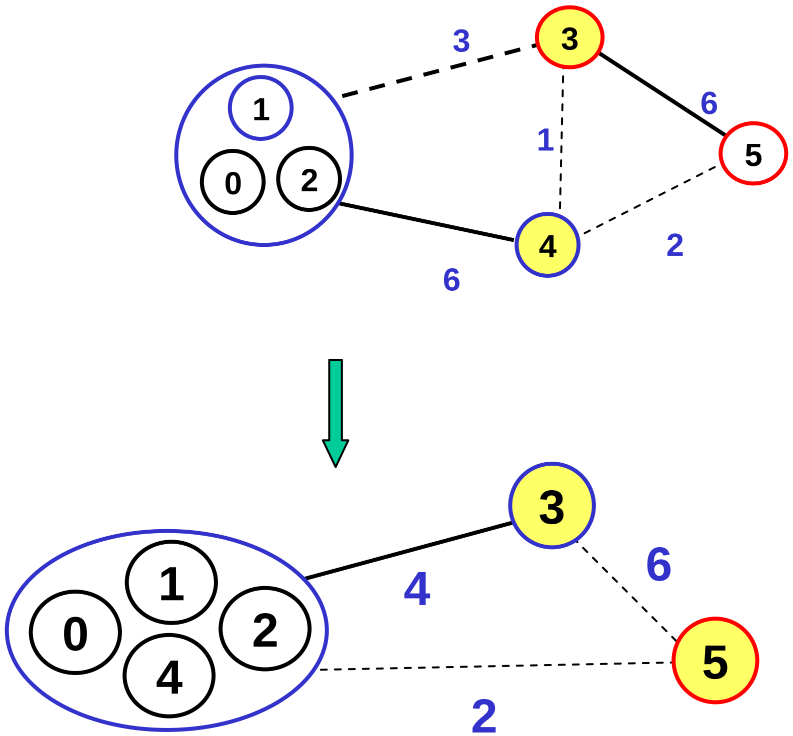


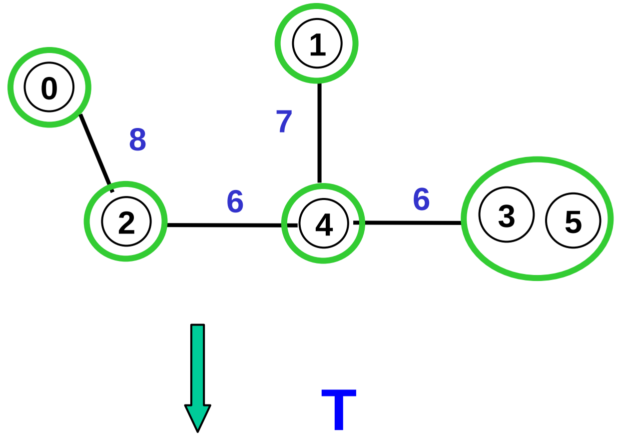


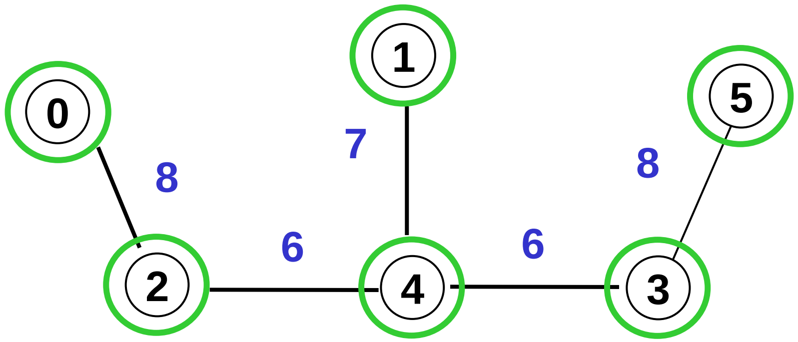
Elegimos dos vértices 3 y 4 en un vértice de T,
calculamos el corte de capacidad mínima entre 3 y 4
cap(3,4) = 6

