

Auxiliar 3-Pauta

Profesor: Fernando Ordóñez

Auxiliar: Renaud Chicoisne

1 Flujo máximo

1.1 Repaso

$$U = \max_{(i,j) \in A} u_{ij}$$

$G(x)$:= Grafo residual de x

$G(x, \Delta)$:= Grafo residual de x conteniendo solamente los arcos (i, j) tales que $r_{ij} \geq \Delta$

Capacity Scaling Algorithm

$x := 0$;

$\Delta := 2^{\lceil \log U \rceil}$;

While $\Delta \geq 1$

 While $G(x, \Delta)$ contiene un camino de s a t

 Encontrar un camino P de s a t en $G(x, \Delta)$;

$\delta := \min_{(i,j) \in P} r_{ij}$;

$\forall (i, j) \in P$

$x_{ij} := x_{ij} + \delta$;

$x_{ji} := x_{ji} - \delta$;

 Actualizar $G(x, \Delta)$;

$\Delta := \Delta/2$

Shortest Augmenting Path Algorithm

$x := 0$;

While $G(x)$ contiene un camino de s a t

 Encontrar un camino más corto P de s a t en $G(x)$;

$\delta := \min_{(i,j) \in P} r_{ij}$;

$\forall (i, j) \in P$

$x_{ij} := x_{ij} + \delta$;

$x_{ji} := x_{ji} - \delta$;

 Actualizar $G(x)$;

Preflow Push Algorithm

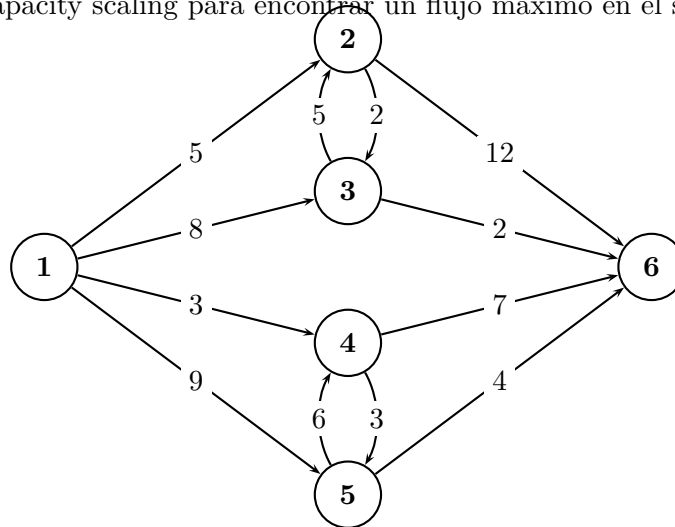
```
 $x := 0;$   
 $(d_i)_{i \in V} := \text{mínimo número de arcos para alcanzar } t \text{ desde } i;$   
 $d_s := n;$   
 $x_{sj} := u_{sj}, \forall (s, j) \in A(s);$   
While  $\exists i \in V : e(i) > 0$   
  Seleccionar un nodo  $i$  tal que  $e(i) > 0;$   
  If  $\exists (i, j) \in A : d_i = d_j + 1$   
     $\delta := \min_{(i,j) \in P} \{e(i), r_{ij}\};$   
     $x_{ij} := x_{ij} + \delta;$   
     $x_{ji} := x_{ji} - \delta;$   
  Else  
     $d_i := \min_{(i,j) \in A(i): r_{ij} > 0} \{d_j + 1\};$ 
```

1.2 Ejercicios

Ejercicio 7.2

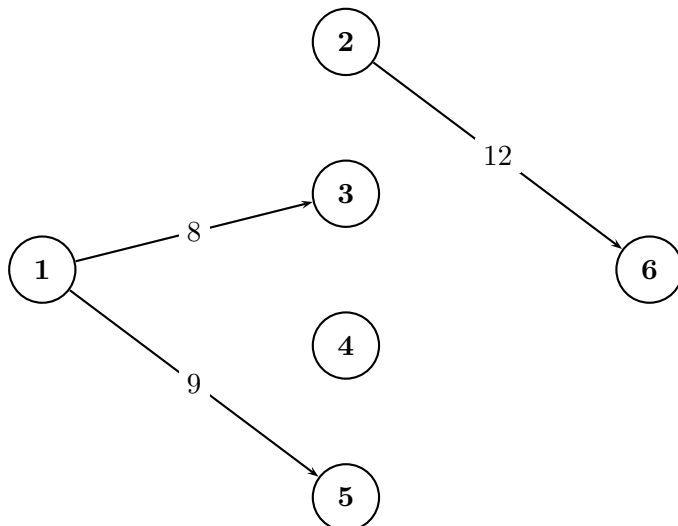
Capacity scaling

Ocupar el algoritmo de Capacity scaling para encontrar un flujo máximo en el siguiente grafo:



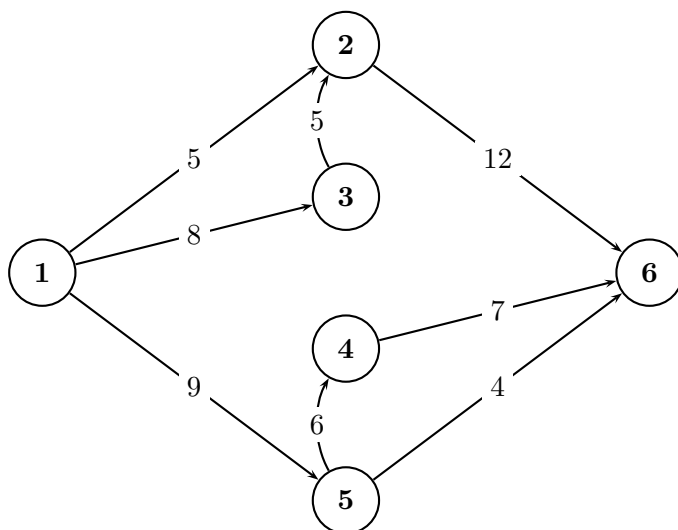
```
 $x := 0;$   
 $U := \max_{(i,j) \in A} u_{ij} = 12;$   
 $\Delta := 2^{\lceil \log U \rceil} = 8;$ 
```

Iteración 1: $G(x, \Delta = 8)$ es el grafo residual de x conteniendo solamente los arcos cuya capacidad residual $r_{ij} \geq \Delta = 8$:

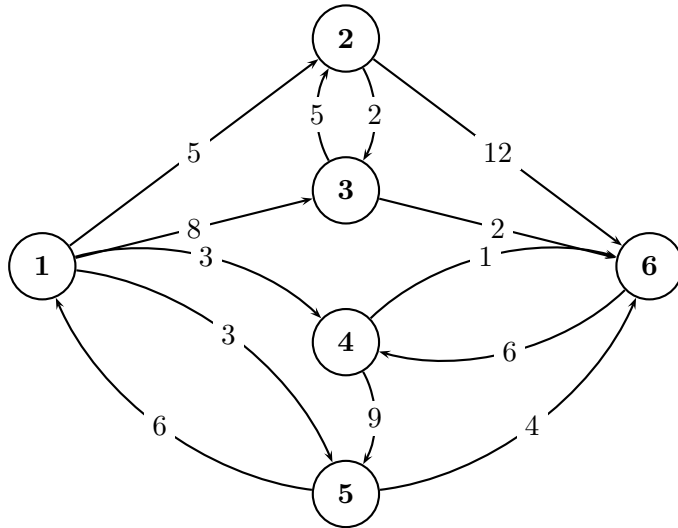


No se puede encontrar ningún camino aumentante, pasamos a la siguiente iteración con $\Delta := \Delta/2 = 4$;

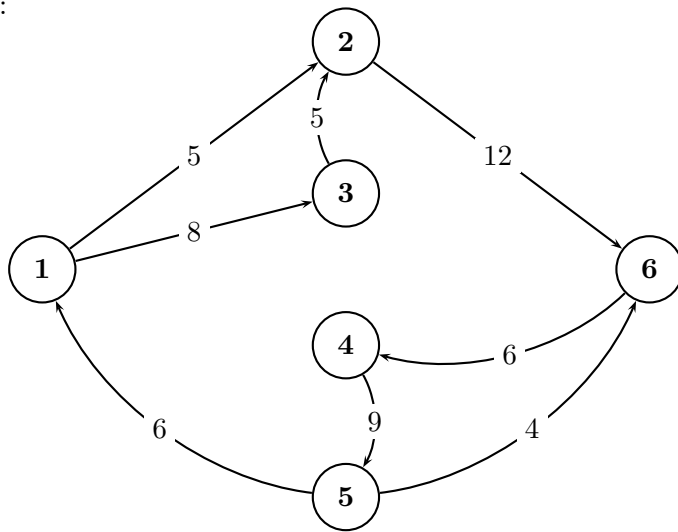
Iteración 2: $G(x, \Delta = 4)$:



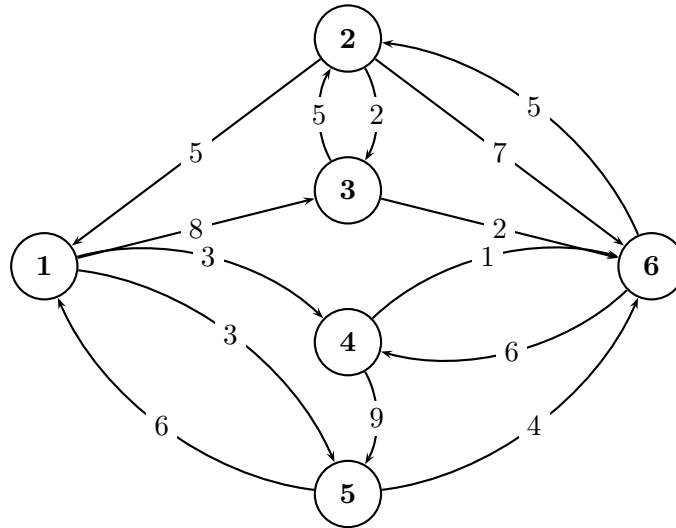
$\{1, 5, 4, 6\}$ es un camino aumentante, y se puede empujar $\delta := \min\{9, 6, 7\} = 6$ unidades de flujo. Por lo tanto $G(x)$ es:



