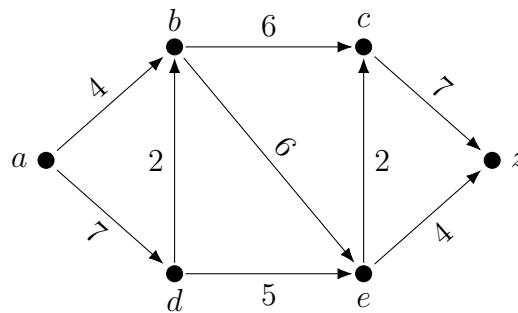


## Flujo en Redes

### Redes de transporte

Supongamos que el grafo dirigido de la siguiente figura representa la red de tuberías de un gasoducto. El yacimiento donde el gas es extraído está representado por el vértice  $a$ , y se desea transportar el gas hacia una terminal distribuidora, representada por el vértice  $z$ . Los demás vértices del grafo representan puntos de distribución intermedios. Las aristas dirigidas representan las tuberías de la red, y muestran en qué dirección puede circular el gas. Finalmente, las etiquetas de las aristas muestran la capacidad, en alguna unidad pertinente por unidad de tiempo, de la tubería. Se desea encontrar una manera de maximizar la cantidad de gas transportado por la red desde el yacimiento  $a$  a la terminal  $z$ .



Recordemos que dado un grafo simple dirigido  $G$  y  $v \in V(G)$ , denotamos por  $d^+(v)$  al grado de salida de  $v$ , i.e.,  $|\{x : vx \in E(G)\}|$ , y  $d^-(v)$  al grado de entrada de  $v$ , i.e.,  $|\{x : xv \in E(G)\}|$ .

**Definición 1.** Una *red* es un grafo dirigido simple ponderado que tiene un vértice distinguido  $a$  con  $d^-(a) = 0$  que llamaremos *origen* o *fuente*, y un vértice distinguido  $z$  con  $d^+(z) = 0$ , que llamaremos *destino* o *sumidero*, y tal que los pesos de las aristas son no negativos. Al peso de una arista  $e = uv$  lo llamaremos *capacidad* de la arista  $e$ , y lo notaremos por  $c(e)$  o  $c(u, v)$ .

**Ejemplo.** El grafo dirigido al inicio de este apunte, que modelaba la red de tuberías de un gasoducto, es una red. El vértice  $a$  es la fuente y el vértice  $z$  el sumidero. La etiqueta de cada arista es la capacidad de la misma.

**Definición 2.** Dada una red  $G$ , un *flujo* en  $G$  es una función  $f : E(G) \rightarrow \mathbb{R}_0^+$ . Dado un flujo  $f$  en una red  $G$  y  $v \in V(G)$ , denotaremos por  $f^-(v)$  y  $f^+(v)$  al flujo entrante a  $v$  y saliente desde  $v$ , respectivamente. Es decir,

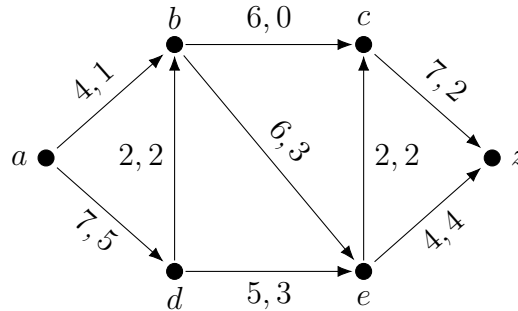
$$f^-(v) = \sum_{u \in V(G)} f(u, v), \quad f^+(v) = \sum_{u \in V(G)} f(v, u)$$

donde  $f(x, y) = 0$  si  $xy \notin E(G)$ .

Diremos que  $f$  es un flujo *factible* si para cada arista  $e$  se verifica  $0 \leq f(e) \leq c(e)$  y para cada vértice  $v \in V(G) \setminus \{a, z\}$  se verifica  $f^+(v) = f^-(v)$ .

**Ejemplo.** En la red del ejemplo anterior, el flujo  $f$  dado por

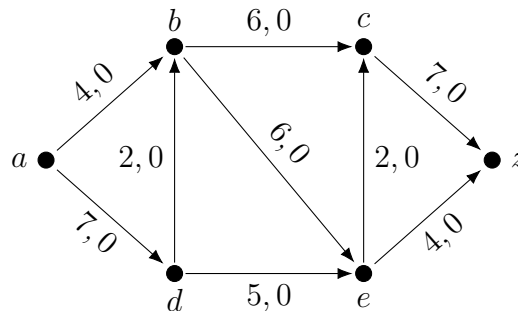
$f(bc) = 0$ ,  $f(ab) = 1$ ,  $f(bd) = f(ec) = f(cz) = 2$ ,  $f(be) = f(de) = 3$ ,  $f(ez) = 4$ ,  $f(ad) = 5$  es un flujo factible.



Podemos chequear que para cada arista  $e$ ,  $0 \leq f(e) \leq c(e)$ , y para cada vértice  $v$  el flujo entrante es igual al flujo saliente. Por ejemplo, para el vértice  $d$  tenemos

$$f^-(d) = \sum_{u \in V(G)} f(u, d) = f(a, d) = 5, \quad f^+(d) = \sum_{u \in V(G)} f(d, u) = f(d, b) + f(d, e) = 2 + 3 = 5$$

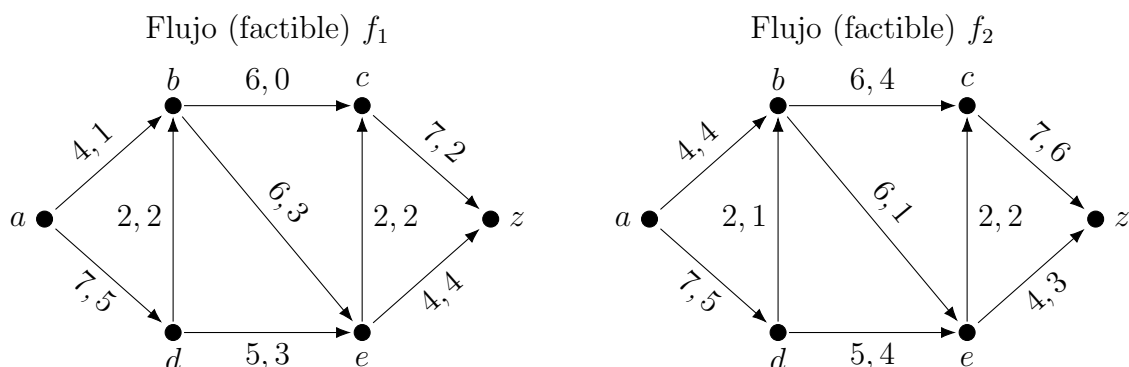
Observemos que en toda red el flujo nulo  $f$ , i.e.  $f(e) = 0$  para toda  $e \in E(G)$ , es un flujo factible.