ÁRBOLES GOMORY-HU

Algoritmo de construcción   Ejemplo

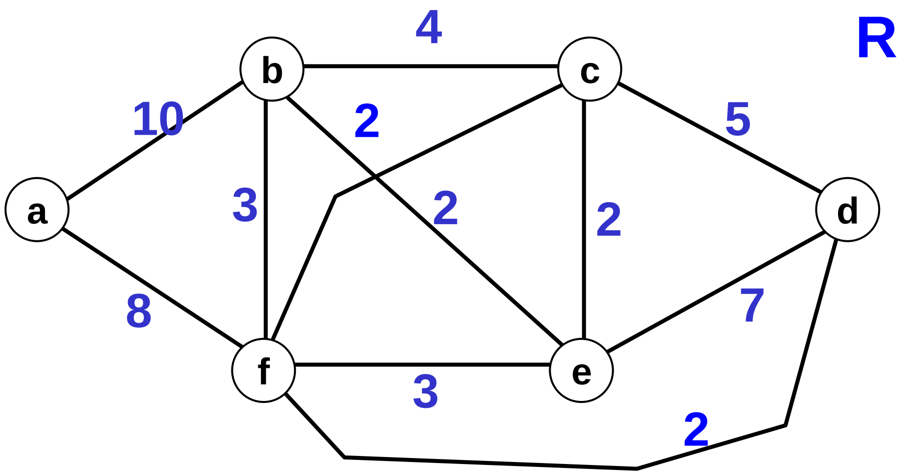


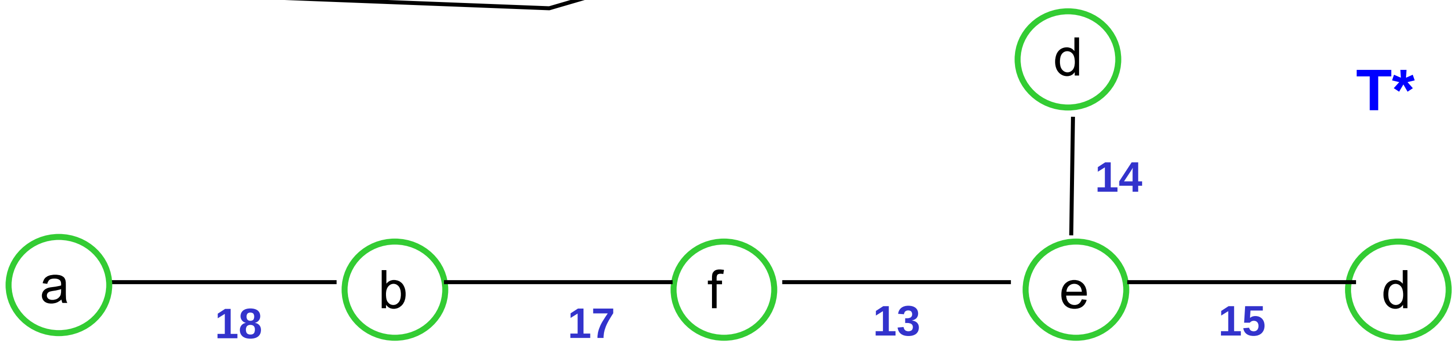




Elegimos dos vértices 3 y 5 en un vértice de T,
calculamos el corte de capacidad mínima entre 3 y 5
cap(3,5) = 8