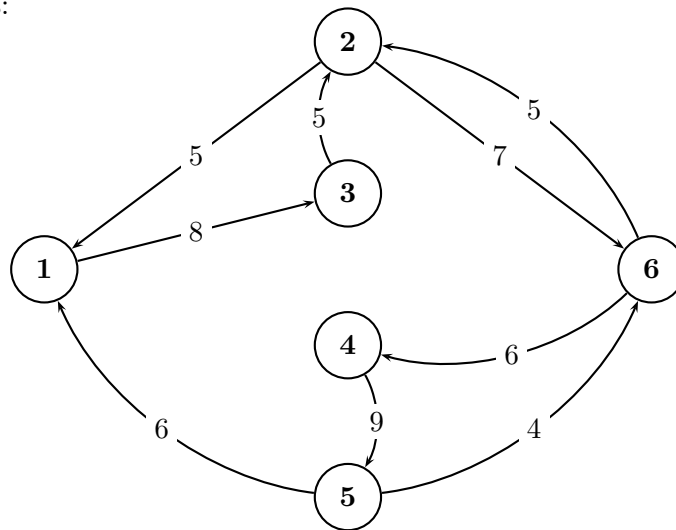
Y entonces $G(x, \Delta = 4)$ es:



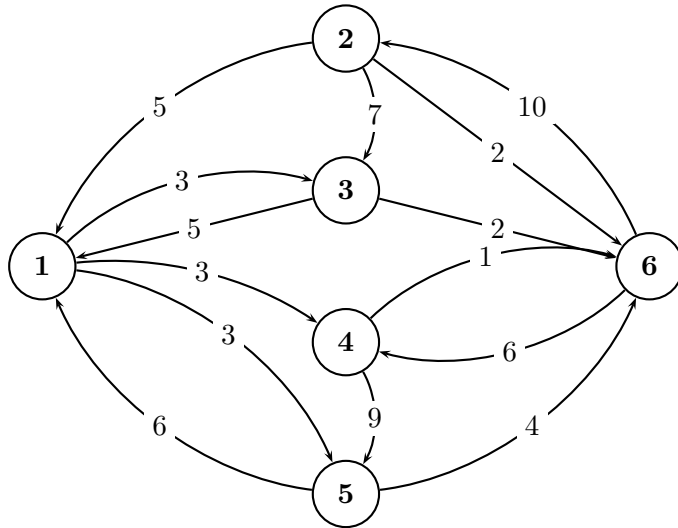
$\{1, 2, 6\}$ es un camino aumentante, y se puede empujar $\delta := \min\{5, 12\} = 5$ unidades de flujo. Por lo tanto $G(x)$ es:



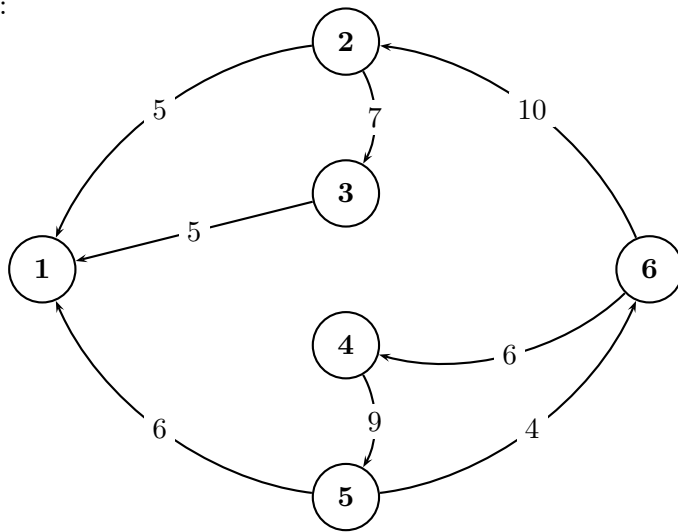
Y entonces $G(x, \Delta = 4)$ es:



$\{1, 3, 2, 6\}$ es un camino aumentante, y se puede empujar $\delta := \min\{8, 5, 7\} = 5$ unidades de flujo. Por lo tanto $G(x)$ es:

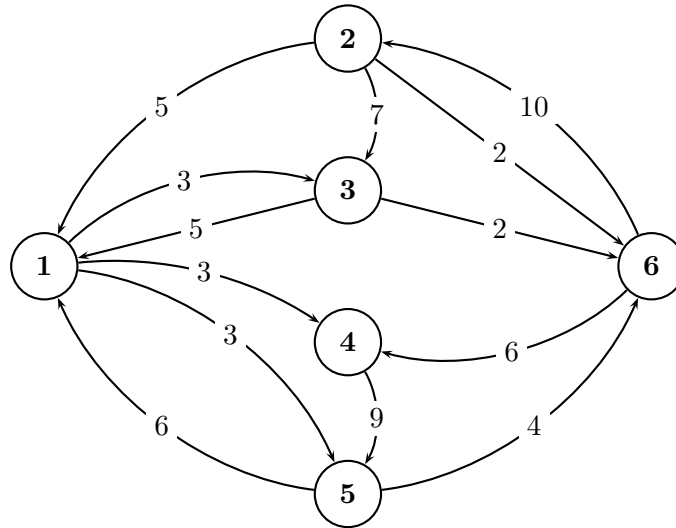


Y entonces $G(x, \Delta = 4)$ es:

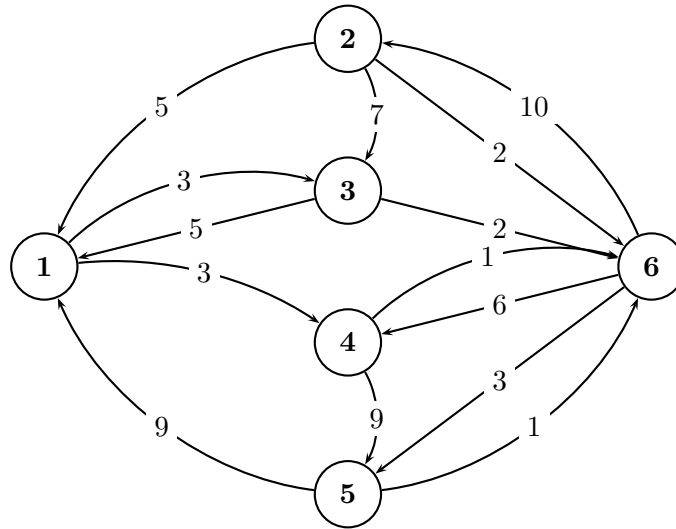


Y no se puede encontrar ningún camino aumentante: $\Delta := \Delta/2 = 2$;

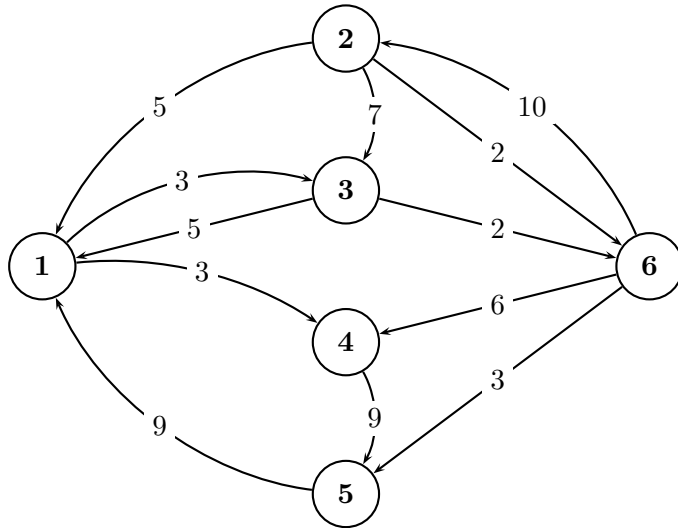
Iteración 3 $G(x, \Delta = 2)$ es:



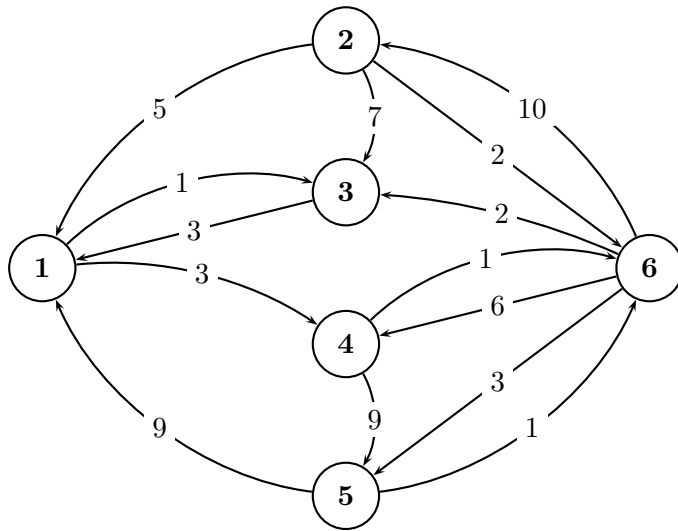
$\{1, 5, 6\}$ es un camino aumentante y se puede empujar $\delta := \min\{3, 4\} = 3$ unidades de flujo. Por lo tanto $G(x)$ es:



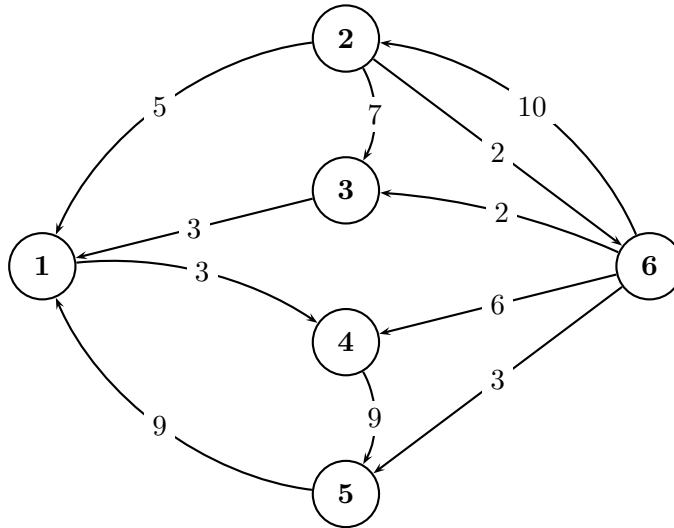
$G(x, \Delta = 2)$ es:



