Melhores momentos

AULA PASSADA

Ford e Fulkerson

Método dos caminhos de incremento

> PASSO DE INCREMENTO: encontre um caminho de incremento para o fluxo corrente. Incremente o valor do fluxo “enviando a maior quantidade possível de fluxo através do caminho”.

Note que o método não especifica como encontrar o pseudo-caminho de incremento.

Incrementos máximos

Método dos incrementos máximos (Edmonds e Karp ’72)

PASSO DE INCREMENTO: encontre na rede residual (do fluxo corrente) um caminho de

capacidade residual máxima.

Incremente o valor do fluxo “enviando a maior quantidade possível de fluxo através desse caminho”.

Capacity-scaling

O algoritmo CAPACITY-SCALING (Edmonds e Karp ’72) é uma variação do MAX-CAPACITY.

Idéia: em cada iteração encontrar um caminho de incremento de capacidade residual “grande”.

Em cada iteração do algoritmo temos um pararâmetro Δ e encontramos uma caminho de incremento de capacidade residual $\geq \Delta$.

Vantagem: é mais fácil de implementar e mais eficiente.

Resumo

Invariante: no início de cada iteração temos um st-pseudofluxo $\check{x}$ que respeita as capacidades.

Algoritmo	número de passos de incremento	consumo de tempo
FORD-FULKERSON	$\mathrm{O}(nU)$	$\mathrm{O}(nU(m+n))$
MAX-CAPACITY	$\leq 2m(1+\lfloor \lg U \rfloor)$ $\mathrm{O}(m \lg U)$	$\mathrm{O}(n^2 m \lg U)$
CAPACITY-SCALING	$\leq 2m(1+\lfloor \lg U \rfloor)$ $\mathrm{O}(m \lg U)$	$\mathrm{O}(m^2 \lg U)$

Estamos supondo que $n = \mathrm{O}(m)$ (grafo conexo)

AULA 13

Edmonds-Karp

PF 15.1, 15.2, 15.3

Edmonds e Karp

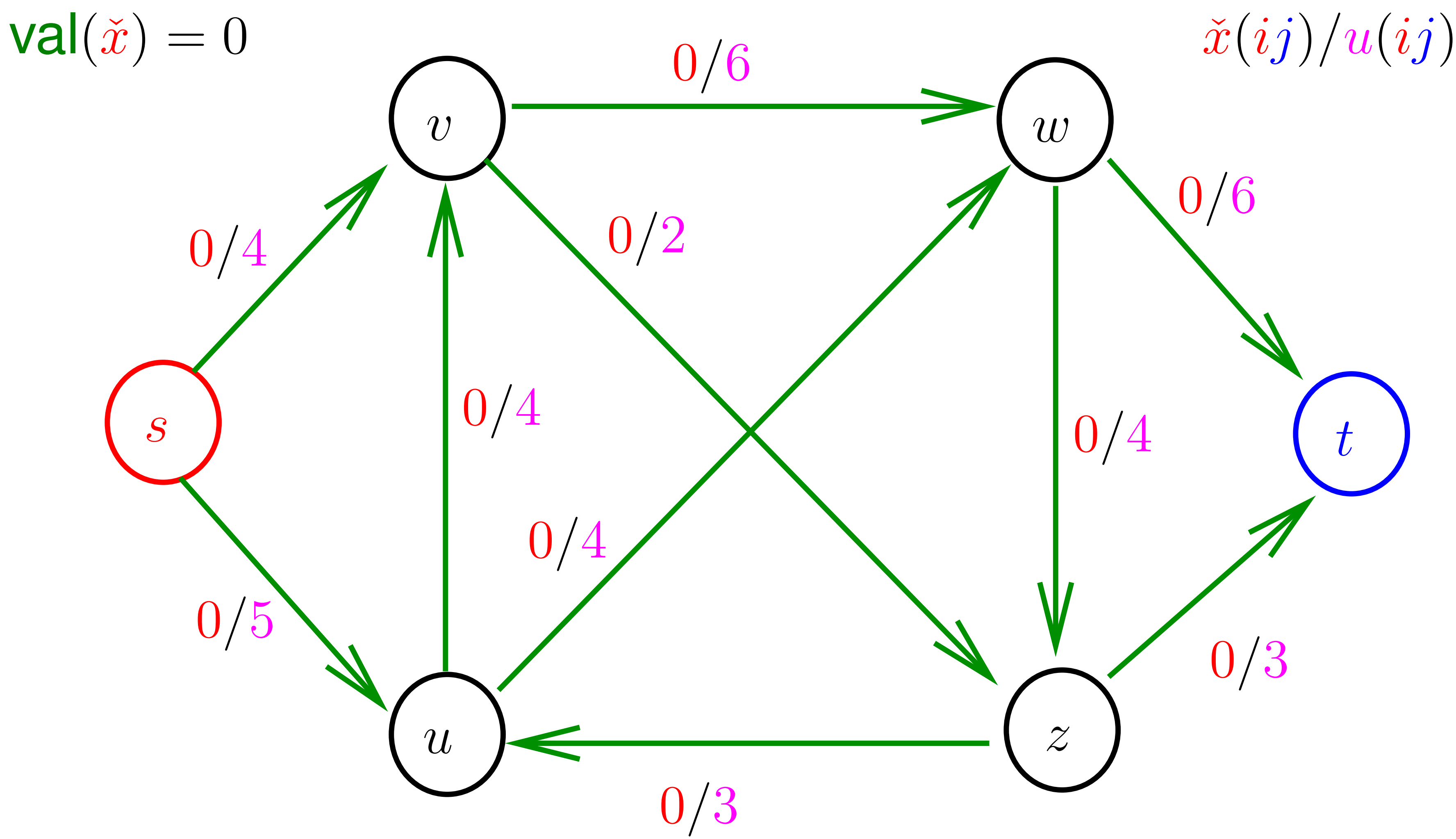
Método dos incrementos através de caminhos “curtos”

PASSO DE INCREMENTO: encontre na rede residual (do fluxo corrente) um caminho de

comprimento mínimo.

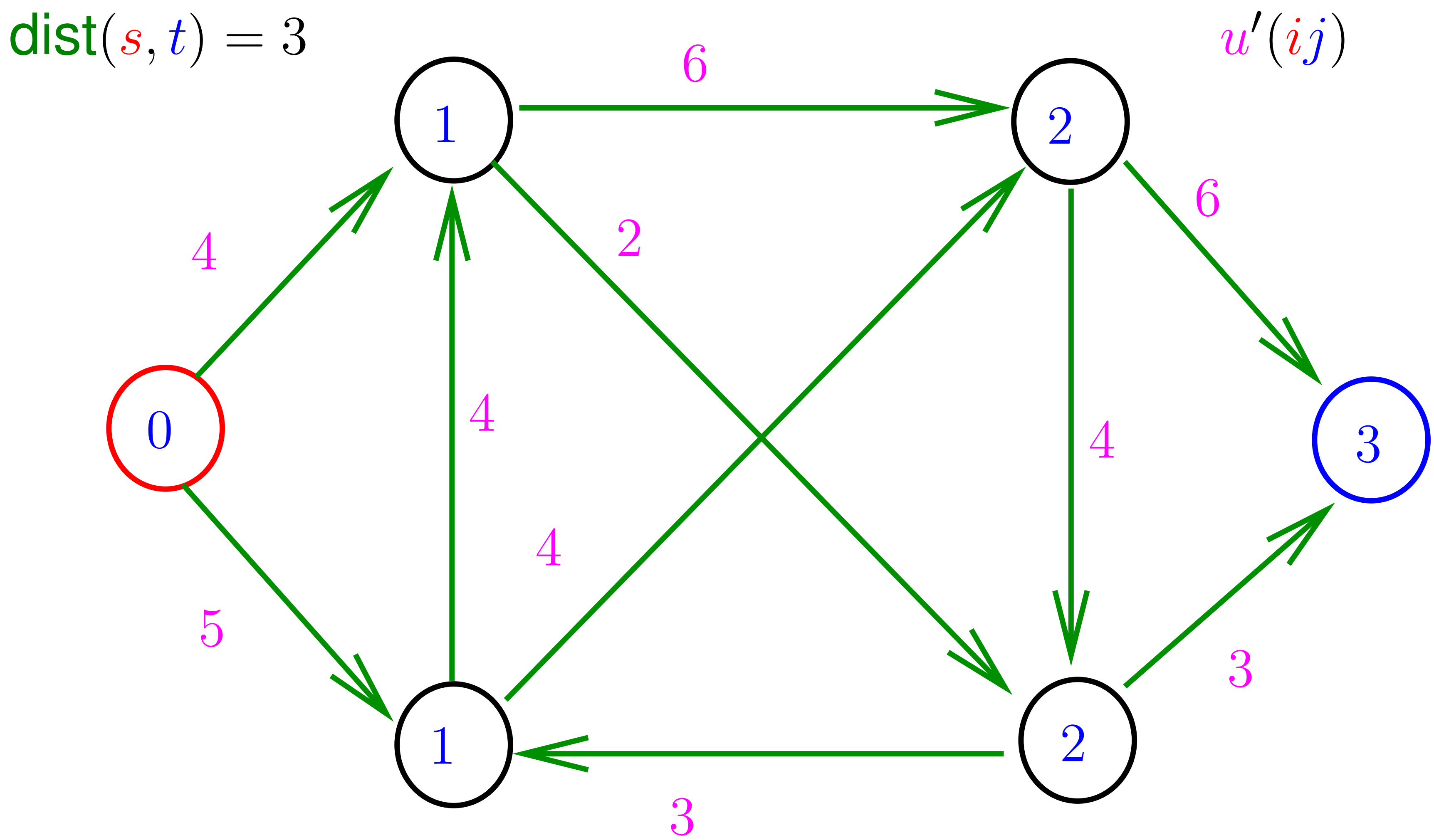
Incremente o valor do fluxo “enviando a maior quantidade possível de fluxo possível através desse caminho”.