ÁRBOLES GOMORY-HU

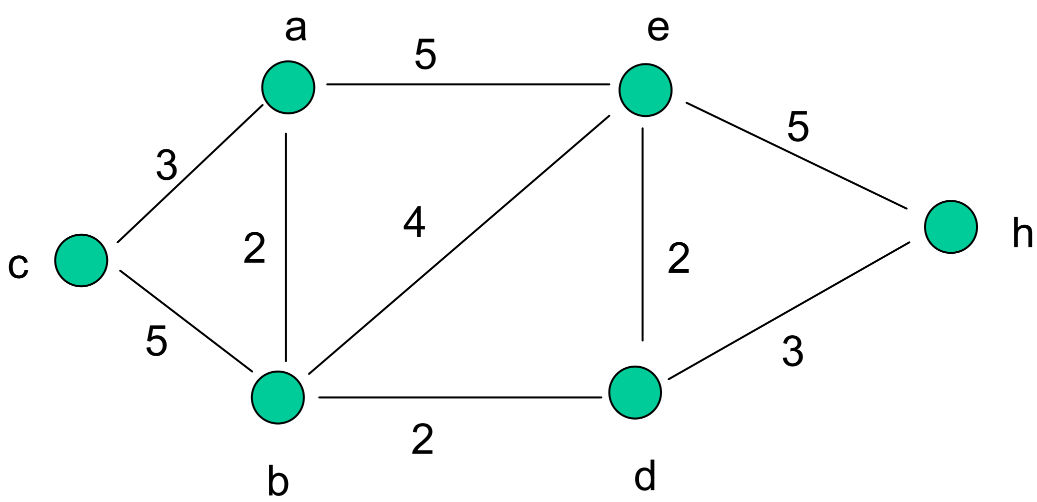


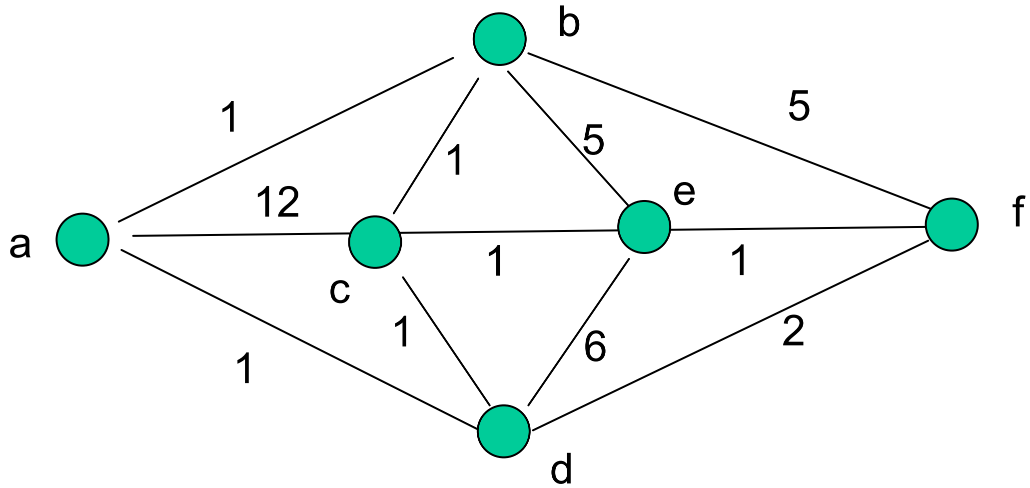


ÁRBOLES GOMORY-HU

Ejemplos

R





ÁRBOLES GOMORY-HU

Demostración de la corrección del algoritmo

Teorema

Para todo par de vértices a,b de R, la mínima capacidad de un corte a-b es la mínima etiqueta de las aristas en el camino entre a y b en el árbol T.

Además si e* es dicha arista entonces el corte a-b de capacidad mínima se alcanza en la bipartición de V correspondiente a los dos árboles de T – e*

ÁRBOLES GOMORY-HU

Demostración de la corrección del algoritmo

Lema 1.

Si u, v, w son vértices de una red R y f(u,v) es la capacidad mínima de un corte u-v en R entonces

$$f(u, v) \geq \min \{f(u, w), f(w, v)\}$$

Lema 2. Desigualdad submodular

Si $R = (V, E, c)$ es una red con $c(e) \geq 0$ para toda arista e, A, B $\subset V$ y designamos con cap(A) a la capacidad del corte (A, V – A) entonces

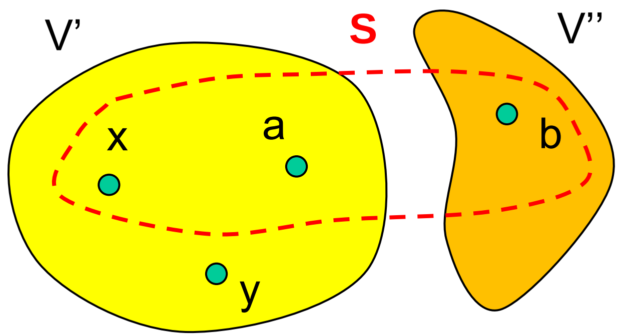
$$cap(A) + cap(B) \geq cap(A \cup B) + cap(A \cap B)$$