$\{1, 3, 6\}$ es un camino aumentante y se puede empujar $\delta := \min\{3, 2\} = 2$ unidades de flujo. $G(x)$ es:

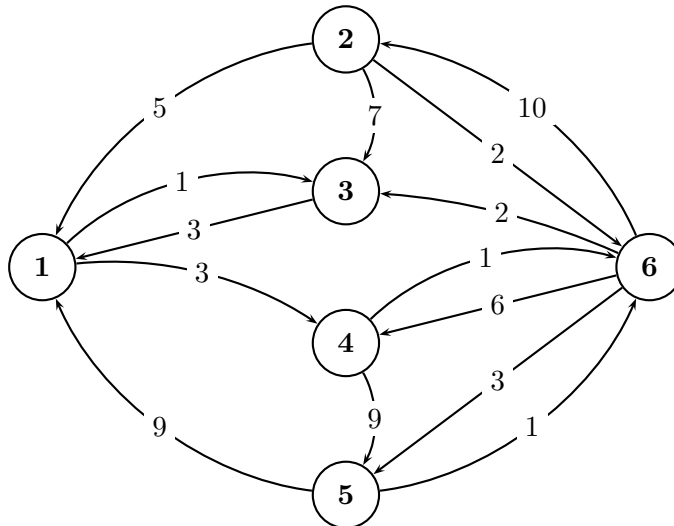


Por lo tanto $G(x, \Delta = 2)$ es:

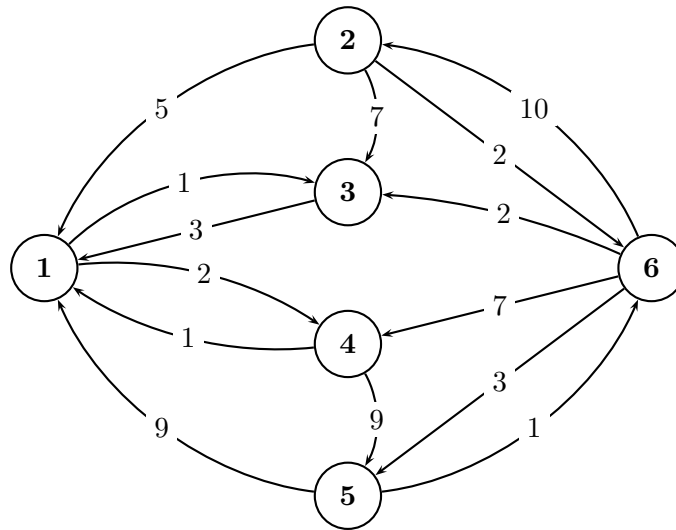


No hay ningún camino aumentante: $\Delta := \Delta/2 = 1$.

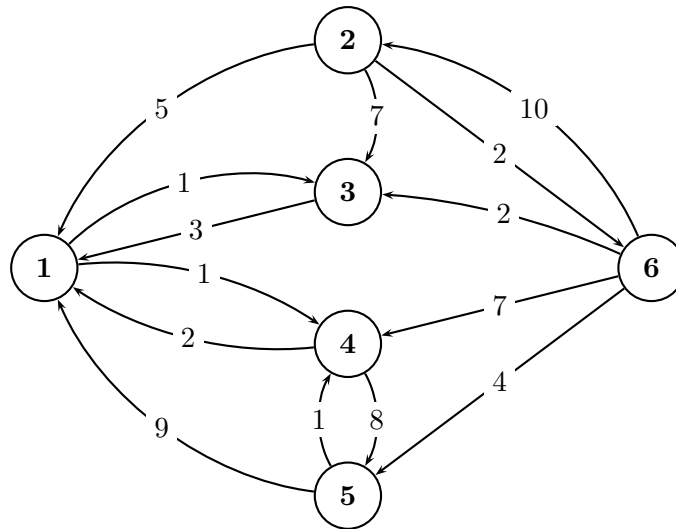
Iteración 4: $G(x, \Delta = 1) = G(x)$ es:



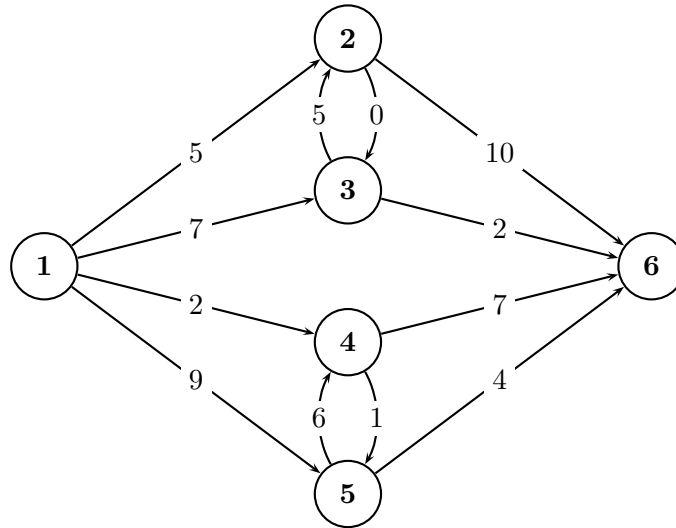
$\{1, 4, 6\}$ es un camino aumentante y se puede empujar $\delta := \min\{3, 1\} = 1$ unidades de flujo. $G(x)$ es:



$\{1, 4, 5, 6\}$ es un camino aumentante y se puede empujar $\delta := \min\{2, 9, 1\} = 1$ unidades de flujo. $G(x)$ es:



No existe ningún camino aumentante y $\Delta = 1$. El algoritmo termina, el grafo residual nos da el siguiente corte mínimo $S = \{1, 3, 4, 5\}$ de capacidad $u_{12} + u_{32} + u_{36} + u_{46} + u_{56} = 5 + 5 + 2 + 7 + 4 = 23$ y con el siguiente flujo máximo:

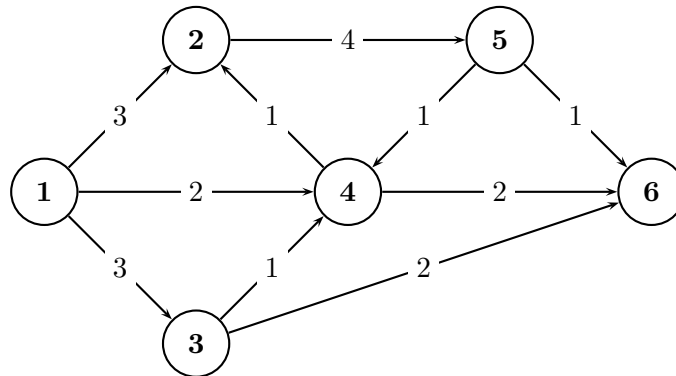


Que vale 23, lo que es igual a la capacidad del corte encontrado gracias al grafo residual.

Ejercicio 7.3

Shortest augmenting path algorithm / Preflow-Push

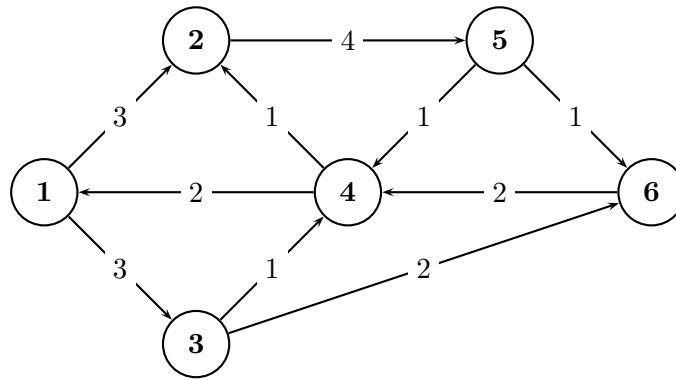
Ocupar el algoritmo de Shortest augmenting path y luego el algoritmo genérico de Preflow-Push para encontrar un flujo máximo en el siguiente grafo:



Shortest augmenting path algorithm Para encontrar un camino más corto en en términos de número de arcos, vimos en la primera auxiliar que se podía ocupar una búsqueda a lo ancho (en $O(m)$). En la pauta de este ejercicio no vamos a detallar las iteraciones de esta búsqueda.

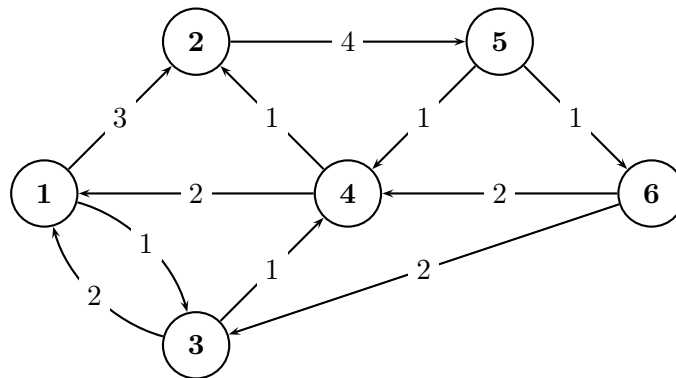
Iteración 1

La búsqueda a lo ancho entrega $\{1, 4, 6\}$ como camino aumentante más corto de 1 a 6. Se puede empujar $\delta := \min\{2, 2\} = 2$ unidades de flujo. $G(x)$ es:



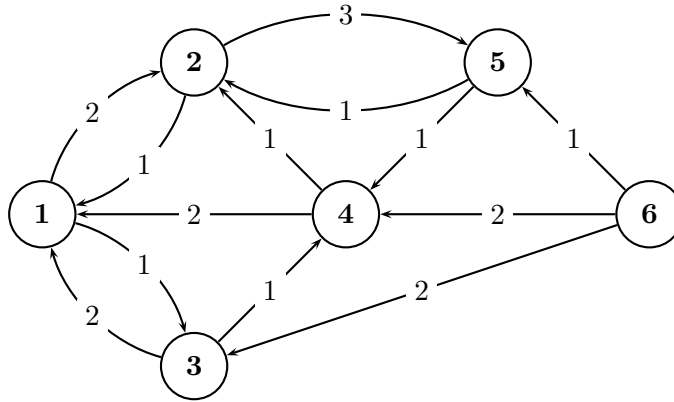
Iteración 2

La búsqueda a lo ancho entrega $\{1, 3, 6\}$ como camino aumentante más corto de 1 a 6. Se puede empujar $\delta := \min\{3, 2\} = 2$ unidades de flujo. $G(x)$ es:



Iteración 3

La búsqueda a lo ancho entrega $\{1, 2, 5, 6\}$ como camino aumentante más corto de 1 a 6. Se puede empujar $\delta := \min\{3, 4, 1\} = 1$ unidades de flujo. $G(x)$ es:



Iteración 4

La búsqueda a lo ancho no puede llegar hasta 6 entonces no existe camino aumentante y el flujo máximo vale 5. Sin embargo, la búsqueda marcó el conjunto $S = \{1, 2, 3, 5, 4\}$ que consiste un corte mínimo de capacidad $u_{36} + u_{46} + u_{56} = 2 + 2 + 1 = 5$, lo que confirma la optimalidad del flujo encontrado.