**Observación 3.** Dado un flujo factible  $f$  en una red  $G$ , sabemos que para todo vértice  $v \in V(G)$  distinto de la fuente y el sumidero tenemos  $f^+(v) = f^-(v)$ . A esta propiedad la llamamos *conservación del flujo*.

**Definición 4.** Dado un flujo  $f$  en una red  $G$ , definimos el *valor* del flujo  $f$ , que denotaremos  $val(f)$ , al número  $f^+(a)$ . Un *flujo máximo* es un flujo factible de valor máximo.

**Ejemplo.** Consideremos los siguientes dos flujos sobre la red de los ejemplos anteriores.



Tenemos que el valor del flujo  $f_1$  es

$$\text{val}(f_1) = f_1^+(a) = \sum_{u \in V(G)} f_1(a, u) = f_1(a, b) + f_1(a, d) = 1 + 5 = 6,$$

mientras que el valor del flujo  $f_2$  es

$$\text{val}(f_2) = f_2^+(a) = \sum_{u \in V(G)} f_2(a, u) = f_2(a, b) + f_2(a, d) = 4 + 5 = 9.$$

**Proposición 5.** Si  $G$  es una red y  $f$  un flujo factible de  $G$ , entonces  $f^+(a) = f^-(z)$ .

**Demostración.** Sea  $G$  una red y  $f$  un flujo factible de  $G$ . Observemos que

$$\sum_{u \in V(G)} \sum_{v \in V(G)} f(u, v) = \sum_{u \in V(G)} \sum_{v \in V(G)} f(v, u)$$

ya que ambas sumatorias son iguales a  $\sum_{e \in E(G)} f(e)$ . Luego, tenemos

$$\begin{aligned} 0 &= \sum_{u \in V(G)} \sum_{v \in V(G)} f(u, v) - \sum_{u \in V(G)} \sum_{v \in V(G)} f(v, u) = \\ &= \sum_{u \in V(G)} \left( \sum_{v \in V(G)} f(u, v) - \sum_{v \in V(G)} f(v, u) \right) = \sum_{u \in V(G)} (f^-(u) - f^+(u)) \end{aligned}$$

Recordemos que para cada  $u \in V(G) \setminus \{a, b\}$ , por la propiedad de conservación del flujo vale  $f^-(u) = f^+(u)$ . Además  $f^-(a) = 0$  pues  $d^-(a) = 0$ , y  $f^+(z) = 0$  pues  $d^+(z) = 0$ . Luego, obtenemos

$$0 = f^-(z) - f^+(a).$$

O, equivalentemente,  $f^+(a) = f^-(z)$ , que es lo que queríamos probar.  $\square$

**Ejemplo.** En vista de este resultado, el valor de los flujos  $f_1$  y  $f_2$  del ejemplo anterior, los podríamos haber calculado mirando el flujo entrante a  $z$ . Esto es,

$$\text{val}(f_1) = f_1^-(z) = \sum_{u \in V(G)} f_1(u, z) = f_1(c, z) + f_1(e, z) = 2 + 4 = 6,$$

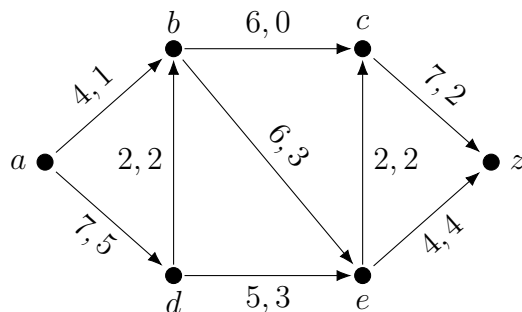
y

$$\text{val}(f_2) = f_2^-(z) = \sum_{u \in V(G)} f_2(u, z) = f_2(c, z) + f_2(e, z) = 6 + 3 = 9.$$

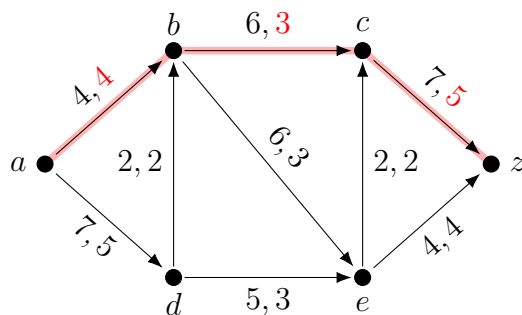
## Algoritmo de flujo máximo

Nos interesa, dada una red, encontrar un flujo máximo. Consideremos nuevamente la red del ejemplo anterior.

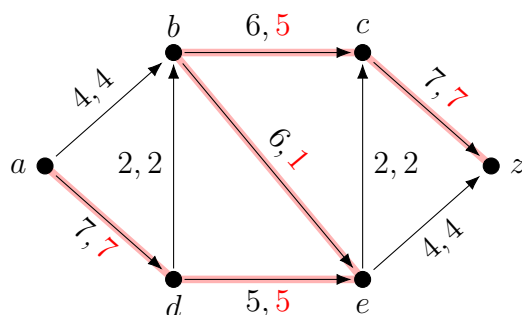
**Ejemplo.** Consideremos nuevamente el flujo (factible) dado por



Observemos el  $a, z$ -camino  $a, b, c, z$ . Notemos que podemos modificar el flujo sobre las aristas de este camino, sumando 3 unidades, para obtener un nuevo flujo factible, aumentando así el valor del flujo.



Ahora, notemos que podemos hacer algo similar con el  $a, z$ -camino sobre el grafo subyacente (la arista  $be$  se toma en sentido opuesto)  $a, d, e, b, c, z$ . Modificando el flujo sobre las aristas de este camino, aumentamos el valor del flujo.



Notemos que en este caso, modificamos el flujo de cada arista en 2 unidades. Pero en algunas sumamos mientras que en otras restamos esa cantidad. Lo hicimos de manera que se preserve la propiedad de conservación del flujo.

A partir del flujo factible dado, pudimos encontrar  $a, z$ -caminos que, modificando el flujo sobre sus aristas, nos permitieron aumentar el valor del flujo. Veremos que ésta es la idea detrás del algoritmo de *Ford-Fulkerson*, que dada una red y un flujo factible de la misma, devuelve como salida un flujo máximo sobre la red.