ÁRBOLES GOMORY-HU

Demostración

Lema 3. (La contracción de vértices NO afecta a los cortes de los vértices no contraídos)
Sean a, b vértices de R, (V’,V’’) un corte de capacidad mínima entre a y b. Sean x, y vértices de V’. Entonces existe un corte x-y de capacidad mínima en V’ (es decir la contracción de V’’ a un único vértice no afecta a los cortes de V’)

Dem.:
Sea (S,S’) corte mínimo x-y.
Debemos probar que $S \subset V'$
Procedemos por reducción al absurdo,
suponemos que $S \not\subset V'$ y que $b \in S$



Caso 1) $a \in S$ Aplicamos el lema 2 a los cortes V’ y V – S

$$cap(V') + cap(V - S) \geq cap(V' \cap (V - S)) + cap(V' \cup (V - S))$$

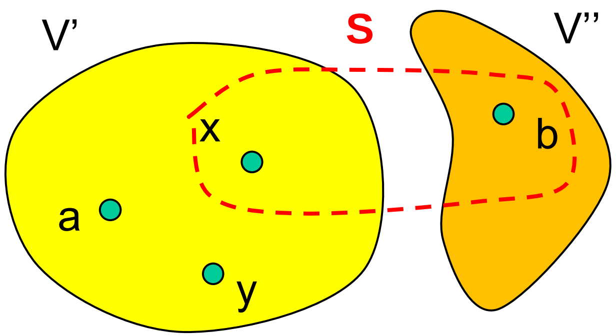
Como $V' \cup (V - S)$ es un a-b corte, $cap(V' \cup (V - S)) \geq cap(V')$
luego, $cap(V' \cap (V - S)) \leq cap(V - S) = cap(S)$ y
$V' \cap (V - S)$ es corte de capacidad mínima dentro de V’

ÁRBOLES GOMORY-HU

Demostración

Lema 3. (La contracción de vértices NO afecta a los cortes de los vértices no contraídos)
Sean a, b vértices de R, (V’,V’’) un corte de capacidad mínima entre a y b. Sean x, y vértices de V’. Entonces existe un corte x-y de capacidad mínima en V’ (es decir la contracción de V’’ a un único vértice no afecta a los cortes de V’)

Dem.:
Sea (S,S’) corte mínimo x-y.
Debemos probar que $S \subset V'$
Procedemos por reducción al absurdo,
suponemos que $S \not\subset V'$ y que $b \in S$



Caso 2) $a \notin S$

Análogamente se demuestra que $V' \cap S$ es un corte x-y de capacidad mínima

ÁRBOLES GOMORY-HU Demostración

Vértices representativos

Si X e Y son nodos adyacentes del árbol con e = $\widehat{XY}$, se dice que $x \in X$, $y \in Y$, son vértices representativos si la etiqueta (peso) de e coincide con la capacidad del corte mínimo entre x e y, es decir, $f(x,y) = c'(e)$

Lema 4.

En cada etapa del algoritmo cada una de las aristas del árbol T tiene vértices representativos.

Dem.:

Es claramente cierto en el paso inicial en el que el árbol T tiene dos nodos.

Comprobemos que la propiedad se mantiene en el paso general:

El nodo A se descompone en dos nodos X e Y, a partir de un (x,y)-corte mínimo con respecto a vértices terminales $x \in X$, $y \in Y$.

Aplicando reiteradamente el lema 3, la capacidad del (x,y)-corte en R_A es $f(x,y)$

Así x,y son representativos de la arista e = XY en el árbol T

¿Qué pasa con las restantes aristas de T?

ÁRBOLES GOMORY-HU

Demostración

Lema 4.

En cada etapa del algoritmo cada una de las aristas del árbol T tiene vértices representativos.