Preflow-Push Algorithm

$x := 0$; Se calculan los caminos más cortos en terminos de número de arcos desde cada nodo i al sumidero 6:

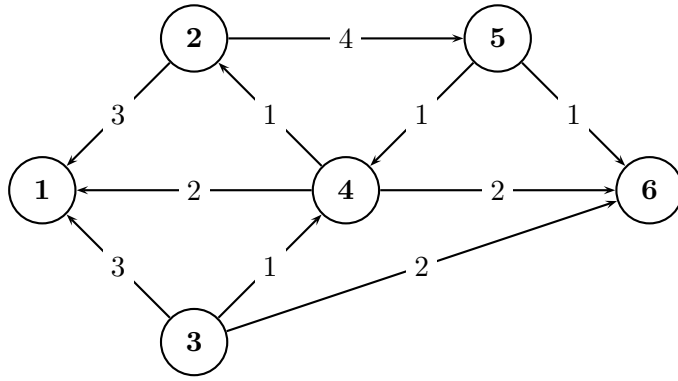
Table 1: Distancias mínimas de i a 6

i	1	2	3	4	5
d_i	$n = 6$	2	1	1	1

Y inicializemos el preflow saturando los arcos saliendo de 1, lo que nos da el siguiente grafo:

Table 2: Excesos $e(\cdot)$

i	1	2	3	4	5	6
$e(i)$	-8	3	3	2	0	0



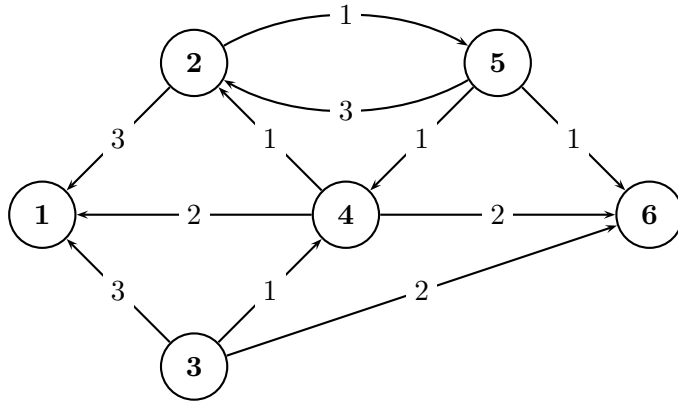
Iteración 1

El nodo 2 tiene exceso $e(2) = 3 > 0$

El arco $(2, 5)$ es admisible porque $d_2 = d_5 + 1$, y se puede empujar $\delta := \min\{e(2) = 3, r_{2,5} = 4\} = 3$ unidades de flujo. Se actualizan el grafo residual y los excesos de los nodos 2 y 5:

Table 3: Iteración 1

i	1	2	3	4	5	6
d_i	6	2	1	1	1	0
$e(i)$	-8	0	3	2	3	0

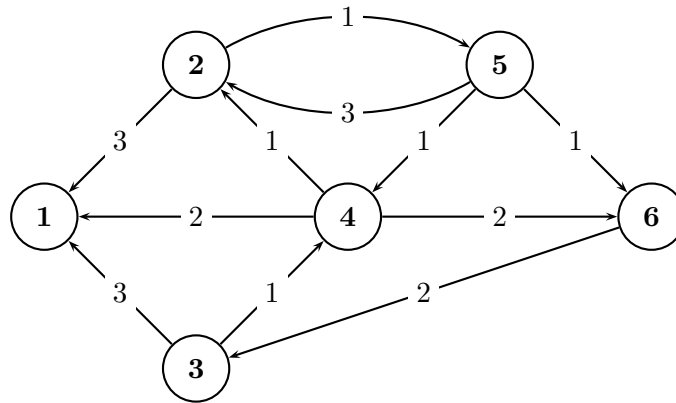


Iteración 2

El nodo 3 tiene exceso $e(3) = 3 > 0$

El arco $(3, 6)$ es admisible porque $d_3 = d_6 + 1$, y se puede empujar $\delta := \min\{e(3) = 3, r_{3,6} = 2\} = 2$ unidades de flujo. Se actualizan el grafo residual y los excesos de los nodos 3 y 6:

i	1	2	3	4	5	6
d_i	6	2	1	1	1	0
$e(i)$	-8	0	1	2	3	2

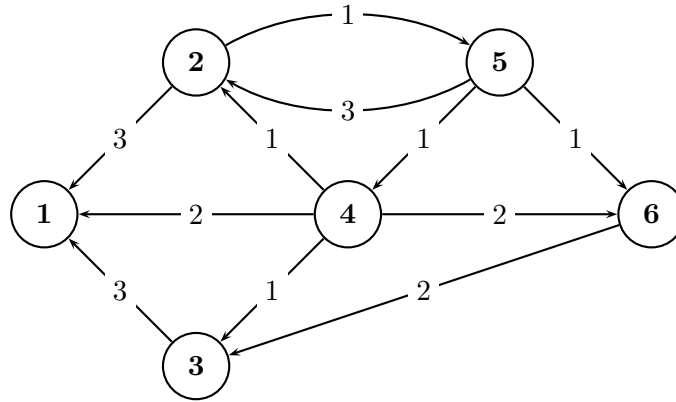


Iteración 3

El nodo 3 tiene exceso $e(3) = 1 > 0$

No hay arco admisible saliendo del nodo, se hace un Relabel de 3: $d_3 := 1 + \min\{d_1, d_4\} = d_4 + 1 = 2$. Ahora el arco $(3, 4)$ es admisible y se puede empujar $\delta := \min\{e(3) = 1, r_{3,4} = 1\} = 1$ unidades de flujo. Se actualizan el grafo residual y los excesos de los nodos 3 y 4:

i	1	2	3	4	5	6
d_i	6	2	2	1	1	0
$e(i)$	-8	0	0	3	3	2



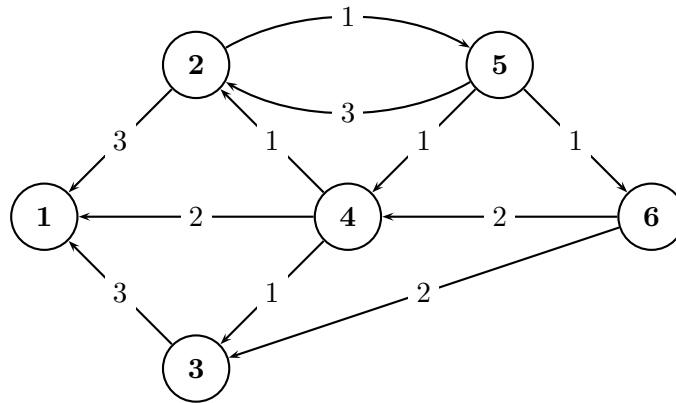
Iteración 4

El nodo 4 tiene exceso $e(4) = 3 > 0$

El arco $(4, 6)$ es admisible y se puede empujar $\delta := \min\{e(4) = 3, r_{4,6} = 2\} = 2$ unidades de flujo. Se actualizan el grafo residual y los excesos de los nodos 4 y 6:

Table 6: Iteración 4

i	1	2	3	4	5	6
d_i	6	2	2	1	1	0
$e(i)$	-8	0	0	1	3	4



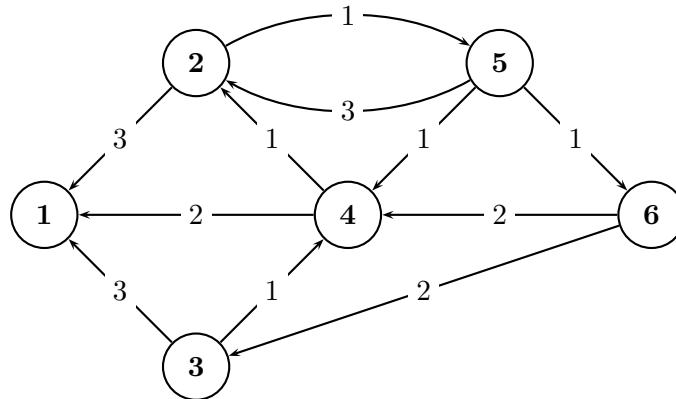
Iteración 5

El nodo 4 tiene exceso $e(4) = 1 > 0$. No hay arco admisible saliendo del nodo, se hace un Relabel

de 4: $d_4 := 1 + \min\{d_1, d_2, d_3\} = d_3 + 1 = 3$. Ahora el arco $(4, 3)$ es admisible y se puede empujar $\delta := \min\{e(4) = 1, r_{4,3} = 1\} = 1$ unidades de flujo. Se actualizan el grafo residual y los excesos de los nodos 4 y 3:

Table 7: Iteración 5

i	1	2	3	4	5	6
d_i	6	2	2	3	1	0
$e(i)$	-8	0	1	0	3	4

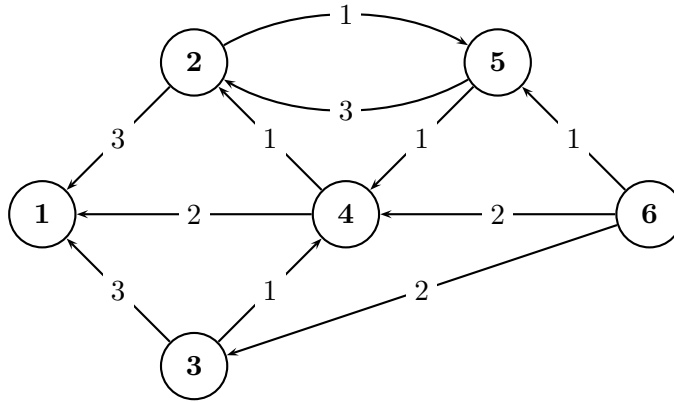


Iteración 6

El nodo 5 tiene exceso $e(5) = 3 > 0$. El arco $(5, 6)$ es admisible y se puede empujar $\delta := \min\{e(5) = 3, r_{5,6} = 1\} = 1$ unidades de flujo. Se actualizan el grafo residual y los excesos de los nodos 5 y 6:

Table 8: Iteración 6

i	1	2	3	4	5	6
d_i	6	2	2	3	1	0
$e(i)$	-8	0	1	0	2	5

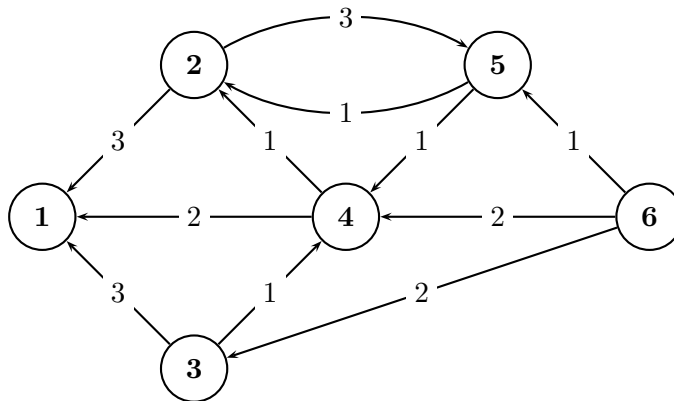


Iteración 7

El nodo 5 tiene exceso $e(5) = 2 > 0$. No hay arco admisible saliendo del nodo, se hace un Relabel de 5: $d_5 := 1 + \min\{d_2, d_4\} = d_2 + 1 = 3$. Ahora el arco $(5, 2)$ es admisible y se puede empujar $\delta := \min\{e(5) = 2, r_{5,2} = 3\} = 2$ unidades de flujo. Se actualizan el grafo residual y los excesos de los nodos 5 y 2:

Table 9: Iteración 7

i	1	2	3	4	5	6
d_i	6	2	2	3	3	0
$e(i)$	-8	2	1	0	0	5



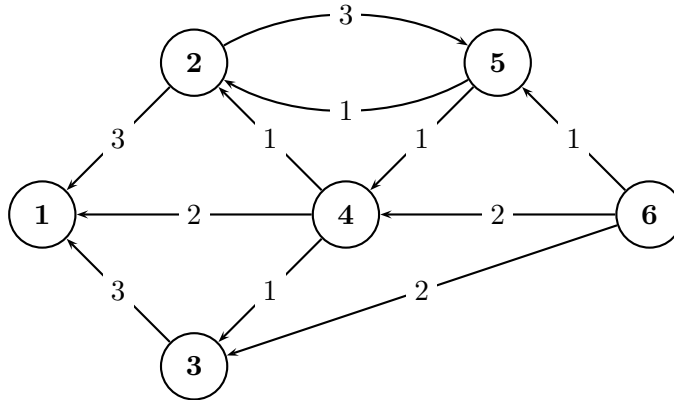
Iteración 8

El nodo 3 tiene exceso $e(3) = 1 > 0$. No hay arco admisible saliendo del nodo, se hace un Relabel de 3:

$d_3 := 1 + \min\{d_1, d_4\} = d_4 + 1 = 4$. Ahora el arco $(3, 4)$ es admisible y se puede empujar $\delta := \min\{e(3) = 1, r_{3,4} = 1\} = 1$ unidades de flujo. Se actualizan el grafo residual y los excesos de los nodos 3 y 4:

Table 10: Iteración 8

i	1	2	3	4	5	6
d_i	6	2	4	3	3	0
$e(i)$	-8	2	0	1	0	5

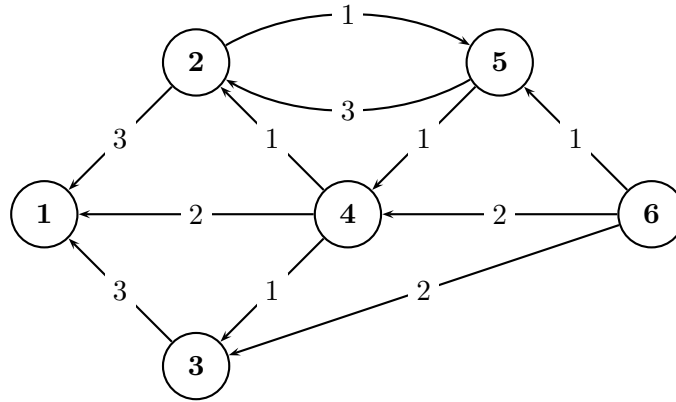


Iteración 9

El nodo 2 tiene exceso $e(2) = 2 > 0$. No hay arco admisible saliendo del nodo, se hace un Relabel de 2: $d_2 := 1 + \min\{d_1, d_5\} = d_5 + 1 = 4$. Ahora el arco $(2, 5)$ es admisible y se puede empujar $\delta := \min\{e(2) = 2, r_{2,5} = 3\} = 2$ unidades de flujo. Se actualizan el grafo residual y los excesos de los nodos 2 y 5:

Table 11: Iteración 9

i	1	2	3	4	5	6
d_i	6	4	4	3	3	0
$e(i)$	-8	0	0	1	2	5

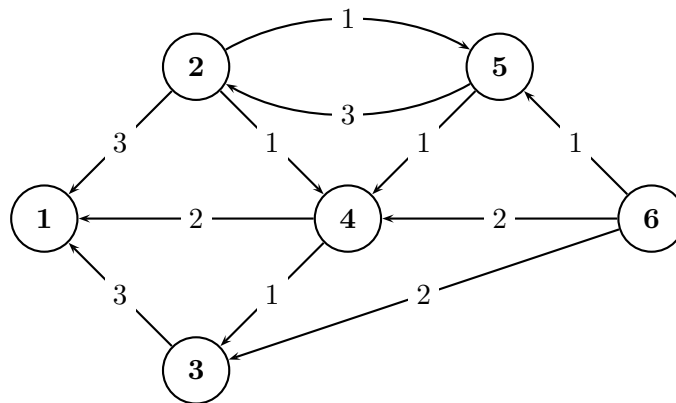


Iteración 10

El nodo 4 tiene exceso $e(4) = 1 > 0$. El arco $(4, 2)$ es admisible y se puede empujar $\delta := \min\{e(4) = 1, r_{4,2} = 1\} = 1$ unidades de flujo. Se actualizan el grafo residual y los excesos de los nodos 4 y 2:

Table 12: Iteración 10

i	1	2	3	4	5	6
d_i	6	4	4	3	3	0
$e(i)$	-8	1	0	0	2	5



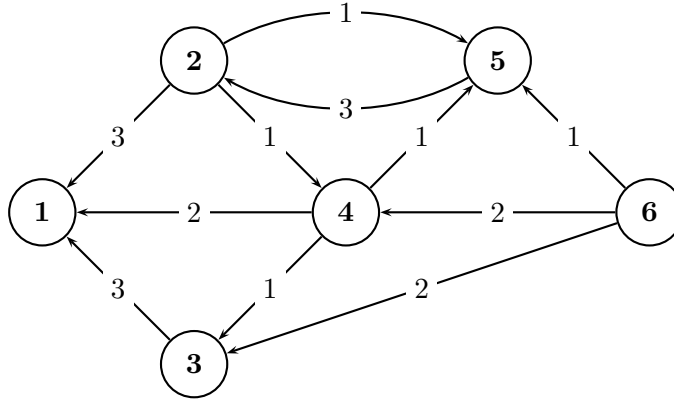
Iteración 11

El nodo 5 tiene exceso $e(5) = 2 > 0$. No hay arco admisible saliendo del nodo, se hace un Relabel de 5: $d_5 := 1 + \min\{d_2, d_4\} = d_4 + 1 = 4$. Ahora el arco $(5, 4)$ es admisible y se puede empujar $\delta := \min\{e(5) =$

$2, r_{5,4} = 1\} = 1$ unidades de flujo. Se actualizan el grafo residual y los excesos de los nodos 5 y 4:

Table 13: Iteración 11

i	1	2	3	4	5	6
d_i	6	4	4	3	4	0
$e(i)$	-8	1	0	1	1	5

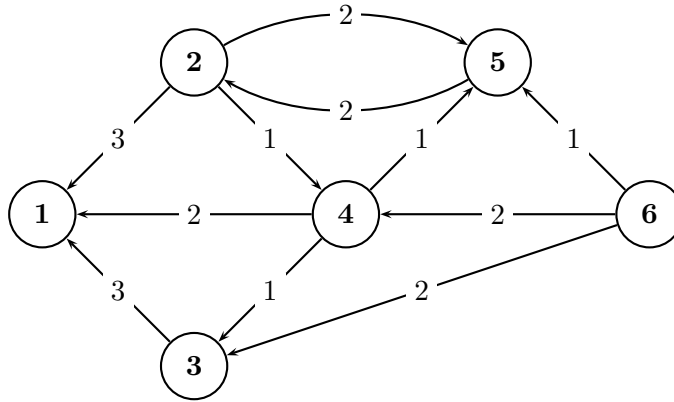


Iteración 12

El nodo 5 tiene exceso $e(5) = 1 > 0$. No hay arco admisible saliendo del nodo, se hace un Relabel de 5: $d_5 := 1 + \min\{d_2\} = d_2 + 1 = 5$. Ahora el arco $(5, 2)$ es admisible y se puede empujar $\delta := \min\{e(5) = 1, r_{5,2} = 3\} = 1$ unidades de flujo. Se actualizan el grafo residual y los excesos de los nodos 5 y 2:

Table 14: Iteración 12

i	1	2	3	4	5	6
d_i	6	4	4	3	5	0
$e(i)$	-8	2	0	1	0	5

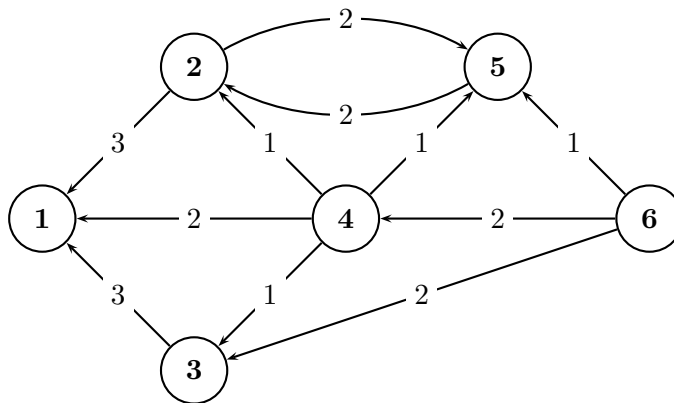


Iteración 13

El nodo 2 tiene exceso $e(2) = 2 > 0$. El arco $(2, 4)$ es admisible y se puede empujar $\delta := \min\{e(2) = 2, r_{2,4} = 1\} = 1$ unidades de flujo. Se actualizan el grafo residual y los excesos de los nodos 2 y 4:

Table 15: Iteración 13

i	1	2	3	4	5	6
d_i	6	4	4	3	5	0
$e(i)$	-8	1	0	2	0	5



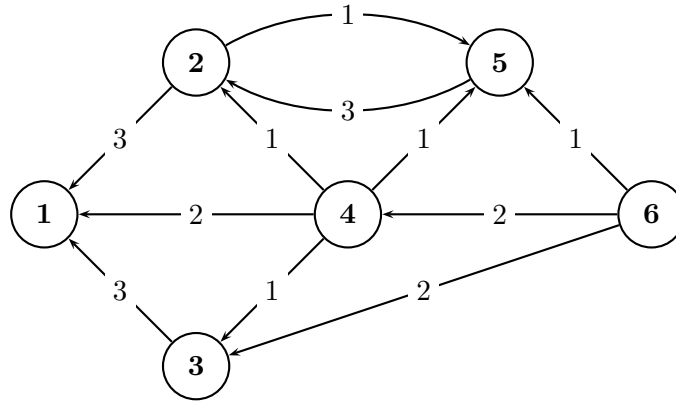
Iteración 14

El nodo 2 tiene exceso $e(2) = 1 > 0$. No hay arco admisible saliendo del nodo, se hace un Relabel de 2: $d_2 := 1 + \min\{d_1, d_5\} = d_5 + 1 = 6$. Ahora el arco $(2, 5)$ es admisible y se puede empujar $\delta := \min\{e(2) =$

$1, r_{2,5} = 2\} = 1$ unidades de flujo. Se actualizan el grafo residual y los excesos de los nodos 2 y 5:

Table 16: Iteración 14

i	1	2	3	4	5	6
d_i	6	6	4	3	5	0
$e(i)$	-8	0	0	2	1	5

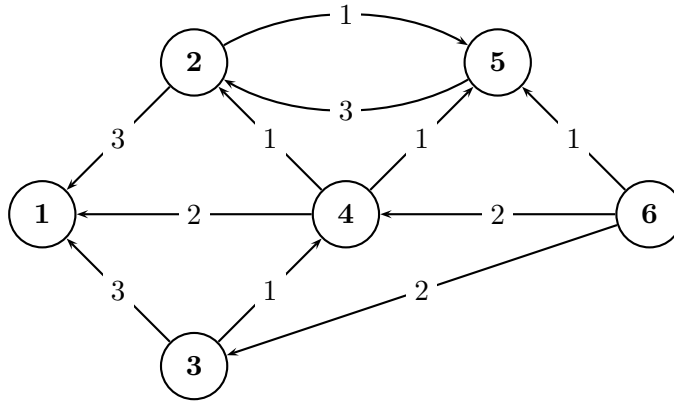


Iteración 15

El nodo 4 tiene exceso $e(4) = 2 > 0$. No hay arco admisible saliendo del nodo, se hace un Relabel de 4: $d_4 := 1 + \min\{d_1, d_2, d_3, d_5\} = d_3 + 1 = 5$. Ahora el arco $(4,3)$ es admisible y se puede empujar $\delta := \min\{e(4) = 2, r_{4,3} = 1\} = 1$ unidades de flujo. Se actualizan el grafo residual y los excesos de los nodos 4 y 3:

Table 17: Iteración 15

i	1	2	3	4	5	6
d_i	6	6	4	5	5	0
$e(i)$	-8	0	1	1	1	5

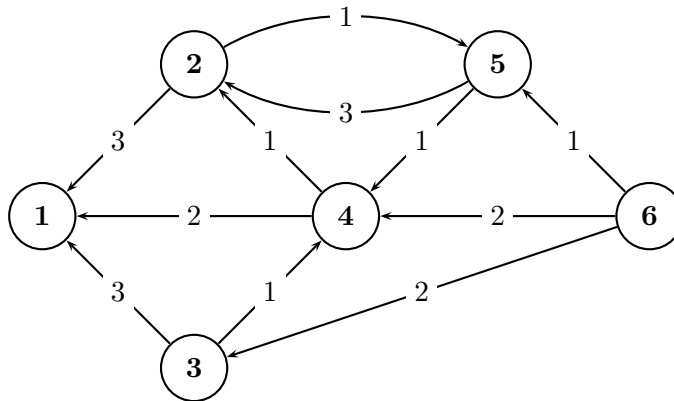


Iteración 16

El nodo 4 tiene exceso $e(4) = 1 > 0$. No hay arco admisible saliendo del nodo, se hace un Relabel de 4: $d_4 := 1 + \min\{d_1, d_2, d_5\} = d_5 + 1 = 6$. Ahora el arco $(4, 5)$ es admisible y se puede empujar $\delta := \min\{e(4) = 1, r_{4,5} = 1\} = 1$ unidades de flujo. Se actualizan el grafo residual y los excesos de los nodos 4 y 5:

Table 18: Iteración 16

i	1	2	3	4	5	6
d_i	6	6	4	6	5	0
$e(i)$	-8	0	1	0	2	5

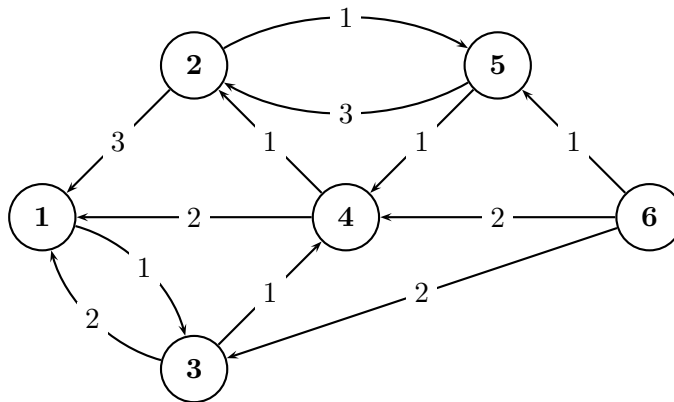


Iteración 17

El nodo 3 tiene exceso $e(3) = 1 > 0$. No hay arco admisible saliendo del nodo, se hace un Relabel de 3: $d_3 := 1 + \min\{d_1, d_4\} = d_1 + 1 = 7$, se nota que la etiqueta paso por encima de $n = 6$, lo que significa que ya no hay caminos hasta el sumidero desde 3 y que tenemos que volver flujo a la fuente. Ahora el arco $(3, 1)$ es admisible y se puede empujar $\delta := \min\{e(3) = 1, r_{3,1} = 3\} = 1$ unidades de flujo. Se actualizan el grafo residual y los excesos de los nodos 3 y 1:

Table 19: Iteración 17

i	1	2	3	4	5	6
d_i	6	6	7	6	5	0
$e(i)$	-7	0	0	0	2	5

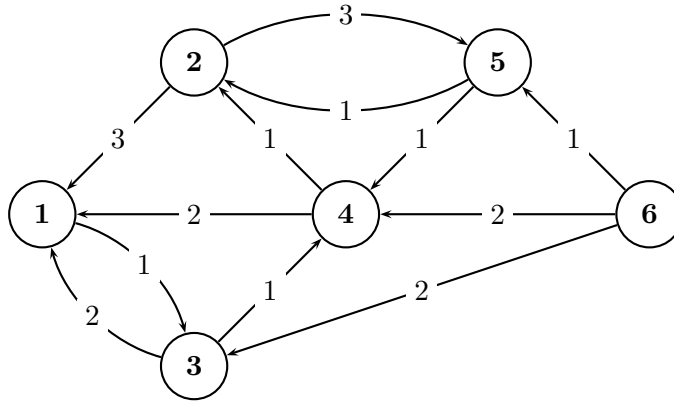


Iteración 18

El nodo 5 tiene exceso $e(5) = 2 > 0$. No hay arco admisible saliendo del nodo, se hace un Relabel de 5: $d_5 := 1 + \min\{d_2, d_4\} = d_2 + 1 = 7$, se nota que la etiqueta paso por encima de $n = 6$, lo que significa que ya no hay caminos hasta el sumidero desde 5 y que tenemos que volver flujo a la fuente. Ahora el arco $(5, 2)$ es admisible y se puede empujar $\delta := \min\{e(5) = 2, r_{5,2} = 3\} = 2$ unidades de flujo. Se actualizan el grafo residual y los excesos de los nodos 5 y 2:

Table 20: Iteración 18

i	1	2	3	4	5	6
d_i	6	6	7	6	7	0
$e(i)$	-7	2	0	0	0	5

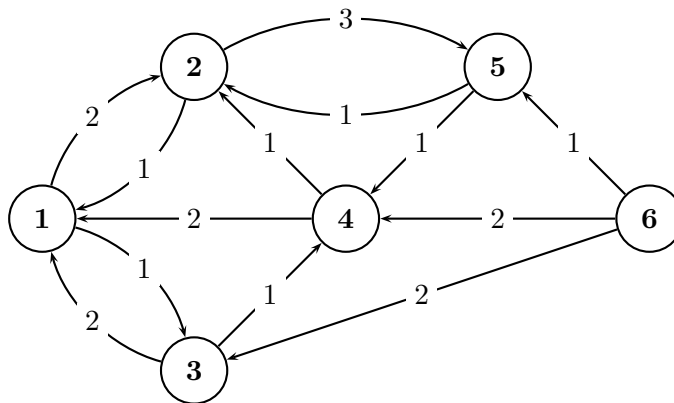


Iteración 19

El nodo 2 tiene exceso $e(2) = 2 > 0$. No hay arco admisible saliendo del nodo, se hace un Relabel de 2: $d_2 := 1 + \min\{d_1, d_5\} = d_1 + 1 = 7$, se nota que la etiqueta paso por encima de $n = 6$, lo que significa que ya no hay caminos hasta el sumidero desde 2 y que tenemos que volver flujo a la fuente. Ahora el arco $(2, 1)$ es admisible y se puede empujar $\delta := \min\{e(2) = 2, r_{2,1} = 3\} = 2$ unidades de flujo. Se actualizan el grafo residual y los excesos de los nodos 2 y 1:

Table 21: Iteración 19

i	1	2	3	4	5	6
d_i	6	7	7	6	7	0
$e(i)$	-5	0	0	0	0	5



Y no hay nodo activo (ie: con $e(i) > 0$) entonces el flujo encontrado es máximo y vale 5, igual que el flujo

encontrado previamente ejecutando el capacity scaling algorithm. Por casualidad, se encontro exactamente el mismo grafo residual, por lo tanto el corte mínimo es lo mismo y tiene capacidad 5.