**Definición 6.** Dada una red  $G$  y  $u, v \in V(G)$  diremos que  $uv \in E(G)$  es una arista *propriadamente orientada*, o simplemente que es una arista *propia*. Diremos que  $uv$  es una arista *impropiamente orientada*, o simplemente que es una arista *impropia*, si  $vu \in E(G)$ .

Si  $f$  es un flujo factible en una red  $G$ , un *camino  $f$ -aumentante* es un  $a, z$ -camino simple  $P$  en el grafo subyacente (es decir, ignorando la dirección de las aristas) tal que para cada arista  $e = uv$  del camino  $P$  se tiene que  $f(e) < c(e)$  si  $e$  es una arista propia en  $P$  (i.e. si en  $P$  se recorre en su sentido), y  $f(e) > 0$  si  $e$  es una arista impropia en  $P$  (i.e. si en  $P$  se recorre en su sentido opuesto).

Si  $f$  es un flujo factible y  $P$  es un camino  $f$ -aumentante, para cada arista  $e$  del camino  $P$  definimos  $\varepsilon(e) = c(e) - f(e)$  si  $e$  es propia y  $\varepsilon(e) = f(e)$  si  $e$  es impropia. La *tolerancia* de un camino  $f$ -aumentante  $P$  es  $\Delta = \min\{\varepsilon(e) : e \in P\}$

**Proposición 7.** Si  $f$  es un flujo factible en una red  $G$  y  $P$  es un camino  $f$ -aumentante con tolerancia  $\Delta$ , entonces  $f' : E(G) \rightarrow \mathbb{R}_0^+$  dado por

$$f'(uv) = \begin{cases} f(uv) + \Delta, & \text{si } uv \text{ es una arista propia en } P, \\ f(uv) - \Delta, & \text{si } uv \text{ es una arista impropia en } P, \\ f(uv), & \text{en otro caso,} \end{cases}$$

es un flujo factible de  $G$  con  $val(f') = val(f) + \Delta$ .

**Demostración.** Sea  $f$  un flujo factible en una red  $G$  y  $P$  un camino  $f$ -aumentante de tolerancia  $\Delta$ . Observemos que por la definición de tolerancia, resulta  $\Delta > 0$ , pues  $\varepsilon(e) > 0$  para cada arista  $e$  de  $P$ . Si  $e$  es una arista que no está en  $P$ , entonces

$$f'(e) = f(e) \geq 0, \quad \text{y} \quad f'(e) = f(e) \leq c(e).$$

Si  $e$  es una arista propia de  $P$ , entonces

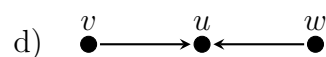
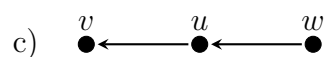
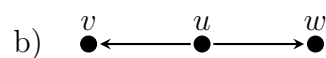
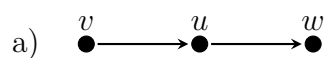
$$f'(e) = f(e) + \Delta \geq f(e) \geq 0, \quad \text{y} \quad f'(e) = f(e) + \Delta \leq f(e) + \varepsilon(e) = f(e) + c(e) - f(e) = c(e).$$

Si  $e$  es una arista impropia de  $P$ , entonces

$$f'(e) = f(e) - \Delta \geq f(e) - \varepsilon(e) = f(e) - f(e) = 0, \quad \text{y} \quad f'(e) = f(e) - \Delta \leq f(e) \leq c(e).$$

Por lo tanto, para cada arista  $e$  de la red tenemos que  $0 \leq f'(e) \leq c(e)$ .

Para ver que  $f'$  es un flujo factible, resta ver que tiene la propiedad de conservación del flujo. Para ello, debemos chequear que para todo vértice interno del camino  $P$  se verifica dicha propiedad, pues el flujo solo fue modificado en las aristas del camino. Sea entonces  $u$  un vértice interno de  $P$ . Luego  $u$  es extremo de exactamente dos aristas de  $P$ , cuyos otros extremos son dos vértices  $v$  y  $w$ , y estamos en una de las siguientes cuatro situaciones con respecto a la dirección de las aristas de  $P$  incidentes en  $u$ :



a) Observemos que tanto  $vu$  como  $uw$  son aristas propias. Tenemos entonces

$$\begin{aligned} f'^+(u) &= \sum_{x \in V(G)} f'(u, x) = \sum_{\substack{x \in V(G) \\ x \neq w}} \underbrace{f'(u, x)}_{=f(u, x)} + \underbrace{f'(u, w)}_{=f(u, w) + \Delta} = f^+(u) + \Delta \\ f'^-(u) &= \sum_{x \in V(G)} f'(x, u) = \sum_{\substack{x \in V(G) \\ x \neq v}} \underbrace{f'(x, u)}_{=f(x, u)} + \underbrace{f'(v, u)}_{=f(v, u) + \Delta} = f^-(u) + \Delta \end{aligned}$$

Luego, tenemos

$$f'^+(u) = f^+(u) + \Delta = f^-(u) + \Delta = f'^-(u)$$

b) En este caso tenemos que  $vu$  es una arista impropia mientras que  $uw$  es una arista propia. Tenemos entonces

$$\begin{aligned} f'^+(u) &= \sum_{x \in V(G)} f'(u, x) = \sum_{\substack{x \in V(G) \\ x \neq v, w}} \underbrace{f'(u, x)}_{=f(u, x)} + \underbrace{f'(u, v)}_{=f(u, v) - \Delta} + \underbrace{f'(u, w)}_{=f(u, w) + \Delta} = f^+(u) \\ f'^-(u) &= \sum_{x \in V(G)} \underbrace{f'(x, u)}_{=f(x, u)} = f^-(u) \end{aligned}$$

Luego, como  $f^+(u) = f^-(u)$  por ser  $f$  un flujo factible, tenemos

$$f'^+(u) = f^+(u) = f^-(u) = f'^-(u)$$

c) Observemos que tanto  $vu$  como  $uw$  son aristas impropias. Tenemos entonces

$$\begin{aligned} f'^+(u) &= \sum_{x \in V(G)} f'(u, x) = \sum_{\substack{x \in V(G) \\ x \neq v}} \underbrace{f'(u, x)}_{=f(u, x)} + \underbrace{f'(u, v)}_{=f(u, v) - \Delta} = f^+(u) - \Delta \\ f'^-(u) &= \sum_{x \in V(G)} f'(x, u) = \sum_{\substack{x \in V(G) \\ x \neq w}} \underbrace{f'(x, u)}_{=f(x, u)} + \underbrace{f'(w, u)}_{=f(w, u) - \Delta} = f^-(u) - \Delta \end{aligned}$$