Dem. (continuación): ¿Qué pasa con las restantes aristas de T?
Hay que controlar las que antes llegaban al nodo A. Sea h la arista AB

Existen vértices $a \in A$ y $b \in B$ tales que $f(a,b) = c'(h)$
Si $a \in X$, entonces a y b siguen siendo representativos de la arista h

Si $a \notin X$, es decir, $a \in Y$, veremos que x y b son representativos de h en el nuevo árbol

El corte que se utilizó para crear h en T, separa x de b, luego $f(x,b) \leq f(a,b)$
Necesitamos la igualdad para la representatividad

ÁRBOLES GOMORY-HU

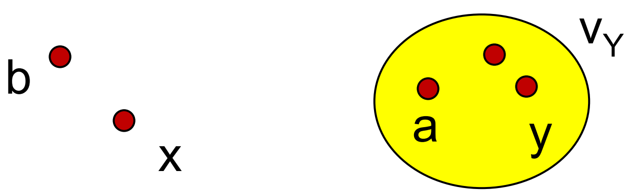
Demostración

Lema 4.

En cada etapa del algoritmo cada una de las aristas del árbol T tiene vértices representativos.

Dem. (continuación): ¿Qué pasa con las restantes aristas de T?

Identificamos todos los vértices de Y en R en un vértice v_Y y obtenemos el grafo R'
Llamamos f '(u,w) a la mínima capacidad de un corte u-w en R'



Por el lema 3, f(x, b) = f '(x, b)

Y por el lema 1, $f'(x,b) \geq \min\{f'(x,v_Y), f'(v_Y, b)\}$
Como $a \in v_Y$ entonces , $f'(v_Y, b) \geq f(a, b)$
Además $f'(x, v_Y) \geq f(x, y) \geq f(a, b)$
(porque el xy-corte que descompone A también separa a de B)

Por tanto, $f'(x, b) \geq f(a, b)$ y $f(x, b) \geq f(a, b)$
Luego f(x, b) = f(a, b) Así x y b son representativos para la arista h

ÁRBOLES GOMORY-HU Demostración

Teorema

Para todo par de vértices a, b de R, la mínima capacidad de un ab-corte es la mínima etiqueta de las aristas en el camino entre a y b en el árbol T. Además si e* es dicha arista entonces el ab-corte de capacidad mínima se alcanza en la bipartición de V correspondiente a los dos árboles de T – e*

Dem.: La segunda afirmación es consecuencia de la primera y de la construcción del árbol. Así cada arista de T corresponde a un corte en R especificado por la partición definida en T – e y de capacidad c'(e)

Sea $V_0, e_1, V_1, e_2, \ldots, e_k, V_k$ el camino en T que une a y b.
Debemos demostrar que $f(a, b) = \min \{c'(e_1), c'(e_2), \ldots, c'(e_k)\}$

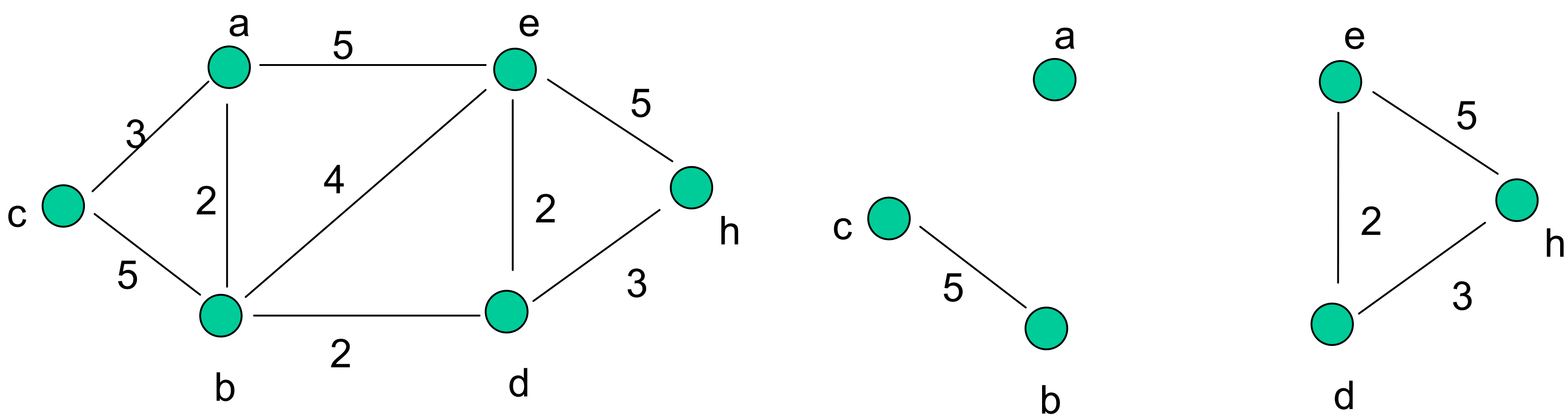
Es claro que $f(a, b) \leq \min \{\ldots\}$ porque cada corte de las aristas e_i separa a de b