Ejercicio 7.27

Minimum cut with the fewest number of arcs. Suponemos que queremos saber cual de todos los cortes mínimos contiene el menor número de arcos. Mostrar que si reemplazamos las capacidades u_{ij} por $u'_{ij} = mu_{ij} + 1$, el corte mínimo respecto a las capacidades u'_{ij} es un corte mínimo respecto a las capacidades u_{ij} que contiene el menor número de arcos.

Sean C y C' dos cortes mínimos respecto a las capacidades u_{ij} y $u'_{ij} = mu_{ij} + 1$. Para cualquier corte \bar{C} definamos $U(\bar{C}) = \sum_{(i,j) \in \bar{C}} u_{ij}$ y $U'(\bar{C}) = \sum_{(i,j) \in \bar{C}} u'_{ij}$. Trivialmente tenemos:

$$U'(\bar{C}) = mU(\bar{C}) + |\bar{C}|$$

Probamos que C' es tambien corte mínimo para las capacidades u_{ij} y que es un corte mínimo con el menor número de arcos. Primero tenemos:

$$U(C') = \frac{1}{m}U'(C') - \frac{|C'|}{m}$$

Y por optimalidad de C' se tiene que:

$$U(C') \leq U'(\bar{C}), \forall \bar{C} \text{ corte}$$

Por lo tanto:

$$\begin{aligned} U(C') &\leq \frac{1}{m}U'(\bar{C}) - \frac{|C'|}{m}, \forall \bar{C} \text{ corte} \\ \Rightarrow U(C') &\leq U(\bar{C}) + \frac{|\bar{C}| - |C'|}{m}, \forall \bar{C} \text{ corte} \end{aligned}$$

Además, cualquier corte posee al menos un arco, y a lo más m arcos, por lo tanto $|\bar{C}| - |C'| \leq m - 1$. Entonces:

$$U(C') \leq U(\bar{C}) + \frac{m-1}{m}, \forall \bar{C} \text{ corte}$$

Por integralidad de las capacidades, y porque $0 < 1 - \frac{1}{m} < 1$ tenemos:

$$U(C') \leq U(\bar{C}), \forall \bar{C} \text{ corte}$$

Lo que significa que C' es corte mínimo para las capacidades u_{ij} tambien. Falta probar que tiene el menor número de arcos: de lo anterior sabemos que:

$$U(C') \leq U(\bar{C}) + \frac{|\bar{C}| - |C'|}{m}, \forall \bar{C} \text{ corte}$$

Por optimalidad de C tenemos $U(C) \leq U(C')$, y en particular para $\bar{C} = C$ se tiene que:

$$U(C) \leq U(C) + \frac{|C|-|C'|}{m}$$

ie: $|C| \geq |C'|$

Por lo tanto un corte C' mínimo respecto a las capacidades u'_{ij} tendrá siempre un número de arcos menor o igual que cualquier corte C mínimo respecto a las capacidades u_{ij} .

2 Flujo a costo mínimo

2.1 Repaso

Cycle-cancelling algorithm

$x :=$ flujo factible;

While Existe un ciclo negativo $C \in G(x)$

Identificar un ciclo negativo $C \in G(x)$;

$$\delta := \min_{(ij) \in C} r_{ij};$$

$\forall (i, j) \in P$

$$x_{ij} := x_{ij} + \delta;$$

$$x_{ji} := x_{ji} - \delta;$$

Actualizar $G(x)$;

Successive Shortest Path algorithm

$x := 0; \pi = 0;$

$e(i) := b_i, \forall i \in N;$

$E := \{i : e(i) > 0\};$

$D := \{i : e(i) < 0\};$

While $E \neq \emptyset$

Seleccionar un par $(k, l) \in E \times D;$

Calcular las distancias más cortas d_i de k a i con los costos $c_{ij}^\pi;$

$P :=$ camino más corto de k a $t;$

$$\pi := \pi - d;$$

$$\delta := \min \left[e(k), -e(l), \min_{(i,j) \in P} r_{ij} \right];$$

$\forall (i, j) \in P$

$$x_{ij} := x_{ij} + \delta;$$

$$x_{ji} := x_{ji} - \delta;$$

Actualizar $G(x), E, D$ y los $c_{ij}^\pi;$

Primal-Dual algorithm

$x := 0; \pi = 0;$

$e(s) := b_s;$

$e(t) := b_t;$

While $e(s) > 0$

 Calcular las distancias más cortas d_i de s a i con los costos c_{ij}^π ;

$\pi := \pi - d;$

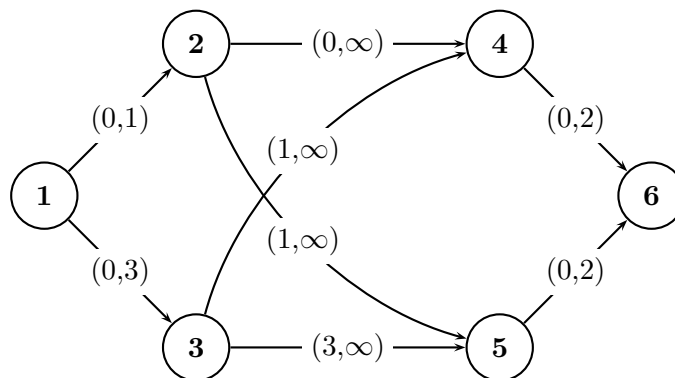
 Definir el grafo admisible $G^o(x)$;

 Calcular un flujo máximo de s a t en $G^o(x)$;

 Actualizar $e(s)$, $e(t)$, $G(x)$ y los c_{ij}^π ;

2.2 Ejercicio

Ocupar los algoritmos de Successive Shortest Path Algorithm y de Primal-Dual para encontrar un flujo a costo mínimo en el siguiente grafo:



Con $b_1 = 4$, $b_6 = -4$ y $b_i = 0 \forall i \neq 1, i \neq 6$

Successive Shortest Path algorithm

$\pi = 0; x := 0;$

Iteración 1:

Seleccionamos el par $(1, 6)$, los potenciales valen cero durante la primera iteración entonces $c_{ij}^\pi = c_{ij}$.

Calculamos las distancias más cortas de 1 a todos:

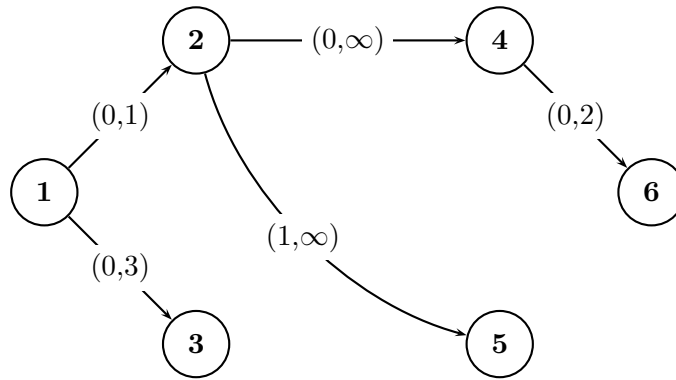


Table 22: Distancias Iteración 1

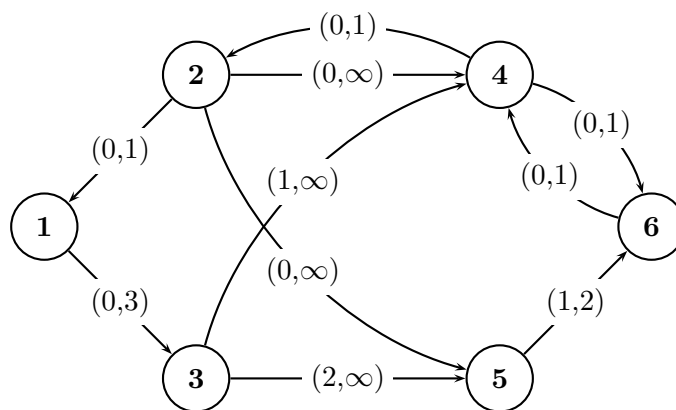
i	1	2	3	4	5	6
d_i	0	0	0	0	1	0

Actualizamos los potenciales:

Table 23: Potenciales Iteración 1

i	1	2	3	4	5	6
π_i	0	0	0	0	-1	0

$\{1, 2, 4, 6\}$ es un camino más corto de 1 a 6, y se puede empujar $\delta := \min\{e(1), -e(6), r_{1,2}, r_{2,4}, r_{4,6}\} = 1$ unidades de flujo. Actualizamos el grafo residual (costos residuales $c_{ij}^\pi = c_{ij} - \pi_i + \pi_j$ y capacidades residuales r_{ij}) y las demandas/ofertas de 1 y 6:



Iteración 2:

Seleccionamos el par (1,6). Calculamos las distancias más cortas de 1 a todos con los precios c_{ij}^π :