Luego, tenemos

$$f'^+(u) = f^+(u) - \Delta = f^-(u) - \Delta = f'^-(u)$$

d) En este caso tenemos que  $vu$  es una arista propia mientras que  $uw$  es una arista impropia. Tenemos entonces

$$\begin{aligned} f'^+(u) &= \sum_{x \in V(G)} \underbrace{f'(u, x)}_{=f(u, x)} = f^+(u) \\ f'^-(u) &= \sum_{x \in V(G)} f'(x, u) = \sum_{\substack{x \in V(G) \\ x \neq v, w}} \underbrace{f'(x, u)}_{=f(x, u)} + \underbrace{f'(v, u)}_{=f(v, u) + \Delta} + \underbrace{f'(w, u)}_{=f(w, u) - \Delta} = f^-(u) \end{aligned}$$

Luego, tenemos

$$f'^+(u) = f^+(u) = f^-(u) = f'^-(u)$$

Luego, como  $f'$  verifica  $0 \leq f'(e) \leq c(e)$  para toda arista  $e \in E(G)$  y  $f'^+(u) = f'^-(u)$  para todo vértice  $u \in V(G) \setminus \{a, z\}$ ,  $f'$  es un flujo factible.  $\square$

### Algoritmo de Ford-Fulkerson

**Idea.** Iniciar con un flujo factible (por ejemplo, flujo  $f(e) = 0$  para toda arista  $e$ ). Buscamos un camino  $f$ -aumentante y aumentamos el flujo  $\Delta$  unidades (donde  $\Delta$  es la tolerancia del camino). Repetimos hasta que no haya caminos  $f$ -aumentantes, en ese momento el flujo es máximo.

A lo largo de la ejecución del algoritmo iremos etiquetando cada vértice  $u$  con etiquetas de la forma  $(x, \alpha)$ , donde diremos que  $x$  es predecesor de  $u$  y  $\alpha = \varepsilon(u)$ .

**Entrada.**  $G$  red con fuente  $a$ , sumidero  $z$ , capacidades  $c(e)$  para  $e \in E(G)$ .

**Salida.** Un flujo máximo para  $G$ .

**Inicialización.** Consideramos un flujo factible  $f$  de  $G$  (puede ser flujo cero).

**Iteración.**

**Paso 1.** Etiquetar  $a$  con  $(-, \infty)$ .  $U = \{a\}$ .

**Paso 2.** Mientras  $z$  no fue etiquetado

Si  $U = \emptyset$ , entonces el algoritmo termina. Si no, elijo  $v \in U$ .

Para todo vértice  $x$  no etiquetado aún tal que  $vx \in E(G)$ , si  $f(vx) < c(vx)$ , entonces agrego  $x$  a  $U$  y lo etiqueto con

$$(v, \min\{\varepsilon(v), c(vx) - f(vx)\})$$

Para todo vértice  $x$  no etiquetado aún tal que  $xv \in E(G)$ , si  $f(vx) > 0$ , entonces agrego  $x$  a  $U$  y lo etiqueto con

$$(v, \min\{\varepsilon(v), f(vx)\})$$

A continuación, borramos a  $v$  de  $U$ .

**Paso 3.** Si  $z$  está etiquetado, sea  $\Delta = \varepsilon(z)$ . Construyo un  $a, z$ -camino  $P$  “de atrás hacia adelante”, de la siguiente manera:  $w_0 = z$ , y para  $i > 0$ ,  $w_i$  es el predecesor de  $w_{i-1}$ , hasta llegar a  $w_k = a$ . De esta manera,  $P : w_k, w_{k-1}, \dots, w_0$  es un  $a, z$ -camino. Para cada arista  $e = w_i w_{i-1}$  de  $P$ , actualizamos el flujo  $f$ . Si  $e$  es una arista propia,  $f(e) \leftarrow f(e) + \Delta$ , y si  $e$  es una arista impropia,  $f(e) \leftarrow f(e) - \Delta$ .

A continuación, borramos todas las etiquetas y volvemos al Paso 1.

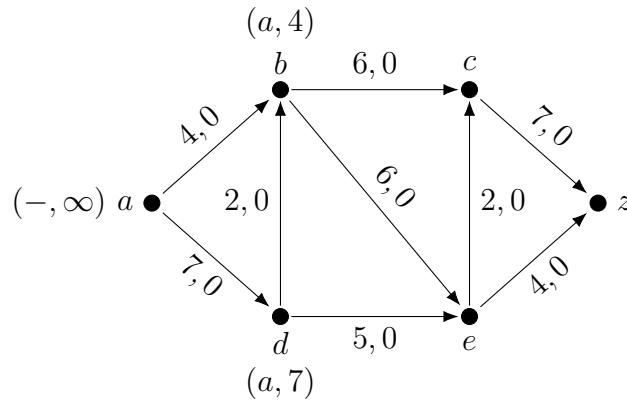
**Ejemplo.** Consideremos nuevamente la red de los ejemplos anteriores. Ejecutamos el algoritmo de Ford-Fulkerson iniciando con el flujo factible nulo.

**Iteración 1. Paso 1.** Etiquetamos  $a \leftarrow (-, \infty)$ .  $U = \{a\}$ .

**Paso 2.** Como  $U \neq \emptyset$ , elegimos un vértice en  $U$ . El único candidato es  $a$ . Exploramos a sus vecinos, que son  $b$  y  $d$ . Como cumplen con las condiciones de etiquetado, los etiquetamos:

$$b \leftarrow (a, \min\{c(ab) - f(ab), \varepsilon(a)\} = 4), \quad d \leftarrow (a, \min\{c(ad) - f(ad), \varepsilon(a)\} = 7)$$

Actualizamos el conjunto  $U$ :  $U = \{b, d\}$ .

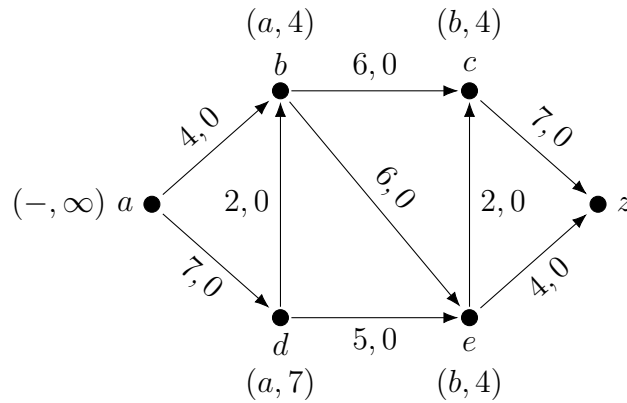


Repetimos el paso 2 hasta haber etiquetado a  $z$ .