Rede 1



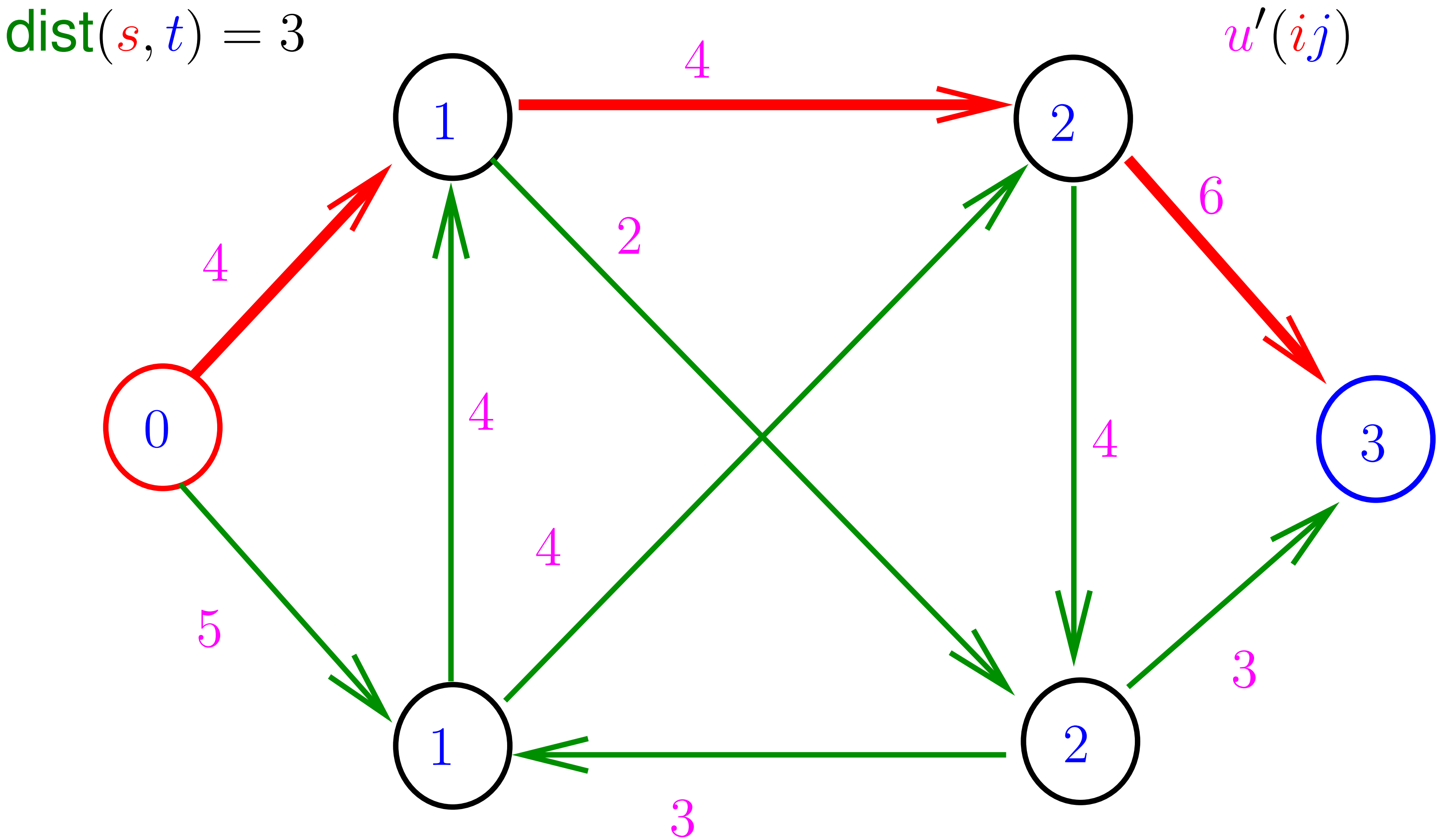
Arcos ausentes têm capacidade nula.

Rede residual 1



Arcos ausentes têm capacidade nula.

