

Problema de máximo flujo

Javier Navarro

Historia

Flujo máximo

Corte mínimo

Soviet Railway
system

US

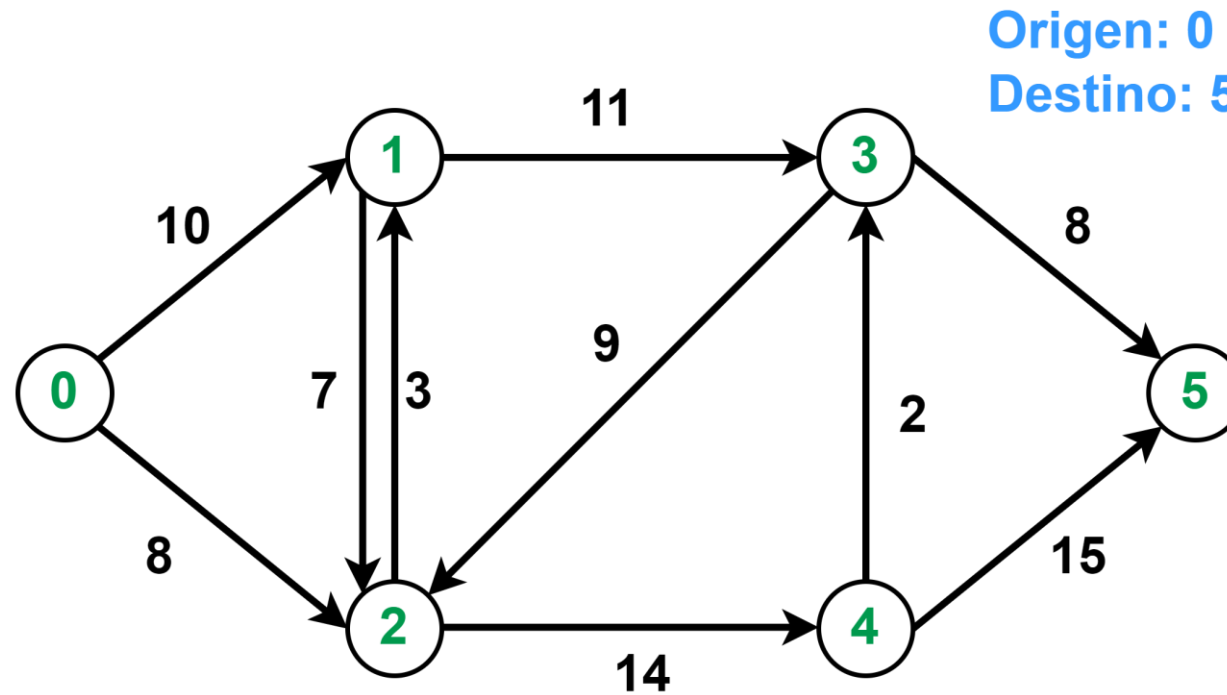
Recursos y
personas



Вадим Анохин - Own work
VL80T-831 electric locomotive hauling a train near Mokry Chaltyr station
•[CC BY-SA 3.0](#)
•File:Electric locomotive VL80T-831.jpg
•Uploaded: 9 August 2009