**Paso 2.** Como  $U \neq \emptyset$ , elegimos un vértice en  $U$ . Elegimos a  $b$ . Exploramos a sus vecinos aún sin etiquetar, que son  $c$  y  $e$ . Como cumplen con las condiciones de etiquetado, los etiquetamos:

$$c \leftarrow (b, \min\{c(bc) - f(bc), \varepsilon(b)\} = 4), \quad e \leftarrow (b, \min\{c(be) - f(be), \varepsilon(b)\} = 4)$$

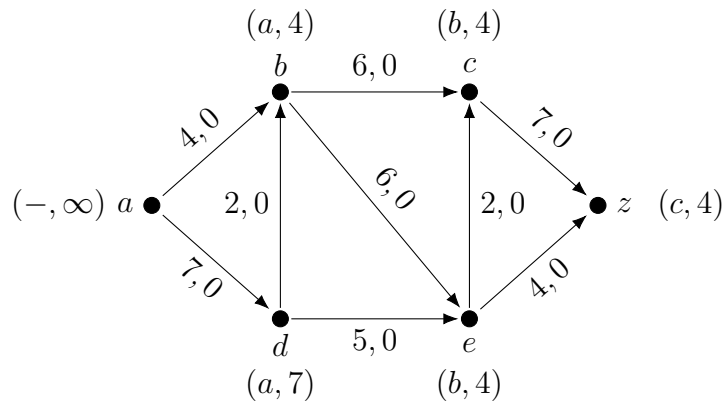
Actualizamos el conjunto  $U$ :  $U = \{d, c, e\}$ .



**Paso 2.** Como  $U \neq \emptyset$ , elegimos un vértice en  $U$ . Elegimos a  $c$ . Exploramos a sus vecinos aún sin etiquetar, el único es  $z$ . Como cumple con las condiciones de etiquetado, lo etiquetamos:

$$z \leftarrow (c, \min\{c(cz) - f(cz), \varepsilon(c)\} = 4)$$

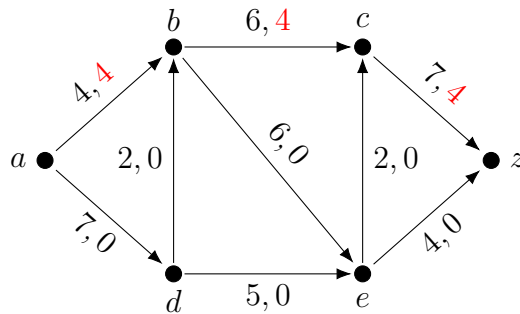
Actualizamos el conjunto  $U$ :  $U = \{d, e, z\}$ .





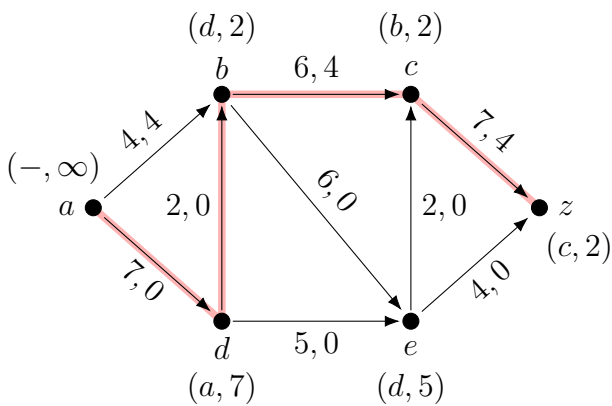
Como  $z$  fue etiquetado, vamos al Paso 3.

**Paso 3.**  $\Delta = \varepsilon(z) = 4$ . “Reconstruimos” el  $a, z$ -camino  $P : a, b, c, z$  y actualizamos el flujo sobre las aristas de este camino. Si las aristas son propias, aumentamos el flujo en  $\Delta = 4$  unidades, y si son impropias, lo restamos. A continuación, borramos las etiquetas.

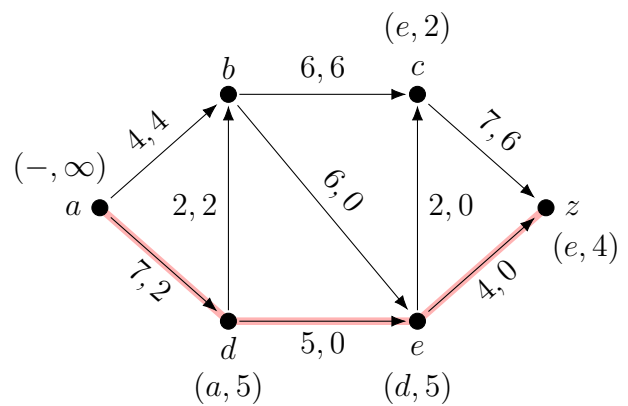


En cada iteración, no es necesario que escribamos paso a paso los detalles como lo hicimos arriba. Sólo lo hicimos para mostrar cómo se ejecuta el algoritmo. Es suficiente con exhibir cómo queda la red etiquetada al finalizar cada iteración. Es lo que haremos con lo que queda de la ejecución.

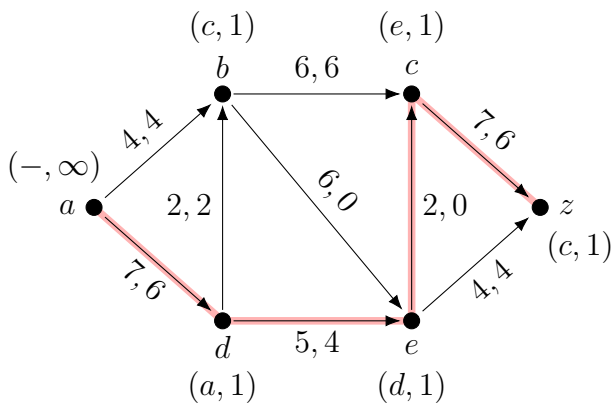
**Iteración 2.**



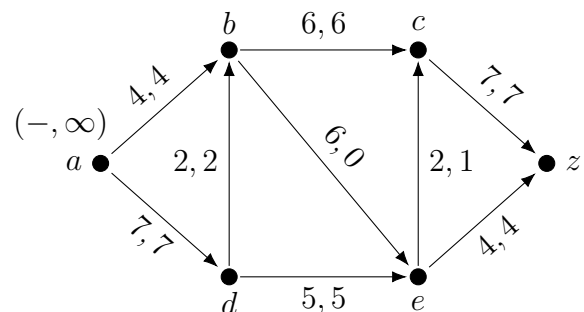
**Iteración 3.**



**Iteración 4.**

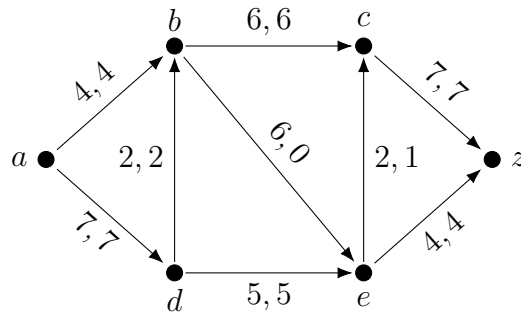


**Iteración 5.**



El algoritmo termina porque en algún momento de esta iteración,  $U = \emptyset$  antes de haber etiquetado a  $z$ .