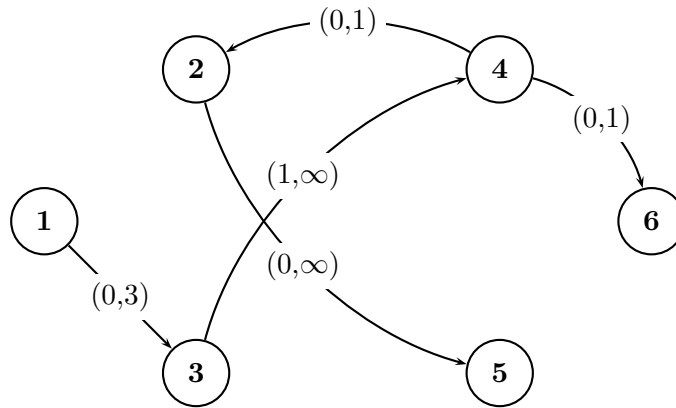


Table 24: Distancias Iteración 2

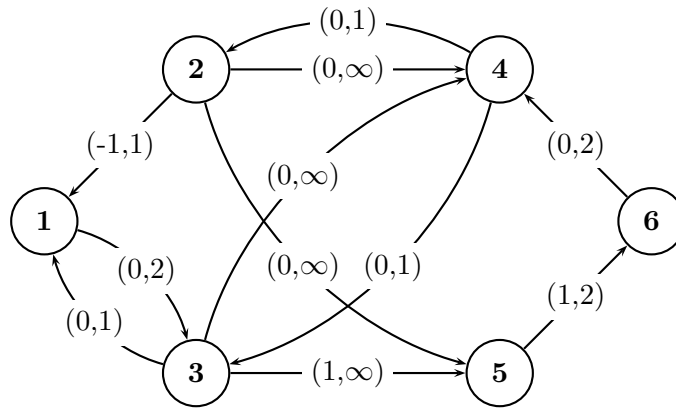
i	1	2	3	4	5	6
d_i	0	1	0	1	1	1

Actualizamos los potenciales:

Table 25: Potenciales Iteración 2

i	1	2	3	4	5	6
π_i	0	-1	0	-1	-2	-1

$\{1, 3, 4, 6\}$ es un camino más corto de 1 a 6, y se puede empujar $\delta := \min\{e(1), -e(6), r_{1,3}, r_{3,4}, r_{4,6}\} = 1$ unidades de flujo. Actualizemos el grafo residual (costos y capacidades) y las demandas/ofertas de 1 y 6:



Iteración 3:

Seleccionamos el par (1,6). Calculamos las distancias más cortas de 1 a todos con los precios c_{ij}^π :

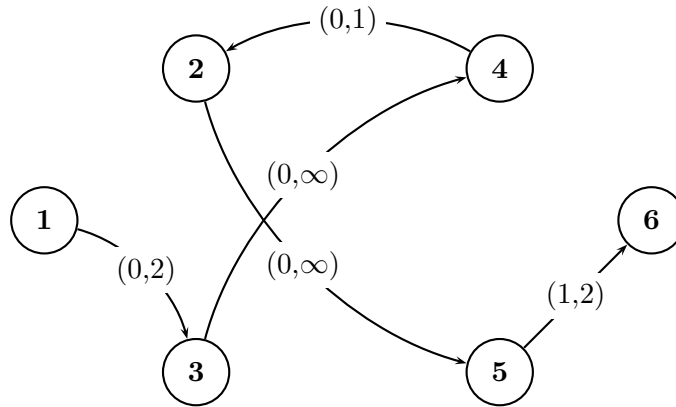


Table 26: Distancias Iteración 3

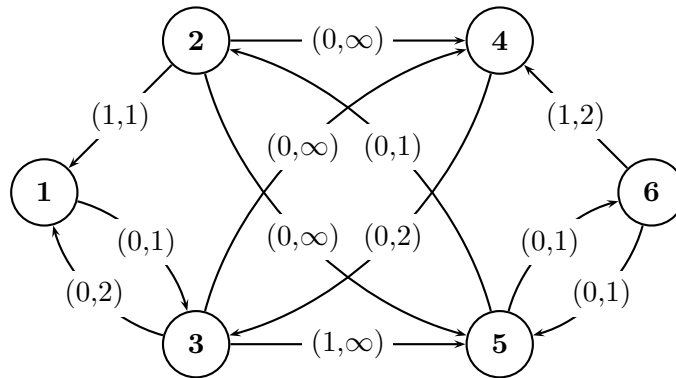
i	1	2	3	4	5	6
d_i	0	0	0	0	0	1

Actualizamos los potenciales:

Table 27: Potenciales Iteración 3

i	1	2	3	4	5	6
π_i	0	-1	0	-1	-2	-2

$\{1, 3, 4, 2, 5, 6\}$ es un camino más corto, y se puede empujar $\delta := \min\{e(1), -e(6), r_{1,3}, r_{3,4}, r_{4,2}, r_{2,5}, r_{5,6}\} = 1$ unidades de flujo. Actualizemos el grafo residual (costos y capacidades) y las demandas/ofertas de 1 y 6:



Iteración 4:

Seleccionamos el par (1,6). Calculamos las distancias más cortas de 1 a todos con los precios c_{ij}^π :

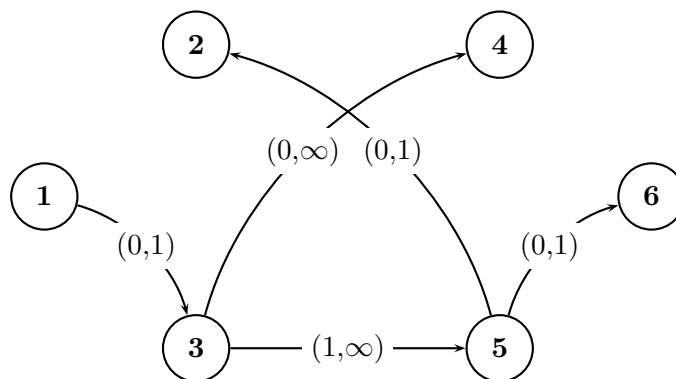


Table 28: Distancias Iteración 4

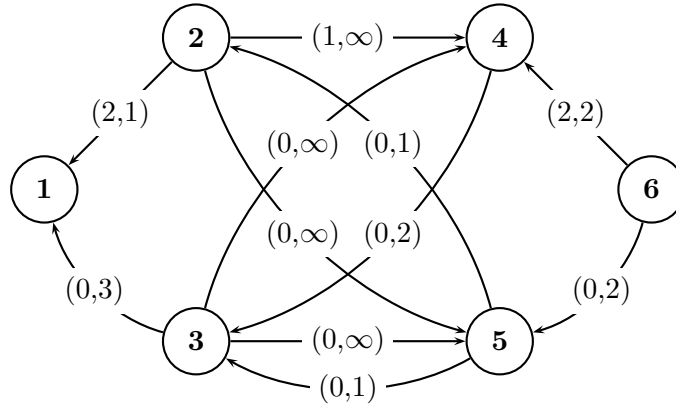
i	1	2	3	4	5	6
d_i	0	1	0	0	1	1

Actualizamos los potenciales:

$\{1, 3, 5, 6\}$ es un camino más corto de 1 a 6, y se puede empujar $\delta := \min\{e(1), -e(6), r_{1,3}, r_{3,5}, r_{5,6}\} = 1$ unidades de flujo. Actualizemos el grafo residual (costos y capacidades) y las demandas/ofertas de 1 y 6:

Table 29: Potenciales Iteración 4

i	1	2	3	4	5	6
π_i	0	-2	0	-1	-3	-3



Ya no hay nodos activos ($\nexists i : e(i) \neq 0$) entonces tenemos un flujo factible. Por seguridad verificamos que los costos reducidos y el flujo encontrados cumplen con las condiciones de optimalidad:

- Si $c_{ij}^\pi > 0$ entonces $x_{ij}^* = 0$
- Si $c_{ij}^\pi < 0$ entonces $x_{ij}^* = u_{ij}$
- Si $c_{ij}^\pi = 0$ entonces $0 \leq x_{ij}^* \leq u_{ij}$

En el proximo grafo para cada arco aparecen entre corchetes el flujo, la capacidad y el costo reducido: $[x_{ij}^*, u_{ij}, c_{ij}^\pi]$. El flujo encontrado es el siguiente:

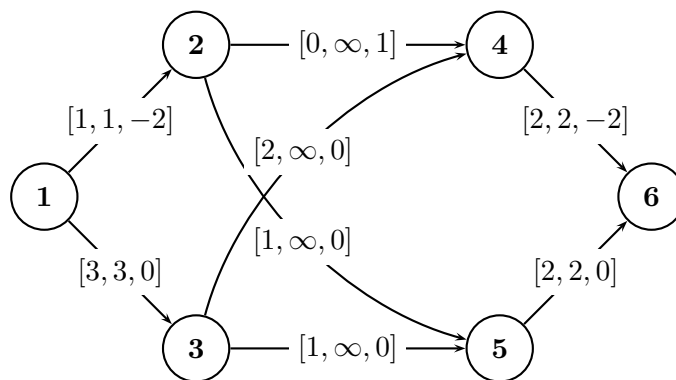


Table 30: Optimalidad

(i, j)	c_{ij}^π	x_{ij}^*	u_{ij}	Criterio
(1, 2)	-2	1	1	$c_{1,2}^\pi < 0$ y $x_{1,2}^* = 1 = u_{1,2}$
(1, 3)	0	3	3	$c_{1,3}^\pi = 0$ y $0 \leq x_{1,3}^* = 3 \leq u_{1,3} = 3$
(2, 4)	1	0	∞	$c_{2,4}^\pi > 0$ y $x_{2,4}^* = 0$
(2, 5)	0	1	∞	$c_{2,5}^\pi = 0$ y $0 \leq x_{2,5}^* = 1 \leq u_{2,5} = \infty$
(3, 4)	0	2	∞	$c_{3,4}^\pi = 0$ y $0 \leq x_{3,4}^* = 2 \leq u_{3,4} = \infty$
(3, 5)	0	1	∞	$c_{3,5}^\pi = 0$ y $0 \leq x_{3,5}^* = 1 \leq u_{3,5} = \infty$
(4, 6)	-2	2	2	$c_{4,6}^\pi < 0$ y $x_{4,6}^* = 2 = u_{4,6}$
(5, 6)	0	2	2	$c_{5,6}^\pi = 0$ y $0 \leq x_{5,6}^* = 2 \leq u_{5,6} = 2$

Los costos reducidos y el flujo factible encontrado cumplen con las condiciones de holgura complementaria, por lo tanto el flujo encontrado es de costo mínimo.