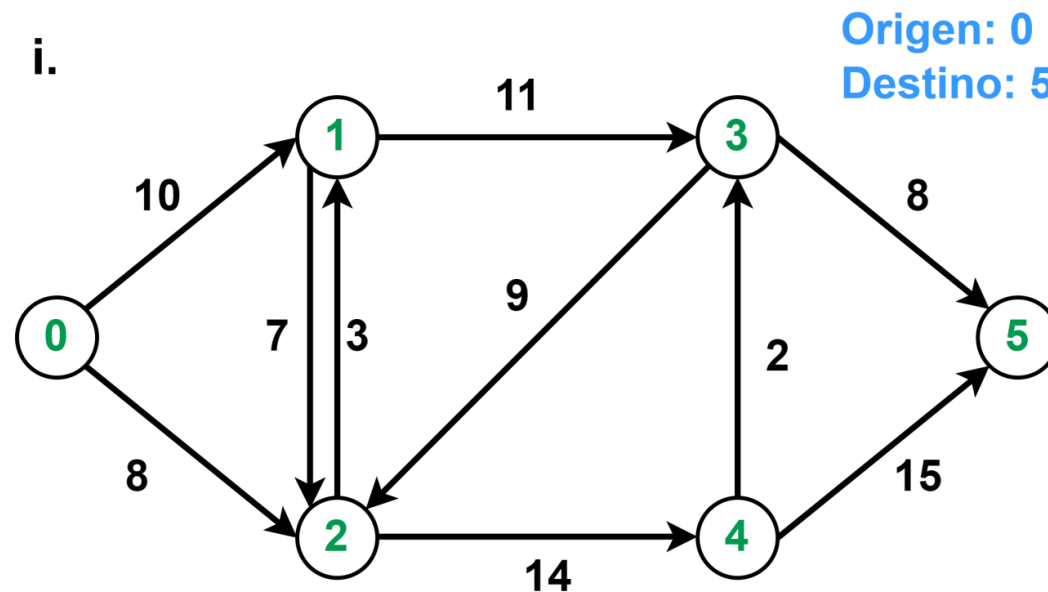


Problema de flujo máximo



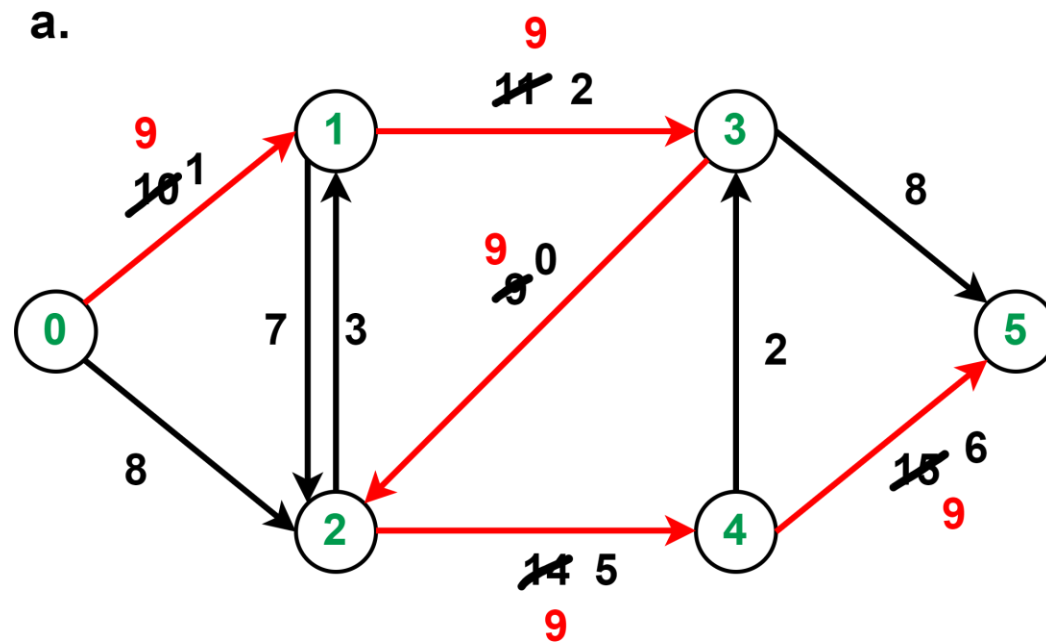
Inundando (flooding)

- Mandar el máximo flujo a todos lados (greedy) y eliminar el exedente.