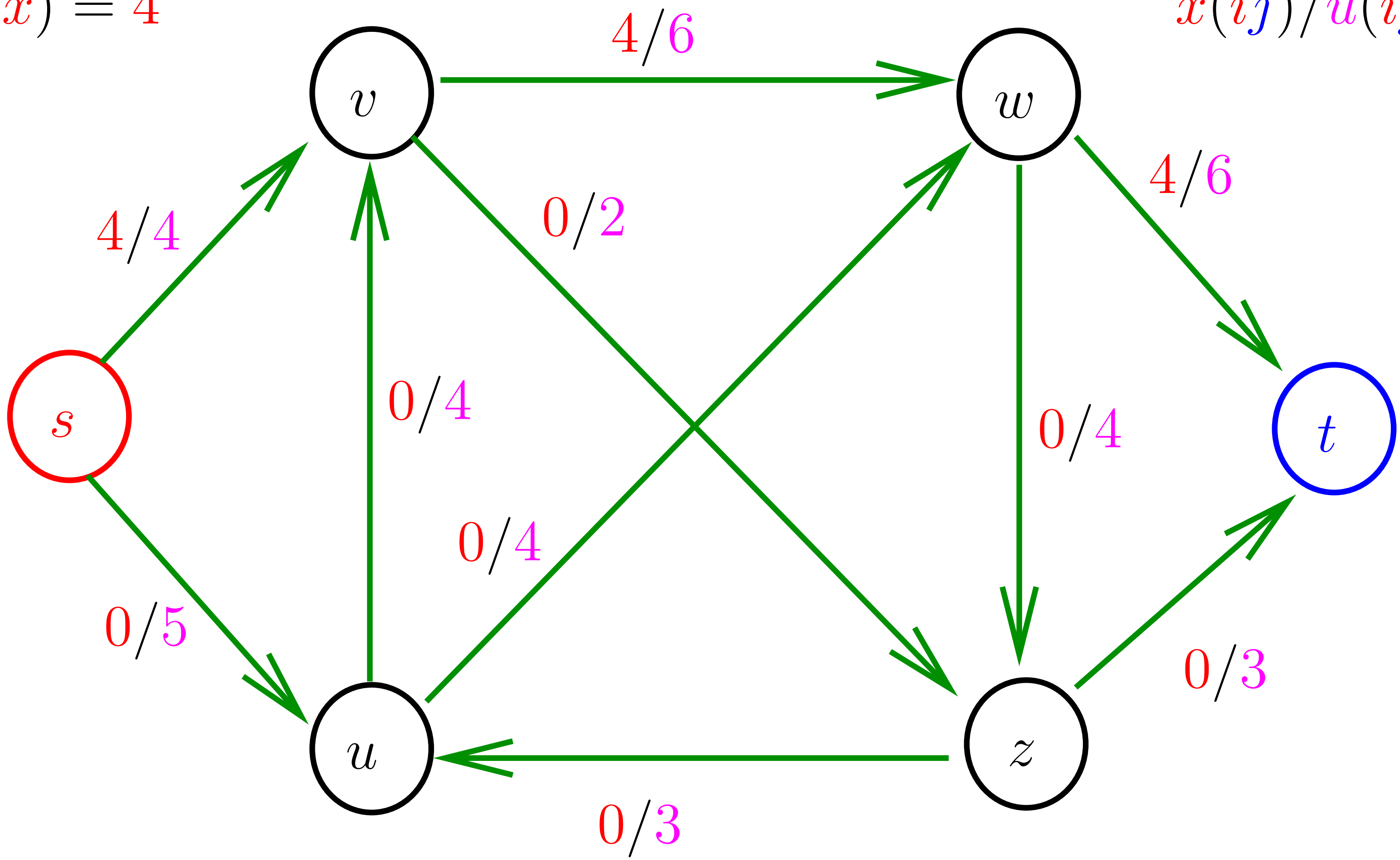
Caminho de incremento 1



Rede 2

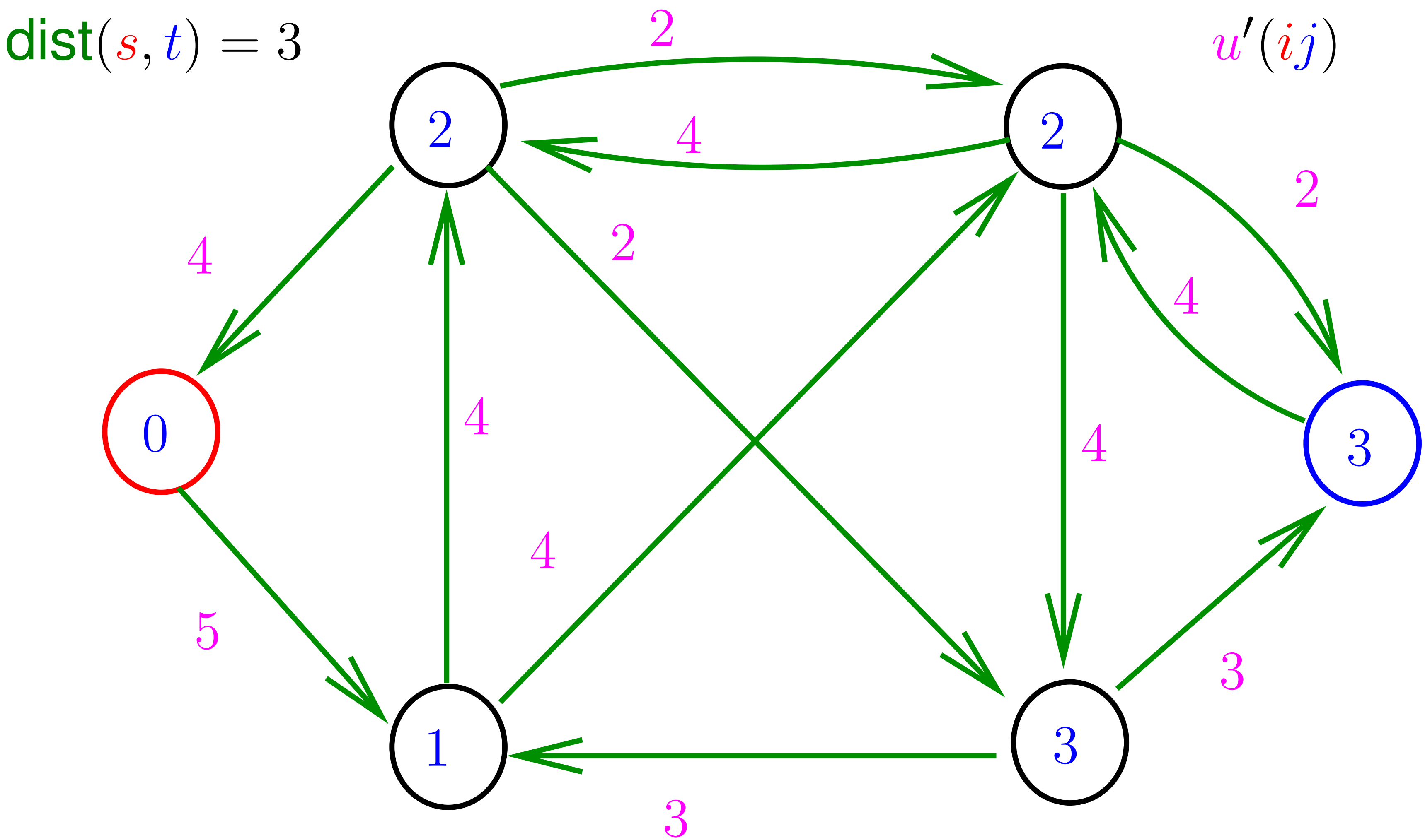
$\mathsf{val}(\check{x}) = 4$

$\check{x}(ij)/u(ij)$



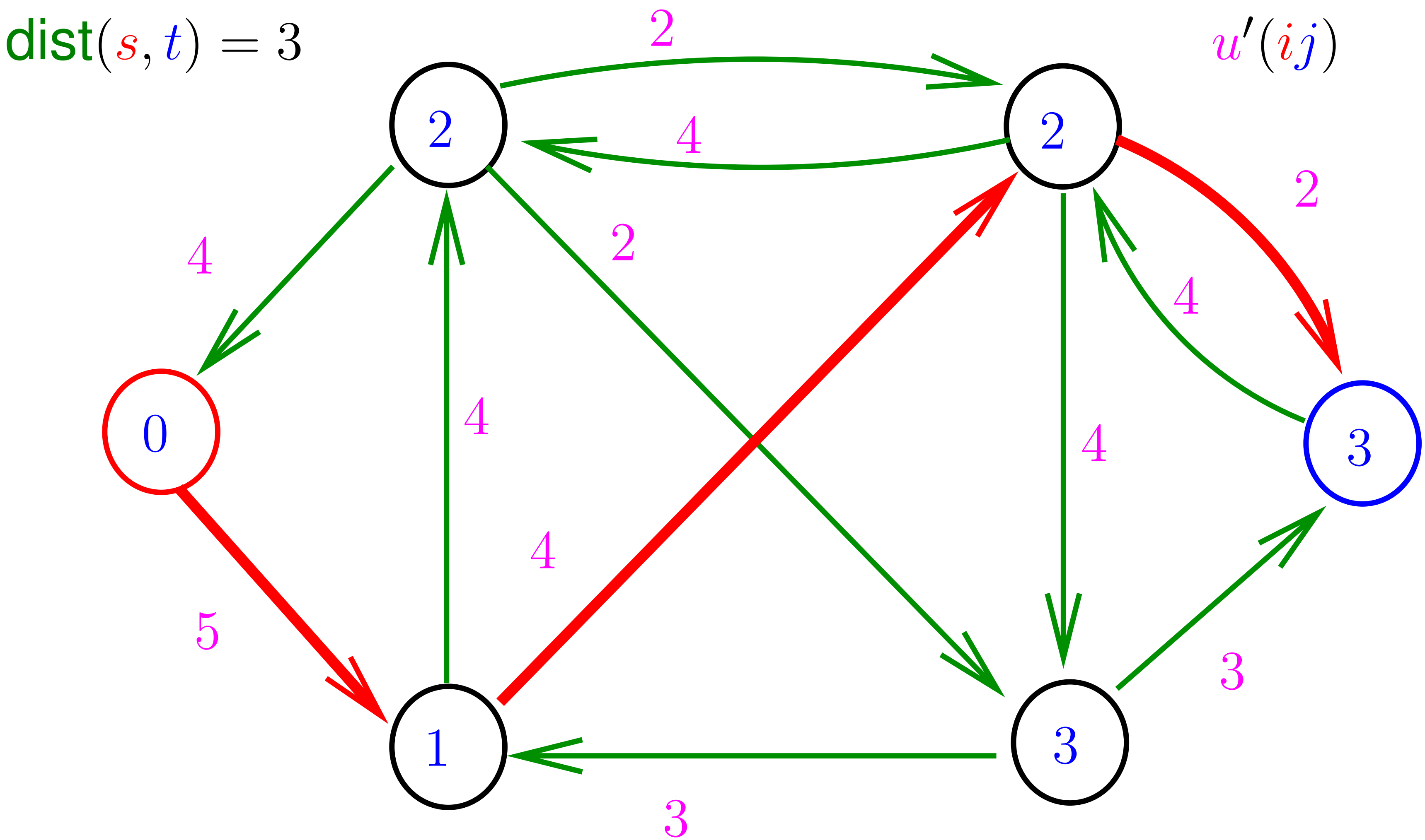
Arcos ausentes têm capacidade nula.