El flujo que devuelve el algoritmo de Ford-Fulkerson sobre esta red es el siguiente flujo  $f$  con  $val(f) = 11$ .



## Flujos máximos y cortes mínimos

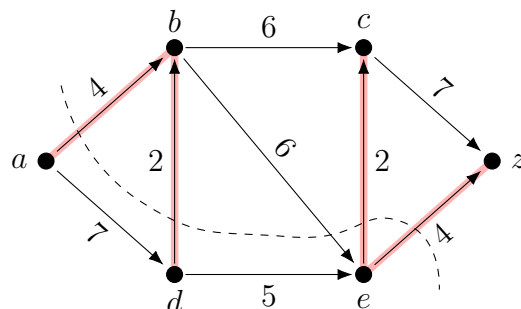
**Definición 8.** En una red  $G$  con fuente  $a$  y sumidero  $z$ , un *corte*  $(S, T)$  consiste en el conjunto de aristas  $\{uv \in E(G) : u \in S, v \in T\}$ , donde  $S$  y  $T$  son conjuntos de vértices que forman una partición de  $V(G)$  tales que  $a \in S, z \in T$ . La *capacidad* del corte  $(S, T)$ , que denotaremos  $cap(S, T)$ , es la suma de las capacidades de las aristas en el corte. Un *corte mínimo* es un corte con capacidad mínima.

**Observación 9.** ■ Notemos que una red es un grafo dirigido. Un corte  $(S, T)$ , tiene solamente aquellas aristas que van **desde** un vértice  $u \in S$  **hacia** un vértice  $v \in T$ .

- Si  $(S, T)$  es un corte, entonces  $T = \bar{S}$ , ya que forman una partición. Luego, al referirnos a un corte, es suficiente indicar el conjunto  $S$ , y a veces notaremos directamente  $(S, \bar{S})$ .

**Ejemplo.** En la red dada en los ejemplos anteriores, tenemos que para  $S = \{a, d, e\}$ ,  $(S, \bar{S})$  es un corte con capacidad

$$cap(S, \bar{S}) = \sum_{u \in S} \sum_{v \notin S} c(uv) = c(ab) + c(db) + c(ec) + c(ez) = 4 + 2 + 2 + 4 = 12.$$



**Proposición 10.** Sea  $f$  un flujo factible en una red  $G$ , y sea  $(S, \bar{S})$  un corte en la red. Entonces

$$val(f) \leq cap(S, \bar{S})$$

**Demostración.** Sea  $G$  un red con fuente  $a$  y sumidero  $z$ , y sean  $f$  un flujo y  $(S, \bar{S})$  un corte de  $G$ . Por un lado, recordemos que  $f^-(a) = 0$ , y tenemos que

$$val(f) = f^+(a) = f^+(a) - f^-(a) = \sum_{u \in V(G)} f(a, u) - \sum_{u \in V(G)} f(u, a). \quad (1)$$

Por otro lado, para cada  $v \in V(G) \setminus \{a, z\}$ , por la propiedad de conservación del flujo, tenemos que

$$f^+(v) - f^-(v) = \sum_{u \in V(G)} f(v, u) - \sum_{u \in V(G)} f(u, v) = 0. \quad (2)$$

En particular, (2) vale para todo  $w \in S \setminus \{a\}$ . Luego, tenemos que

$$\begin{aligned} val(f) &= f^+(a) - f^-(a) = (f^+(a) - f^-(a)) + \sum_{\substack{w \in S \\ w \neq a}} (f^+(w) - f^-(w)) = \\ &= \sum_{w \in S} (f^+(w) - f^-(w)) = \sum_{w \in S} \sum_{u \in V(G)} f(w, u) - \sum_{w \in S} \sum_{u \in V(G)} f(u, w) = \\ &= \sum_{w \in S} \sum_{u \in S} f(w, u) + \sum_{w \in S} \sum_{u \notin S} f(w, u) - \sum_{w \in S} \sum_{u \in S} f(u, w) - \sum_{w \in S} \sum_{u \notin S} f(u, w) = \\ &= \sum_{w \in S} \sum_{u \notin S} f(w, u) - \sum_{w \in S} \sum_{u \notin S} \underbrace{f(u, w)}_{\geq 0} \leq \sum_{w \in S} \sum_{u \notin S} \underbrace{f(w, u)}_{\leq c(wu)} \leq \sum_{w \in S} \sum_{u \notin S} c(wu) = cap(S, \bar{S}) \end{aligned}$$

Luego,  $val(f) \leq cap(S, \bar{S})$ , como queríamos.  $\square$

**Corolario 11.** Sea  $G$  una red. Si  $f$  es un flujo y  $(S, \bar{S})$  un corte de  $G$  tales que  $val(f) = cap(S, \bar{S})$ , entonces  $f$  es un flujo máximo,  $(S, \bar{S})$  es un corte mínimo. Más aún, la igualdad  $val(f) = cap(S, \bar{S})$  vale si y solo si para toda arista  $uv$  con  $u \in S$  y  $v \notin S$ ,  $f(u, v) = c(uv)$  y para toda arista  $uv$  con  $u \notin S$  y  $v \in S$ ,  $f(u, v) = 0$ .

Hemos visto entonces que dada una red  $G$ , si encontramos un flujo  $f$  y un corte  $(S, \bar{S})$  para los cuales  $val(f) = cap(S, \bar{S})$ , ambos son óptimos. A continuación, probaremos que esto siempre es posible. Más aún, probaremos que cuando el algoritmo de Ford-Fulkerson termina, obtenemos un flujo máximo y además, si  $S$  es el conjunto de los vértices que han sido etiquetados en la última iteración, entonces  $(S, \bar{S})$  es un corte mínimo.

**Teorema 12.** En toda red, el valor de un flujo máximo es igual a la capacidad de un corte mínimo.

**Demostración.** Sea  $G$  una red, con fuente  $a$  y sumidero  $z$ . Realizaremos aquí la demostración para el caso en el cual las capacidades son números racionales no negativos. Consideremos el flujo inicial cero, esto es,  $f(e) = 0$  para toda arista  $e \in E(G)$ .  $f$  es un flujo factible. Ejecutamos para la red  $G$  y el flujo  $f$  el algoritmo de Ford-Fulkerson.