La otra desigualdad se obtiene aplicando el lema 1 (extendido),
porque llamando v_i al único vértice en R del nodo V_i resulta que $c'(e_i) = f(v_{i-1}, v_i)$
y así $f(a, b) \geq \min \{c'(e_1), c'(e_2), \ldots, c'(e_k)\}$

MINIMUM k-CUT Problem

Dado un grafo G = (V, A) un k-corte es un conjunto de aristas F tal que G – F tiene k componentes conexas

El MINIMUM k-CUT Problem pide un k-corte de peso mínimo



F = {ac, ab, ae ,be, bd} es un 3-corte de peso 16

MINIMUM k-CUT Problem

Dado un grafo G = (V, A) un k-corte es un conjunto de aristas F tal que G – F tiene k componentes conexas

El MINIMUM k-CUT Problem pide un k-corte de peso mínimo

Si k es fijo el problema se puede resolver en tiempo $O(n^{k^2})$

Si k forma parte de los datos de entrada el problema es NP-completo

Algoritmo aproximado utilizando árboles de Gomory-Hu

MINIMUM k-CUT Problem

Algoritmo aproximado

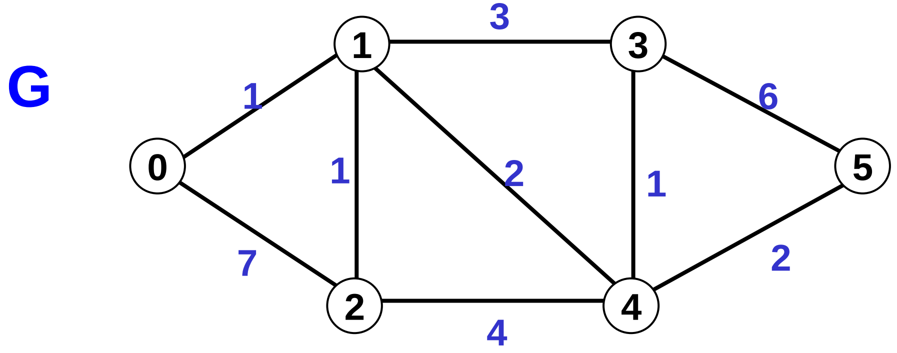
Paso 1. Calcular el árbol T de Gomory-Hu. Cada arista de T origina un corte en el grafo inicial G. Tenemos $n - 1$ cortes

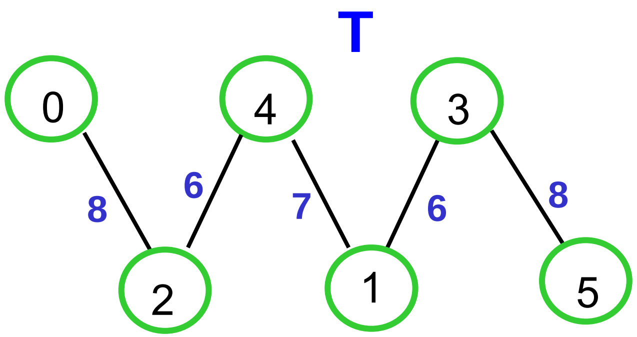
Paso 2. Consideramos los $k - 1$ cortes más ligeros. La unión de de las aristas correspondientes en G es un k-corte F