Rede residual 2

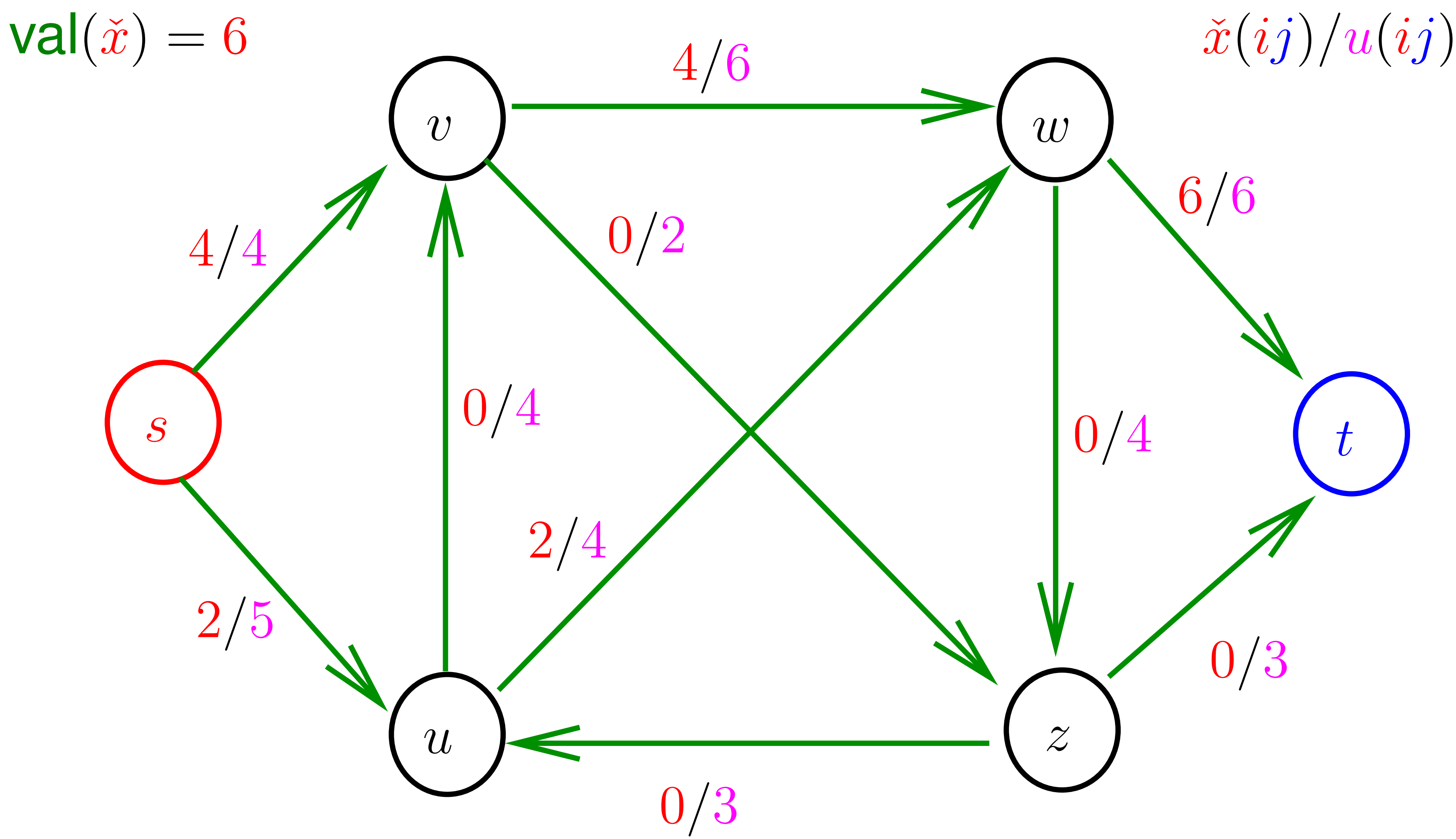


Arcos ausentes têm capacidade nula.