Observemos que en cada iteración del algoritmo en la que  $z$  es etiquetado, lo que obtenemos es un camino  $f$ -aumentante de tolerancia  $\Delta = \varepsilon(z)$ , lo que nos permite aumentar el flujo en  $\Delta$  unidades. Cuando las capacidades de la red son números racionales, es decir, de la forma  $\frac{p}{q}$ ,  $\Delta$  resulta un múltiplo de  $\frac{1}{\alpha}$ , donde  $\alpha$  es el mínimo común múltiplo de los denominadores de las capacidades de toda la red. Luego, por la propiedad arquimediana de los números reales, existe una cantidad

finita de iteraciones después de las cuales el flujo alcanza la capacidad de algún corte. Esto es, si  $(S, \bar{S})$  es un corte, existe  $N$  tal que  $\frac{N}{a} \geq \text{cap}(S, \bar{S})$ , con lo cual después de a lo sumo  $N$  iteraciones, el algoritmo termina, ya que  $\text{cap}(S, \bar{S})$  es una cota superior para el flujo máximo de la red.

El algoritmo termina cuando no es posible encontrar un camino  $f$ -aumentante. Llamemos  $S$  al conjunto de vértices que fueron etiquetados en la última iteración. Tenemos que  $z \notin S$ , de lo contrario habríamos encontrado un camino  $f$ -aumentante, y  $a \in S$  pues  $a$  es etiquetado en cada iteración. Luego,  $(S, \bar{S})$  es un corte.

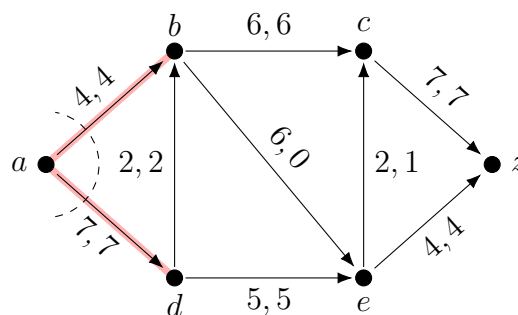
Notemos que si  $uv \in E(G)$  con  $u \in S$  y  $v \notin S$ , entonces  $f(u, v) = c(uv)$ . De lo contrario,  $v$  habría sido etiquetado en esta iteración con la etiqueta  $v \leftarrow (u, \min\{\varepsilon(u), c(uv) - f(u, v)\})$ . Por otro lado, si  $uv \in E(G)$  con  $u \notin S$  y  $v \in S$ , entonces  $f(uv) = 0$ , pues de no ser así  $u$  habría sido etiquetado en esta iteración con la etiqueta  $u \leftarrow (v, \min\{\varepsilon(v), f(u, v)\})$ . Por lo tanto, por el Corolario 11 resulta que  $\text{val}(f) = \text{cap}(S, \bar{S})$ .

En consecuencia, el flujo  $f$  hallado con el algoritmo de Ford-Fulkerson es un flujo máximo de la red que coincide con la capacidad del corte  $(S, \bar{S})$ , donde  $S$  es el conjunto de vértices etiquetados en la última iteración del algoritmo.  $\square$

**Observación 13.** En la prueba del Teorema 12, hemos considerado las capacidades como números racionales. Si esto no es así, esta prueba no es válida. Más aún, si las capacidades de la red no son racionales, el algoritmo de Ford-Fulkerson podría no terminar. Una demostración general puede verse, por ejemplo, en el libro *Graph Theory* de J. A. Bondy y U. S. R. Murty.

En general, trabajaremos con redes donde las capacidades son racionales, y por ende, podemos utilizar el algoritmo de Ford-Fulkerson.

**Ejemplo.** Observemos que en la ejecución que realizamos del algoritmo de Ford-Fulkerson, en la quinta y última iteración, el único vértice etiquetado fue  $a$ . Luego,  $(\{a\}, V(G) \setminus \{a\})$  resulta un corte mínimo para la red de capacidad 11. Todas las aristas que salen de  $a$  están saturadas por el flujo, es decir,  $f(e) = c(e)$ .



Notemos que éste no es el único corte mínimo de la red. Por ejemplo, tomando  $S = \{a, d\}$  o  $S = \{a, b, c, d, e\}$ , resulta  $\text{cap}(S, \bar{S}) = 11$ .

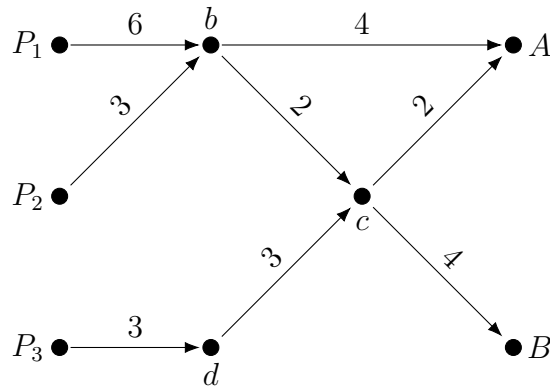
**Corolario 14.** Si todas las capacidades en una red son números naturales, entonces existe un flujo máximo que le asigna un flujo entero a cada arista.

## Algunos ejemplos de aplicación

### Red de bombeo

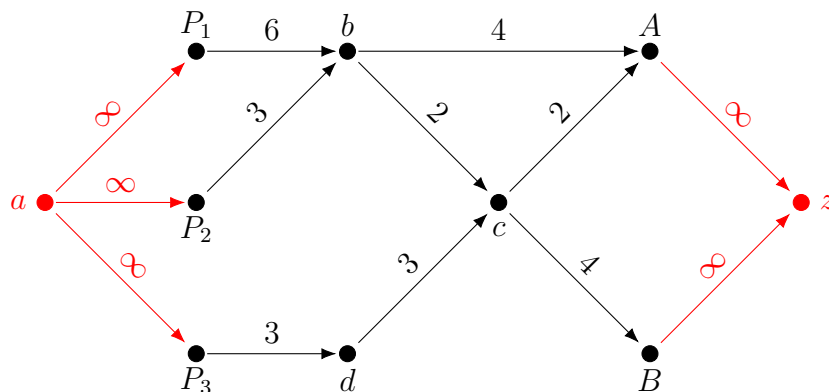
El siguiente grafo dirigido representa una red de bombeo por medio de la cual se entrega agua para dos ciudades  $A$  y  $B$ , desde tres pozos  $P_1$ ,  $P_2$  y  $P_3$ . Los vértices  $b$ ,  $c$  y  $d$  representan las estaciones de bombeo intermedias. Las capacidades de los sistemas intermedios se muestran en las aristas, en alguna unidad pertinente por unidad de tiempo.