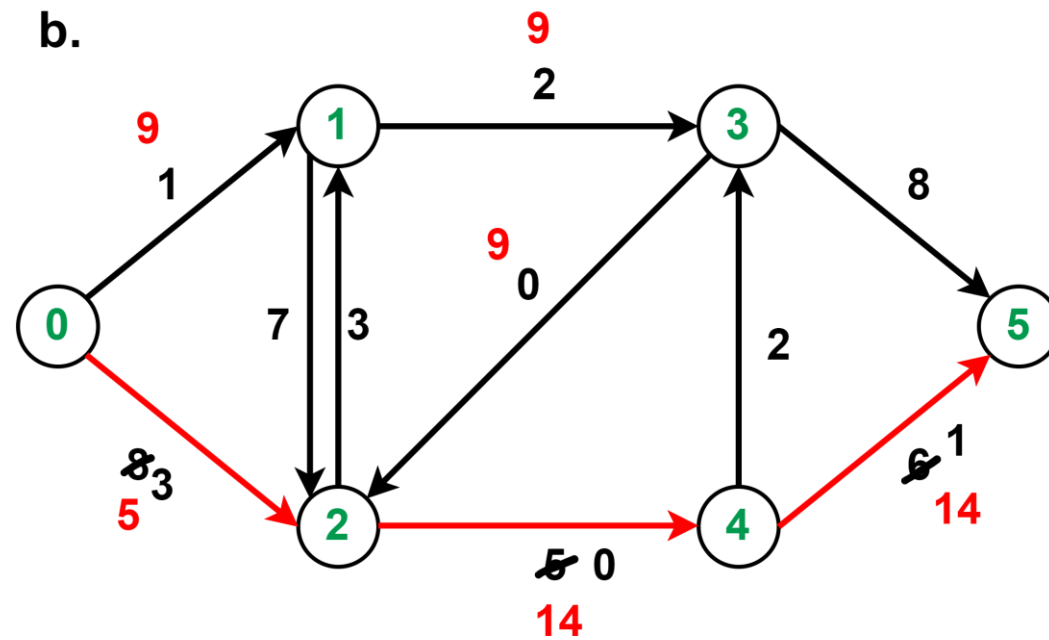
Inundando (flooding)

- Mandar el máximo flujo a todos lados (greedy) y eliminar el excedente.



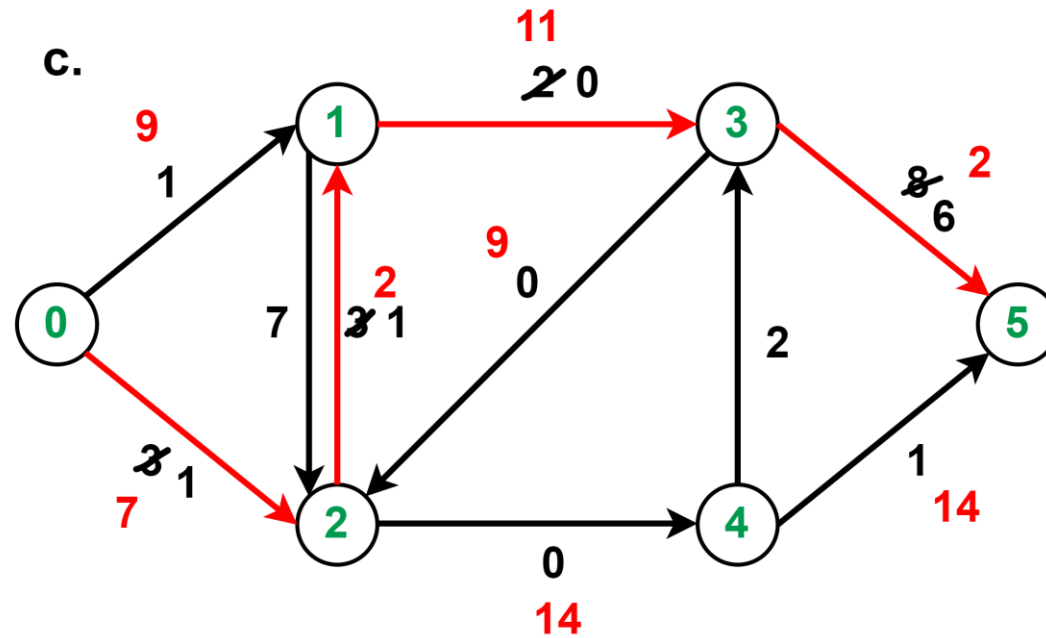
Inundando (flooding)

- Mandar el máximo flujo a todos lados (greedy) y eliminar el excedente.