Caminho de incremento 2

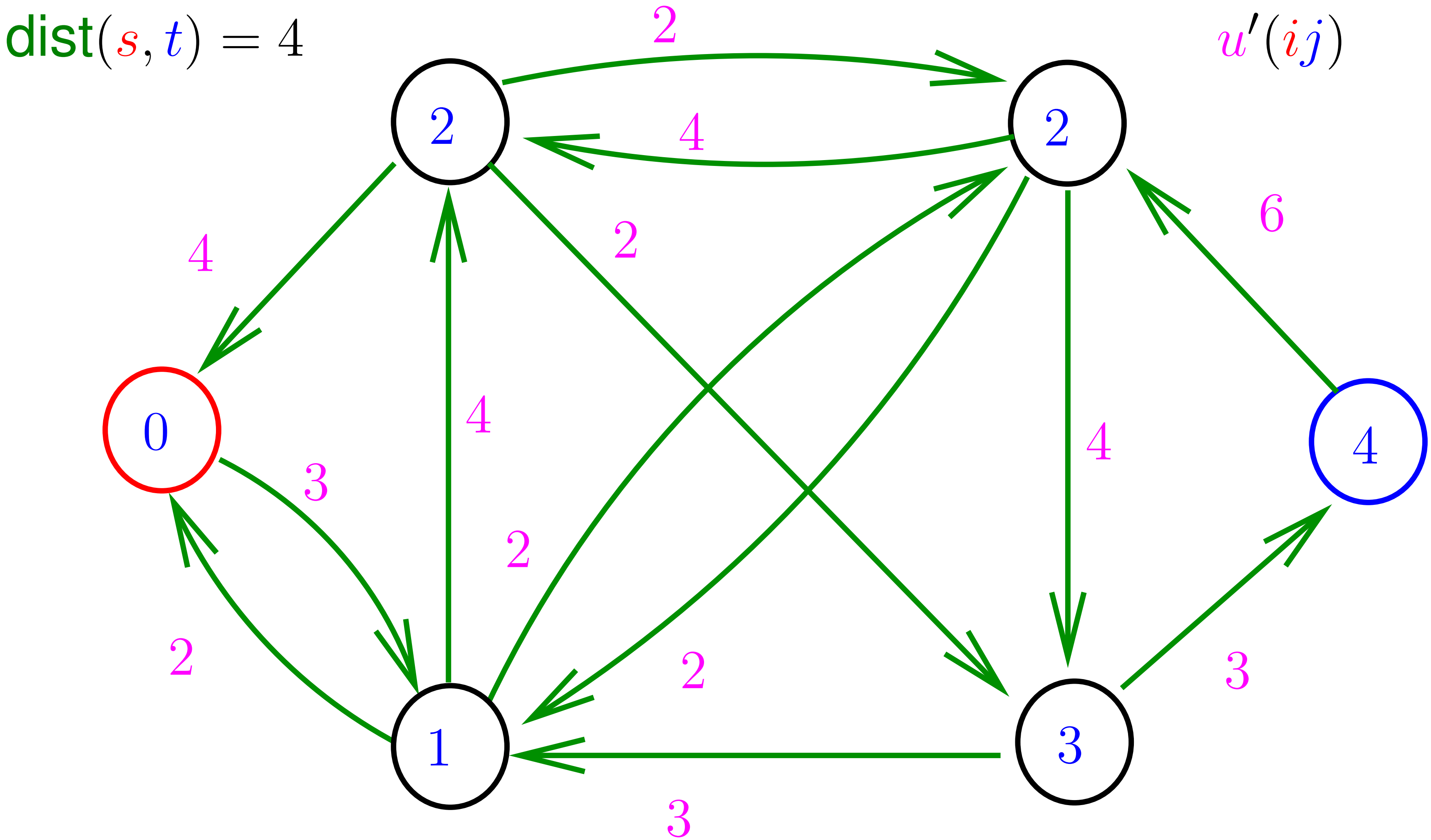


Rede 3



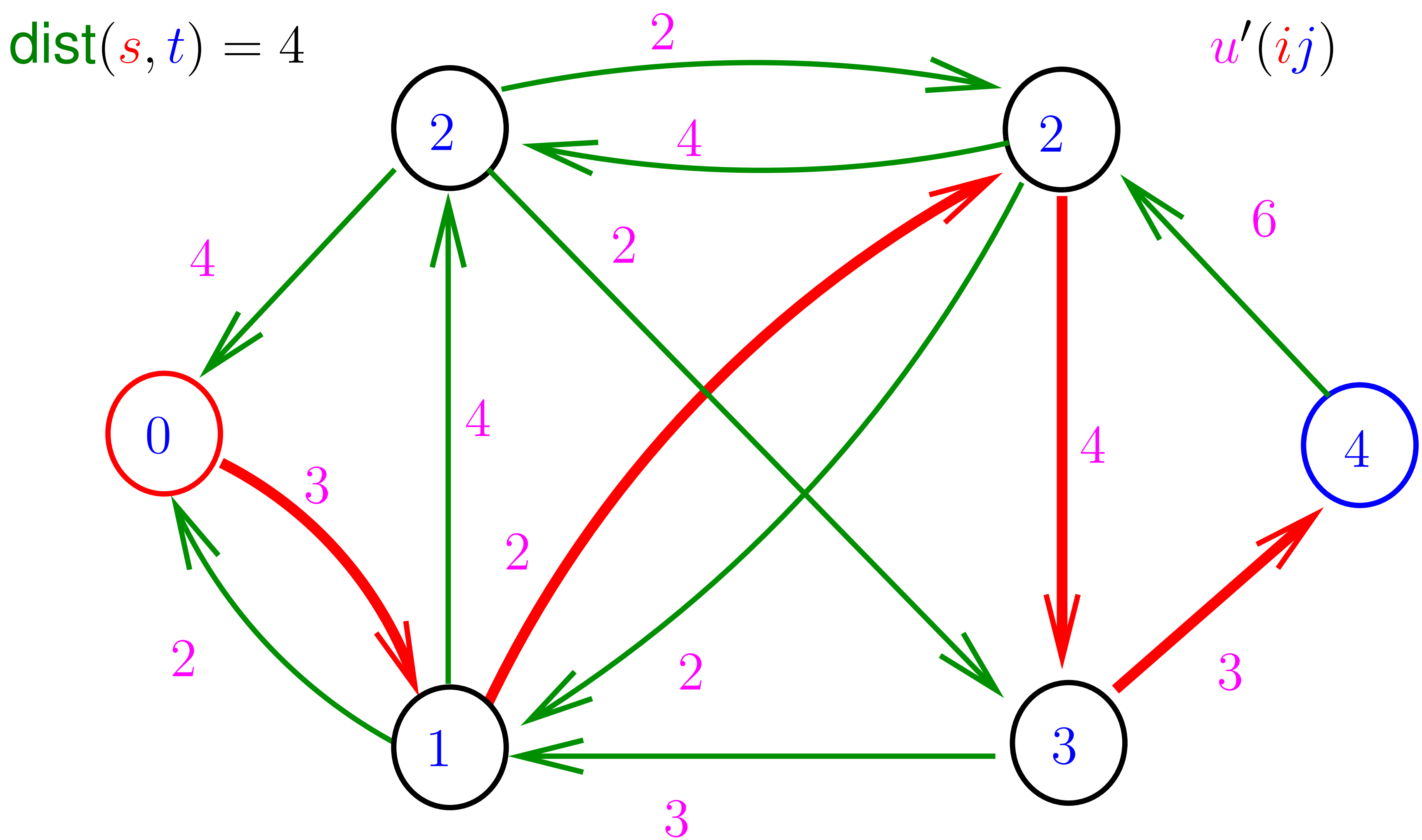
Arcos ausentes têm capacidade nula.

Rede residual 3



Arcos ausentes têm capacidade nula.