Este algoritmo es $(2 - 2/k)$-aproximado

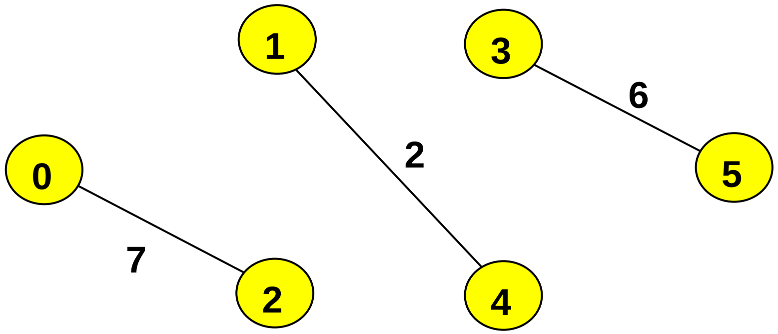
MINIMUM k-CUT Problem

Algoritmo 2-aproximado Ejemplo





Las dos aristas más ligeras en T son 24 y 13. Eliminando las aristas de G correspondientes a los cortes 2-4 y 1-3 tenemos un 3-corte F de peso 12



En este ejemplo F es el 3-corte de peso mínimo

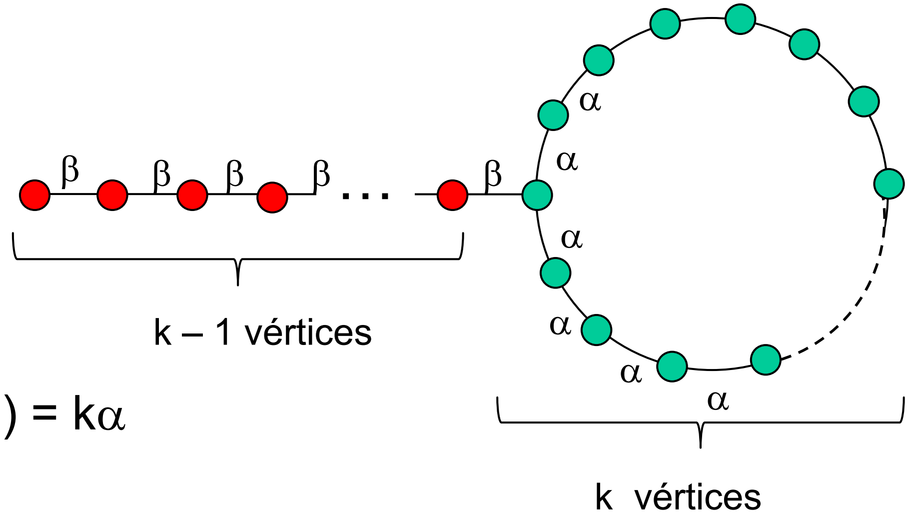
MINIMUM k-CUT Problem

Algoritmo 2-aproximado Ejemplo “malo”

G

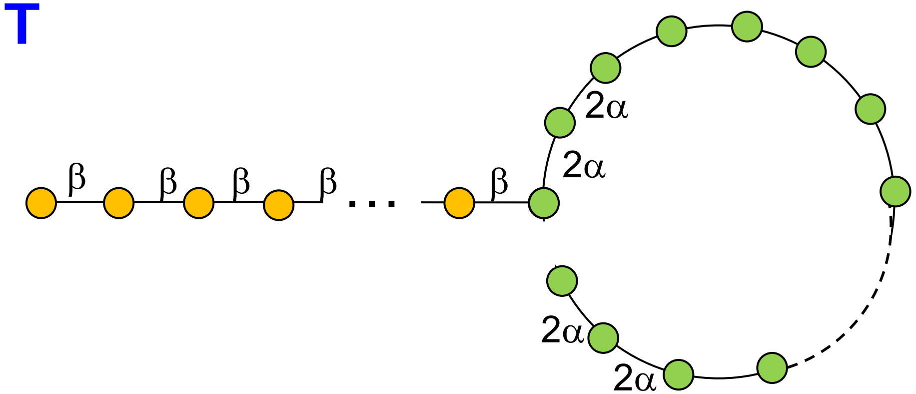
Grafo con 2k – 1 vértices

$\beta = 2\alpha(1 - \varepsilon)$



k-corte óptimo F*, de peso $w(F^*) = k\alpha$

T



El algoritmo elige un k-corte F formado por las aristas de peso β

$w(F) = (k - 1)\beta$

$$\text{Razón} = \frac{w(F)}{w(F^*)} = \frac{(k-1)\beta}{k\alpha} = (2 - \frac{2}{k})(1-\varepsilon)$$