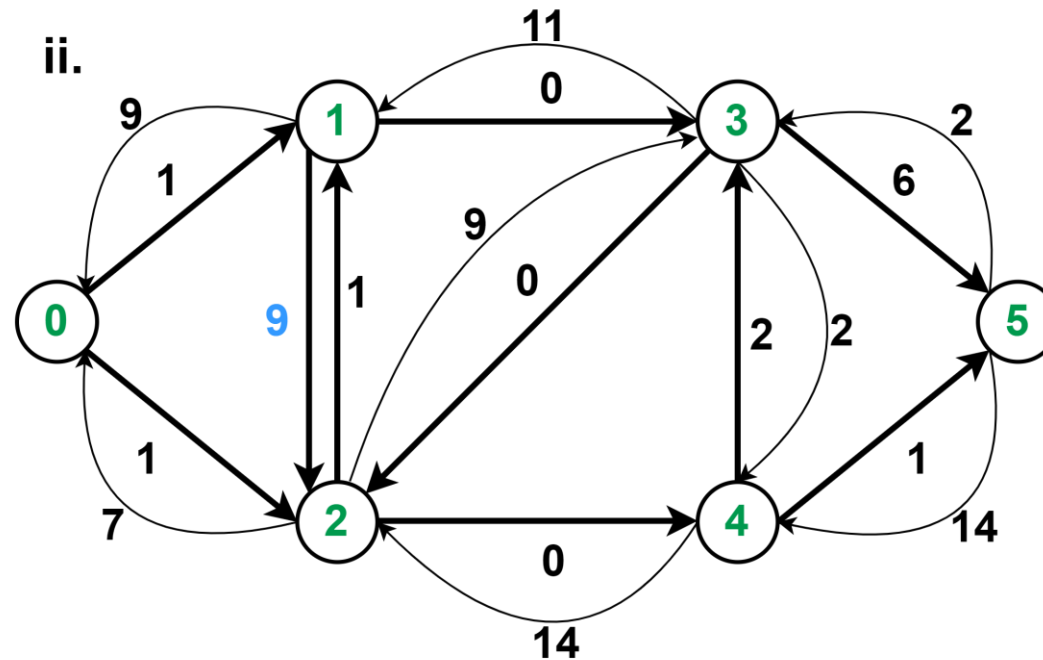
Inundando (flooding)

- Mandar el máximo flujo a todos lados (greedy) y eliminar el excedente.



Ford-Fulkerson (grafo residual)

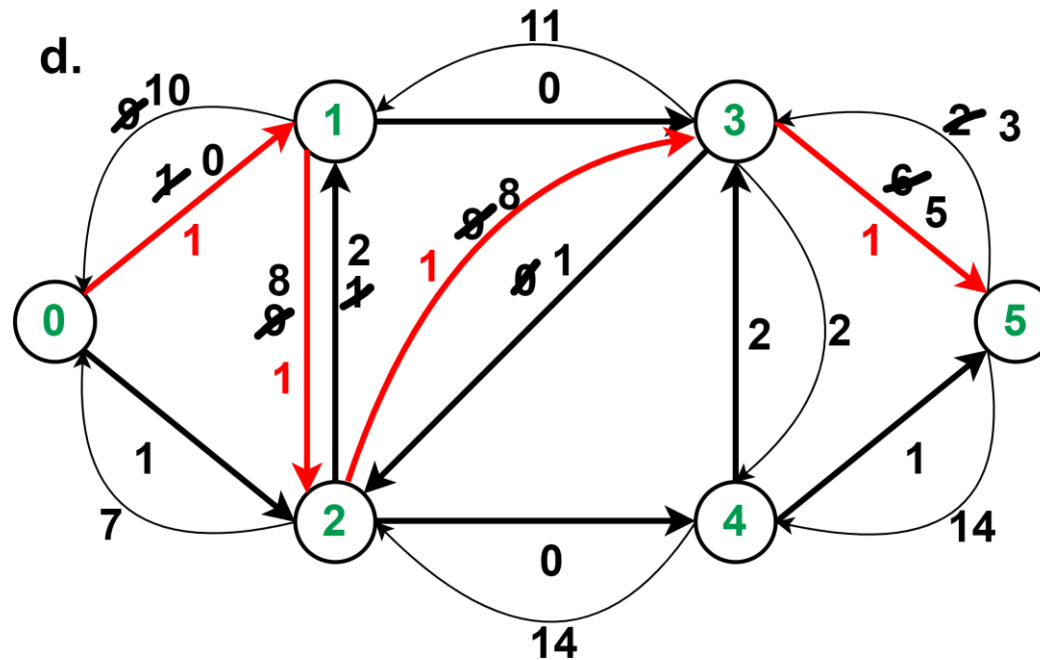
- Permite regresar (ctrl-z) para hacer un flujo mejor. Extensión del algoritmo greedy anterior.