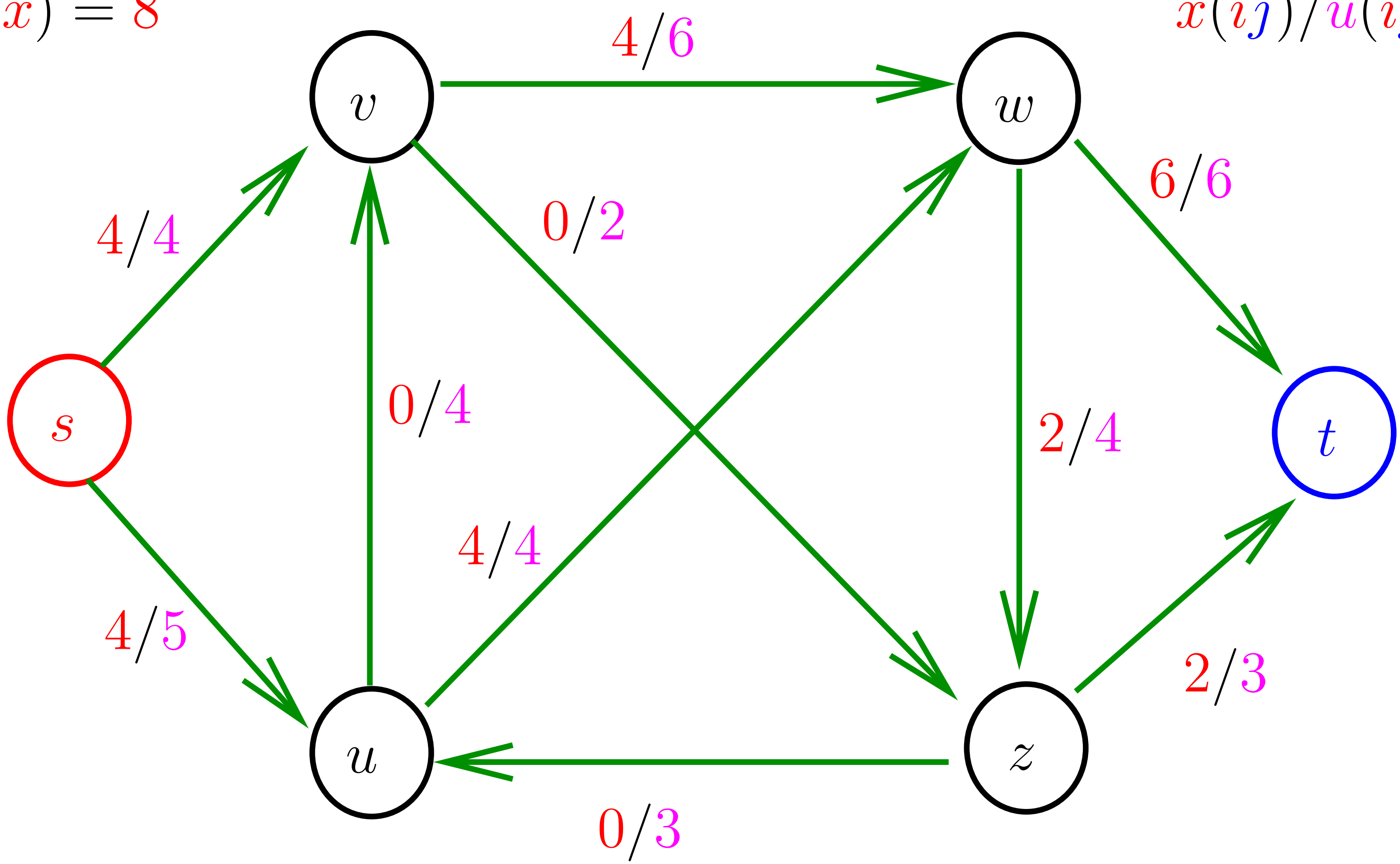
Caminho de incremento 3



Rede 4

$\text{val}(\check{x}) = 8$

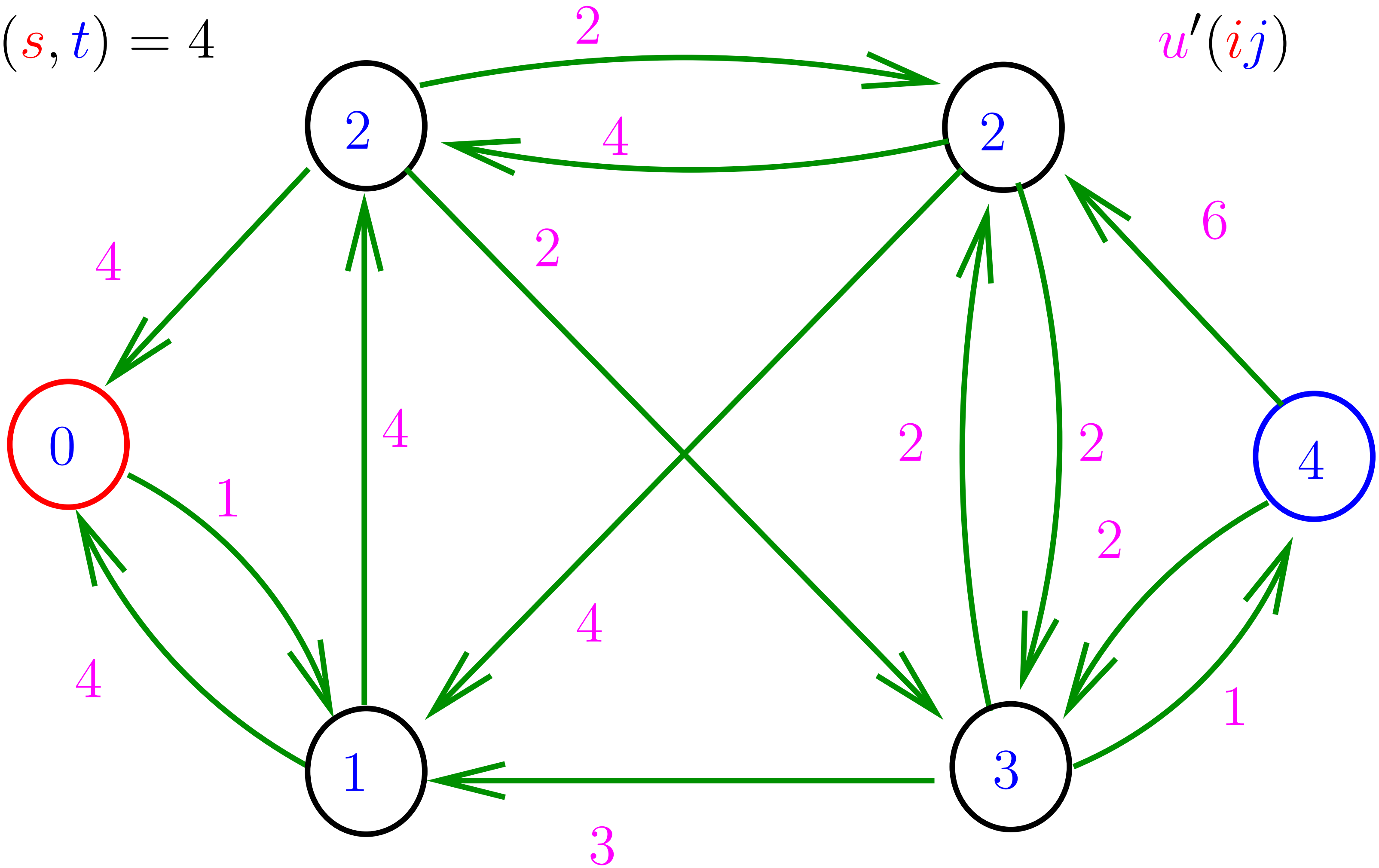
$\check{x}(ij)/u(ij)$



Arcos ausentes têm capacidade nula.