Se desea calcular la mayor cantidad de agua por unidad de tiempo que se puede transportar.



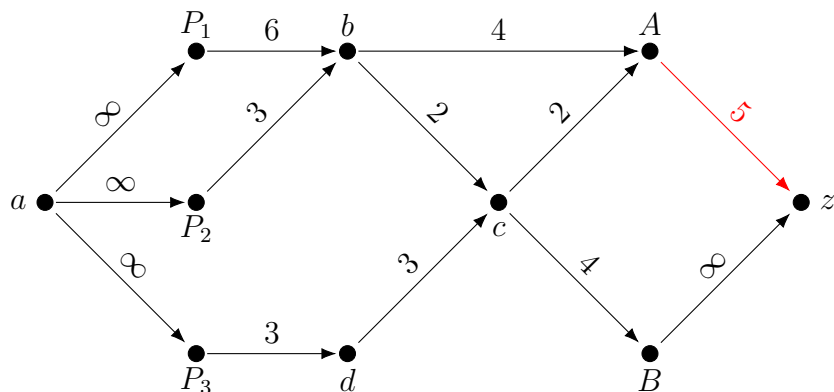
Notemos que para modelizar este problema como una *red*, hace falta identificar la fuente y el sumidero. Tenemos 3 vértices candidatos a ser fuente ( $P_1$ ,  $P_2$  y  $P_3$ ), y dos candidatos a ser sumideros ( $A$  y  $B$ ), pero ninguna elección parece apropiada en términos del problema.

Como alternativa, podemos agregar dos nuevos vértices  $a$  y  $z$ , que serán el origen y destino de la red. Agregamos las aristas  $aP_1$ ,  $aP_2$  y  $aP_3$ , cada una de ellas con capacidad  $\infty$ , y las aristas  $Az$  y  $Bz$  también con capacidad  $\infty$ .

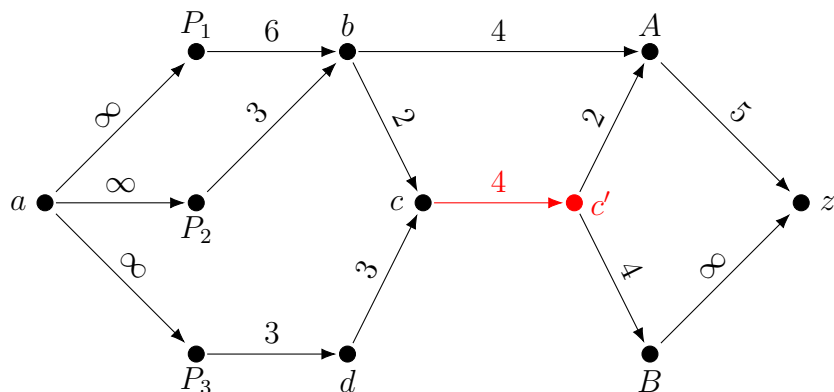


De esta manera, hemos modelizado la situación mediante una red, con origen  $a$  y destino  $z$ . Ahora para calcular la mayor cantidad de agua por unidad de tiempo que se puede transportar, basta con ejecutar el algoritmo de Ford-Fulkerson sobre la misma. Al finalizar, simplemente consideramos el valor del flujo sobre las aristas originales del grafo.

Supongamos ahora que la ciudad  $A$  tiene una limitación por la cual no puede recibir más de 5 unidades. ¿Cómo modelizamos en la red esta nueva restricción? Lo que podemos hacer es representar este requisito en el peso de la arista agregada  $Az$ . Esto es, poniendo  $c(Az) = 5$ . Luego, como en cualquier flujo factible de esta red se debe cumplir con la propiedad de conservación del flujo para el vértice  $A$ , tendremos que el flujo entrante a  $A$  será igual al flujo saliente desde  $A$ , que está acotado superiormente por  $c(Az) = 5$ .



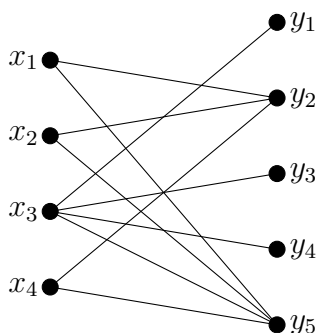
Supongamos ahora que la estación de bombeo intermedia  $c$  no puede bombear más de 4 unidades. ¿Cómo representamos esta restricción en la red? Podemos agregar un nuevo vértice  $c'$ . Agregamos una arista  $cc'$  con capacidad 4, y reemplazamos cada arista  $cx$  por una arista  $c'x$  con la misma capacidad.



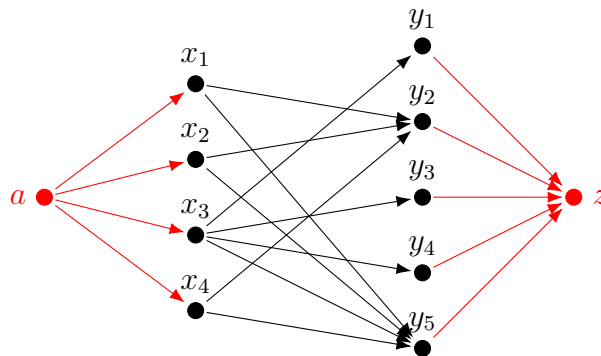
De esta manera, al conseguir un flujo factible en esta red, lo que haremos es considerar el flujo sobre las aristas originales del grafo original, y sobre las aristas  $cA$  y  $cB$ , el flujo asignado a las aristas  $c'A$  y  $c'B$ . Por la propiedad de conservación del flujo sobre el vértice  $c$ , este no excede 4 unidades.

### Matchings en grafos bipartitos

Supongamos que tenemos el siguiente grafo bipartito  $G[X, Y]$  y queremos hallar un matching máximo.



Podemos modelizar este problema como el de encontrar el flujo máximo en una red. Para eso, consideramos la red que se obtiene reemplazando cada arista  $xy$  de  $G$  con  $x \in X$  e  $y \in Y$  por la arista dirigida  $xy$  con capacidad 1. Además, agregamos dos nuevos vértices  $a$  y  $z$ , que serán respectivamente la fuente y el sumidero de la red, y las aristas  $ax$  para cada  $x \in X$ , e  $yz$  para cada  $y \in Y$ , todas ellas también de capacidad 1.



(En la imagen omitimos las capacidades de las aristas, pero todas tienen capacidad 1)

Encontrar un matching máximo es equivalente a hallar un flujo máximo (entero) en esta red (ejercicio). Las aristas saturadas por dicho flujo forman un matching máximo.