*Se añaden aristas en sentido contrario donde falten. Entre 1 y 2 ya había 2 aristas, entonces ahí ya no se agrega.

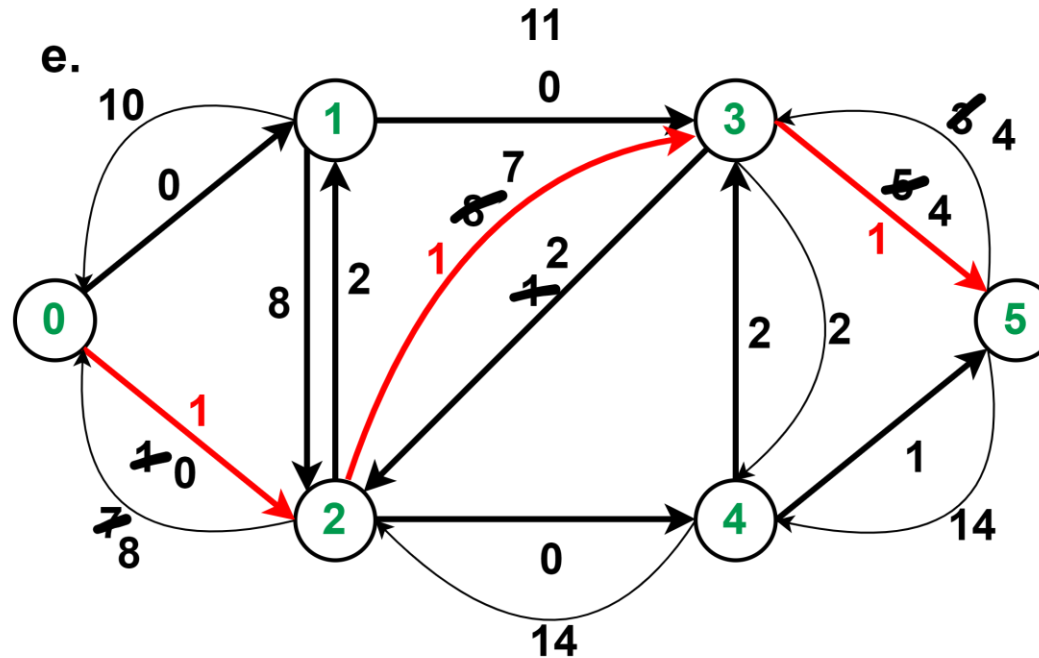
Grafo residual

- Permite regresar (ctrl-z) para hacer un flujo mejor. Extensión del algoritmo greedy anterior.



Grafo residual

- Permite regresar (ctrl-z) para hacer un flujo mejor. Extensión del algoritmo greedy anterior.