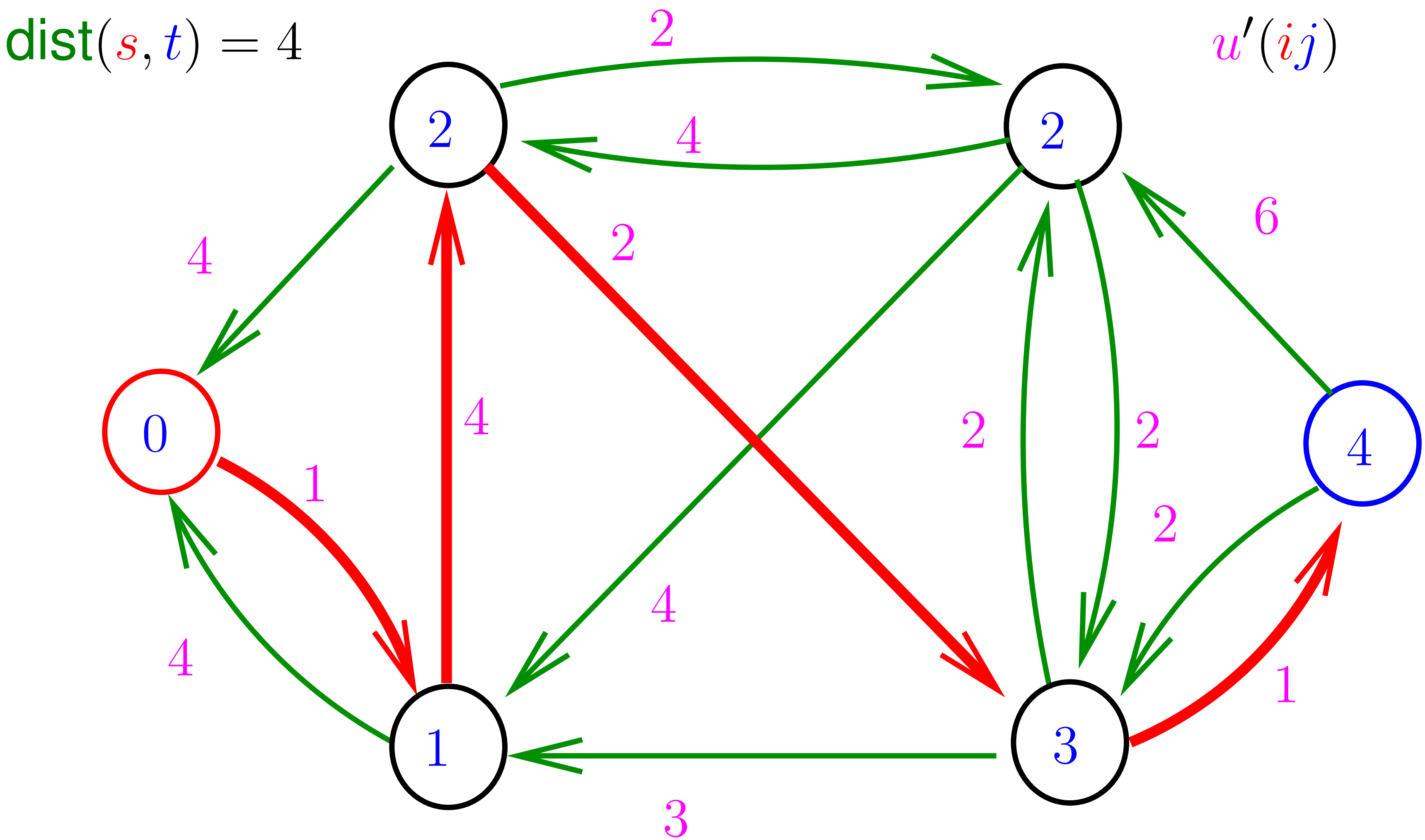
Rede residual 4

$\text{dist}(s,t) = 4$

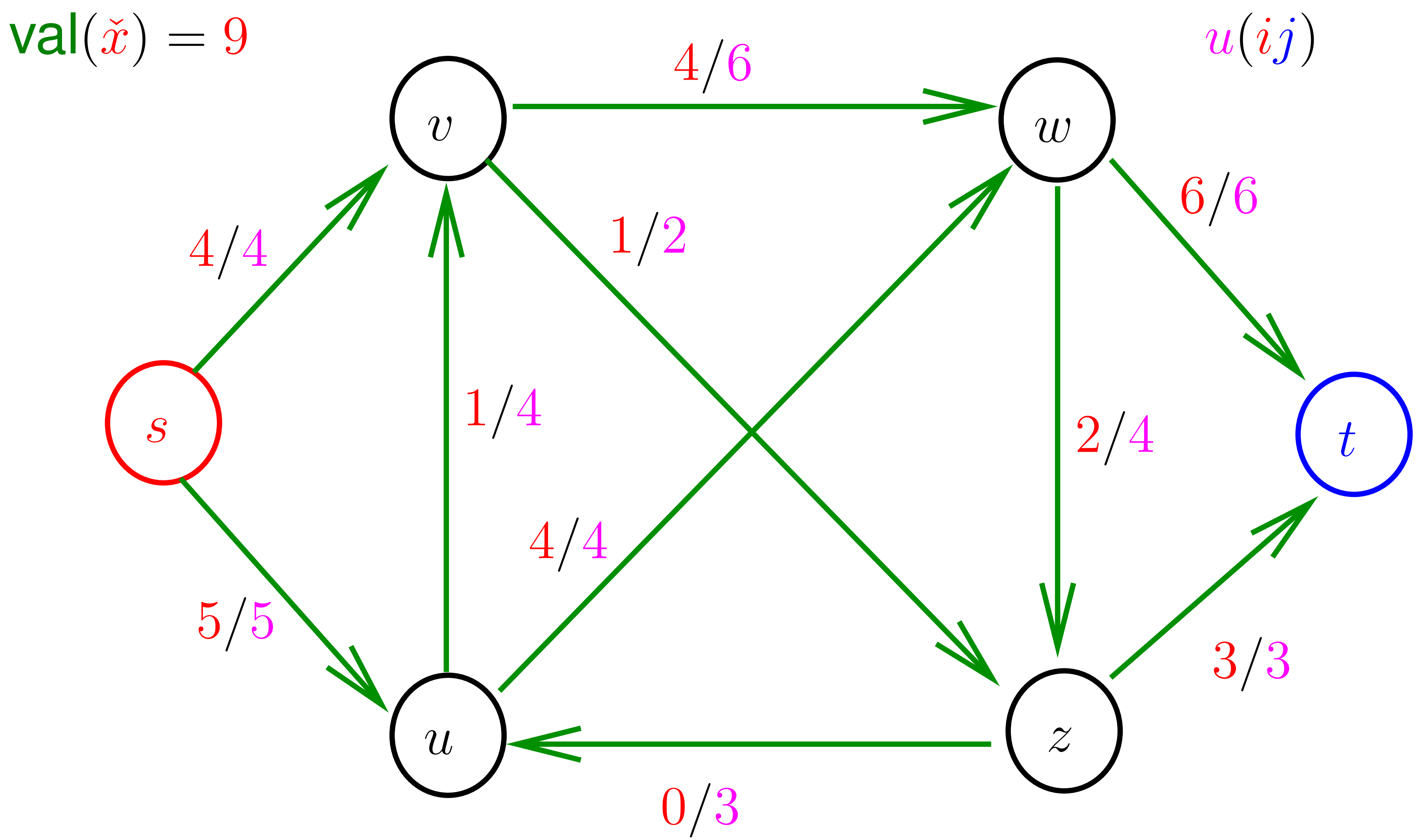


Arcos ausentes têm capacidade nula.