Flujo máximo = 18

Grafo residual

- Pseudocódigo

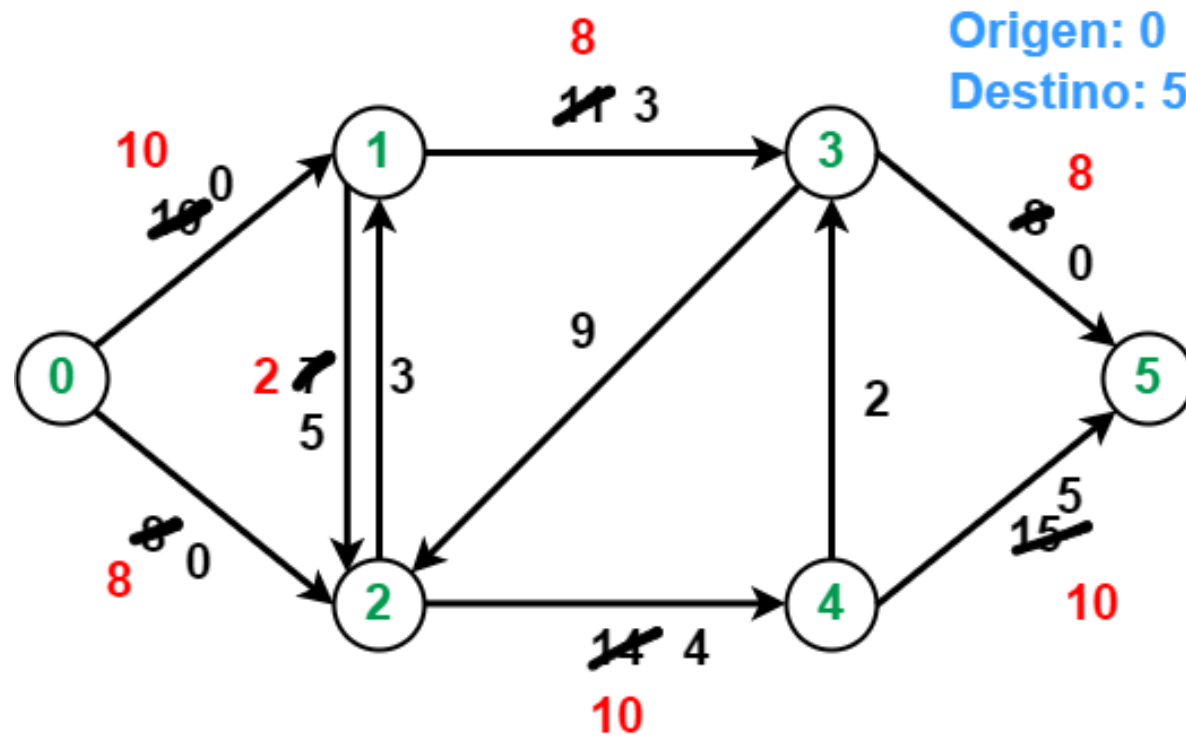
complejidad = $O(Ef)$

Algoritmo 1 Ford-Fulkerson

```
Grafo_original ← copy(Grafo)  
camino_posible ← busq_anchura(Origen, Destino, Grafo)  
while camino_posible ≠ False do  
    camino_posible ← busq_anchura(Origen, Destino, Grafo)  
    actualizar_grafo(Grafo, camino_posible)  
end while  
flujo ← contar_flujo(Grafo, Grafo_original, Origen)
```

Grafo residual

- Solución



Flujo máximo = 18

PPL

Notación

C_{ij} = capacidad de arista de i a j .

$$C_{13} = 11$$

$$C_{31} = 0$$

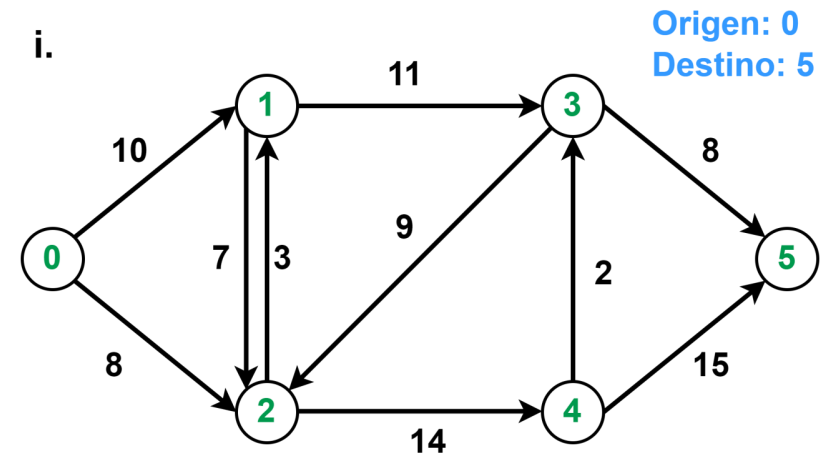
X_{ij} = flujo de arista de i a j

$$X_{35} = 2$$

$$X_{53} = 0$$

E siendo el set de aristas

n = # vertices



PPL

Problema planteado

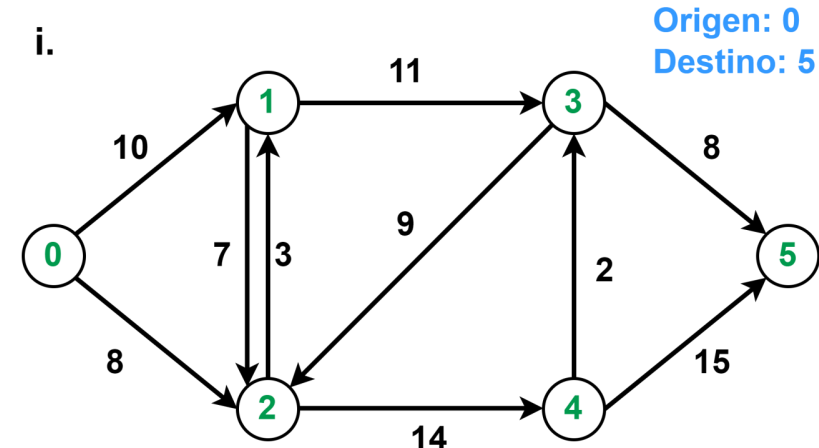
$$\text{Min } c^T x \text{ sujeta a } Ax \leq b, x \geq 0$$

donde :

$$x^T = [f \ x_{01} \ x_{02} \ x_{12} \ x_{13} \ x_{21} \ x_{24} \ x_{32} \ x_{35} \ x_{43} \ x_{45}]$$

$$c^T = [0 \ -1 \ -1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]$$

$$b^T = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 10 \ 8 \ 7 \ 11 \ 3 \ 14 \ 9 \ 8 \ 2 \ 15]$$



$$A = \left[\begin{array}{c|cccccccccccc} -1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 & 1 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 \\ \hline 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array} \right]$$

Para ejemplificar, la segunda restricción equivale a :

$$-x_{01} + x_{12} + x_{13} - x_{21} \leq 0$$

La última restricción es:

$$x_{45} \leq 15$$

PPL

Pseudocódigo

Algoritmo 2 SIMPLEX

$A \leftarrow \text{matriz}$

while $\text{condicion_paro}(A) = \text{False}$ **do**

$\text{pivote_i}, \text{pivote_j} \leftarrow \text{sacar_pivote}(A)$

$\text{operaciones_elementales}(\text{pivote_i}, \text{pivote_j}, A)$

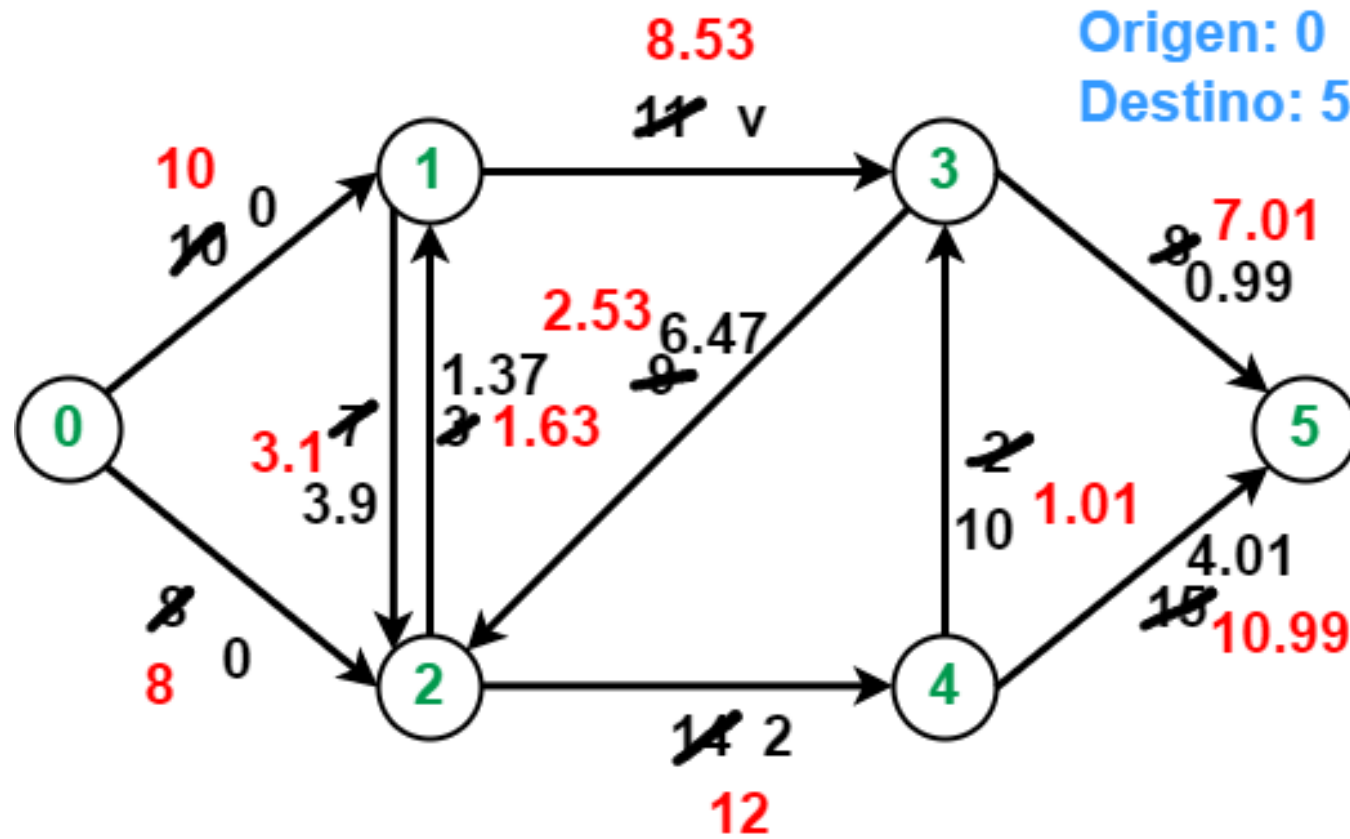
end while

$\text{valores_de_variables} \leftarrow A[-1, : -1]$

$f \leftarrow A[-1, -1]$

PPL

Solución



Conclusiones

- Enteros en la solución
- Número de aristas utilizadas
- Complejidad
- Interpretabilidad

Problema dual

En el problema dual se maximizan las capacidades de cada arista. Las variables de decisión son:

1. La “capacidad” de cada nodo.
2. La capacidad de cada arista.

x es el vector del flujo de los aristas.

Aplicaciones

- Energía eléctrica
- Planificación de vuelos
- Redes de telecomunicaciones
- Ciencias de la computación
- Bombardeo de redes ferroviarias

- http://www.pvamu.edu/mathematics/wp-content/uploads/sites/49/33_R1095_Ram_AAM_V13_1_pp_508_515_060118.pdf
- <https://homepages.cwi.nl/~lex/files/histtrpclean.pdf>
- <https://www.geeksforgeeks.org/max-flow-problem-introduction/#:~:text=Maximum%20flow%20problems%20involve%20finding,stuff%20that%20it%20can%20carry.>
- <https://www.geeksforgeeks.org/ford-fulkerson-algorithm-for-maximum-flow-problem/>
- <http://theory.stanford.edu/~tim/w16/l/l1.pdf>
- https://www.ripublication.com/irph/ijert20/ijertv13n7_06.pdf
- <https://youtu.be/MFKVx-8rf30>