Caminho de incremento 4

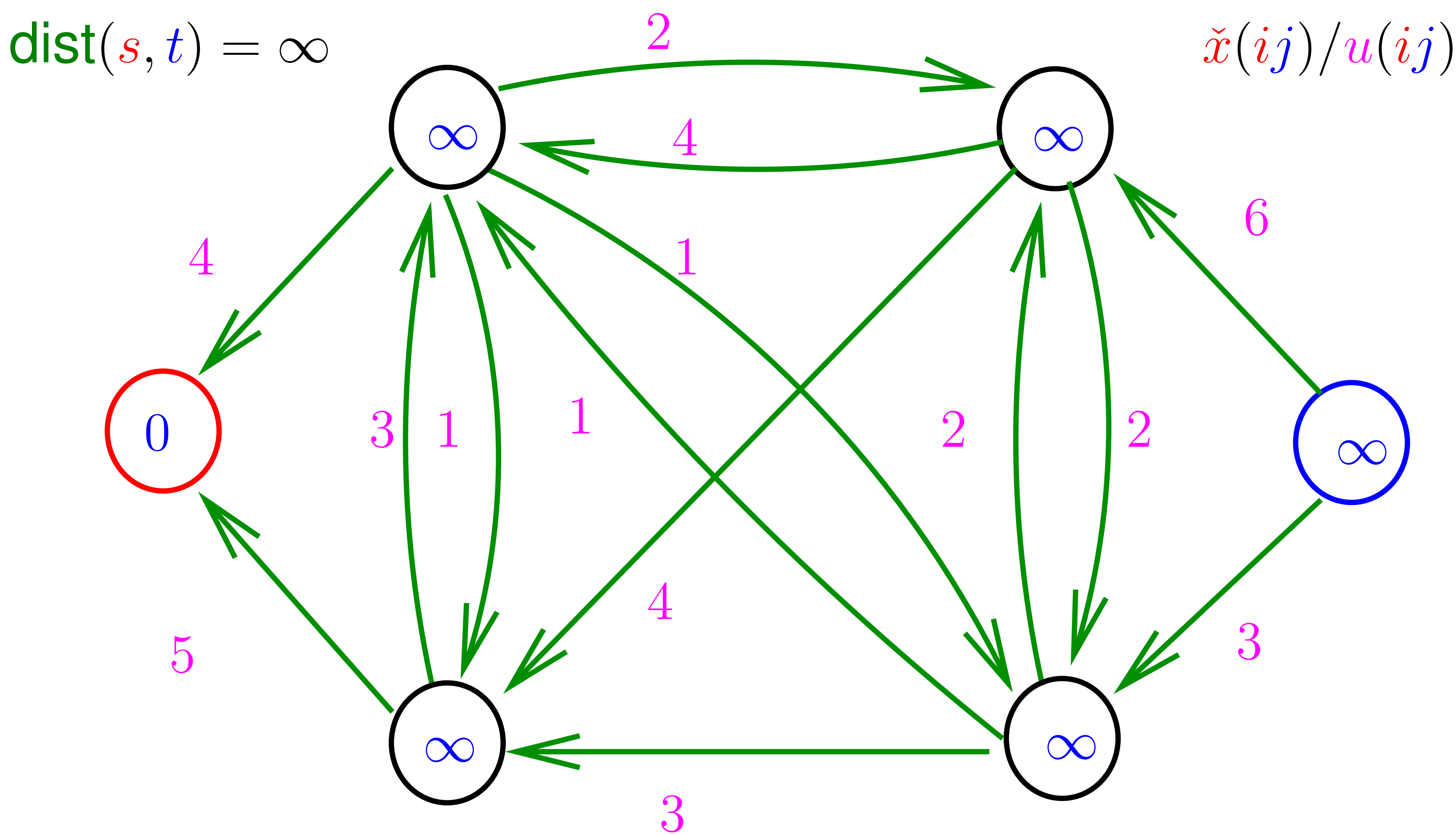


Rede 5



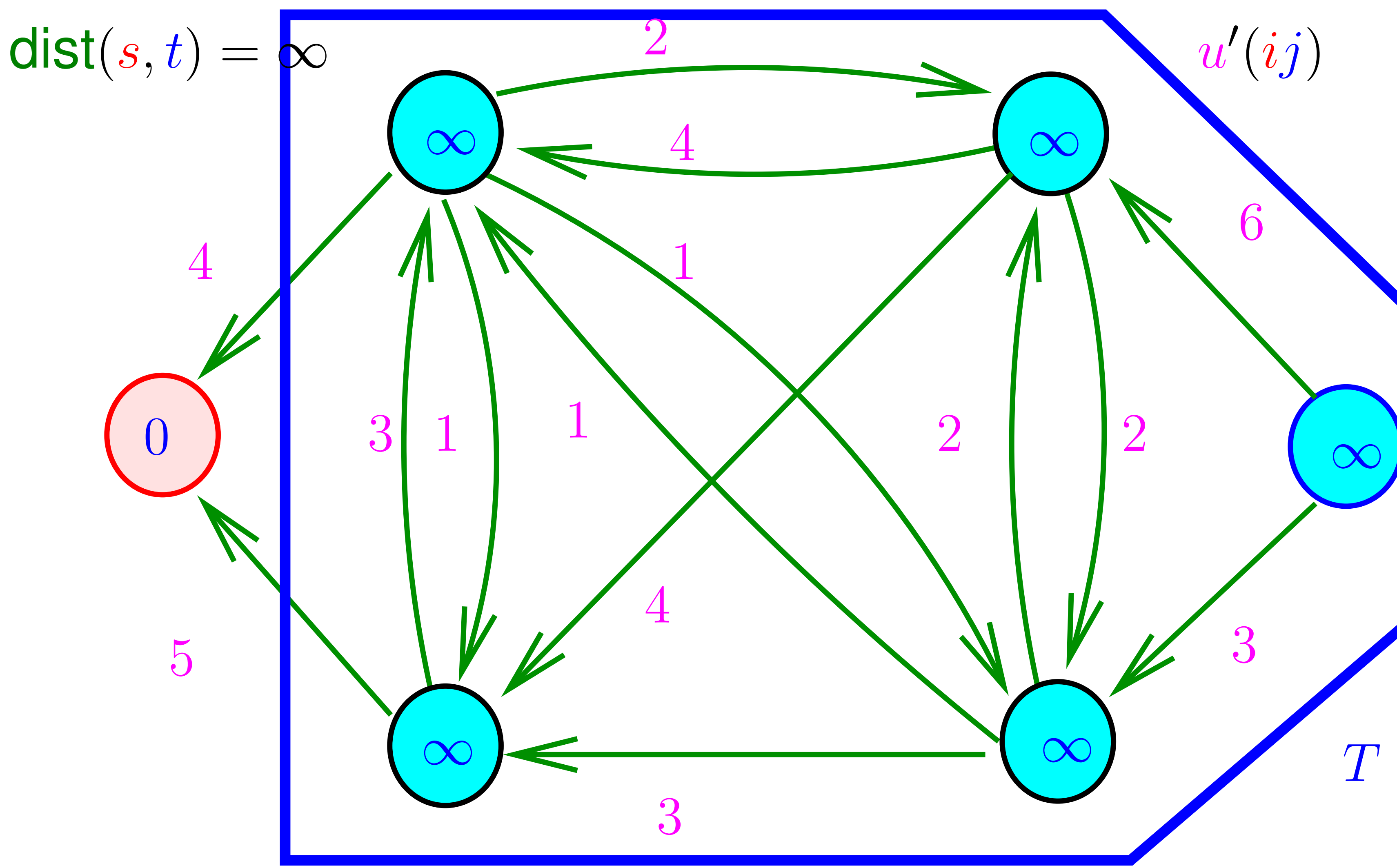
Arcos ausentes têm capacidade nula.

Rede residual 5



Arcos ausentes têm capacidade nula.

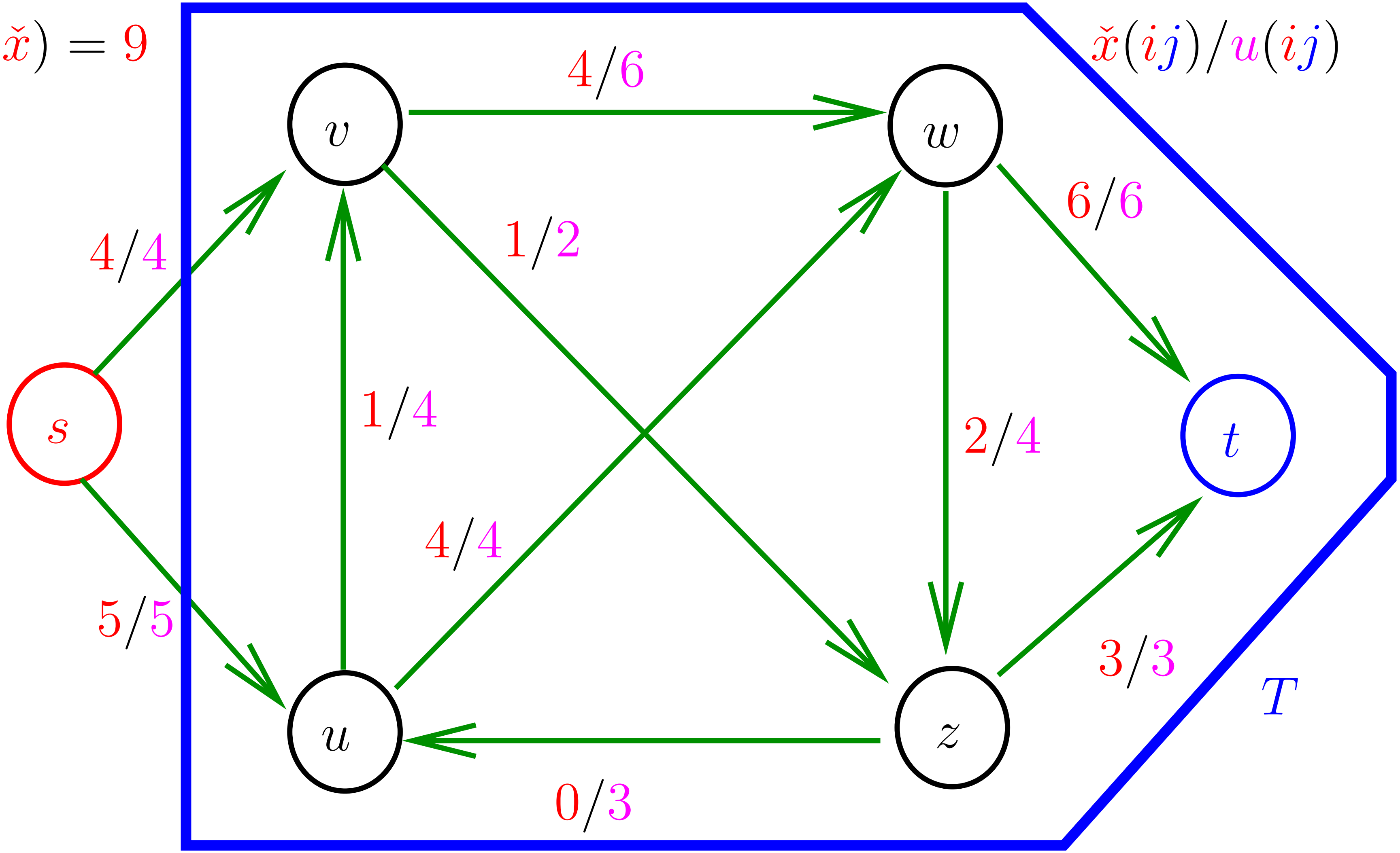
Caminho de incremento 5



Arcos ausentes têm capacidade nula.

Fluxo máximo e corte mínimo

$\text{val}(\check{x}) = 9$



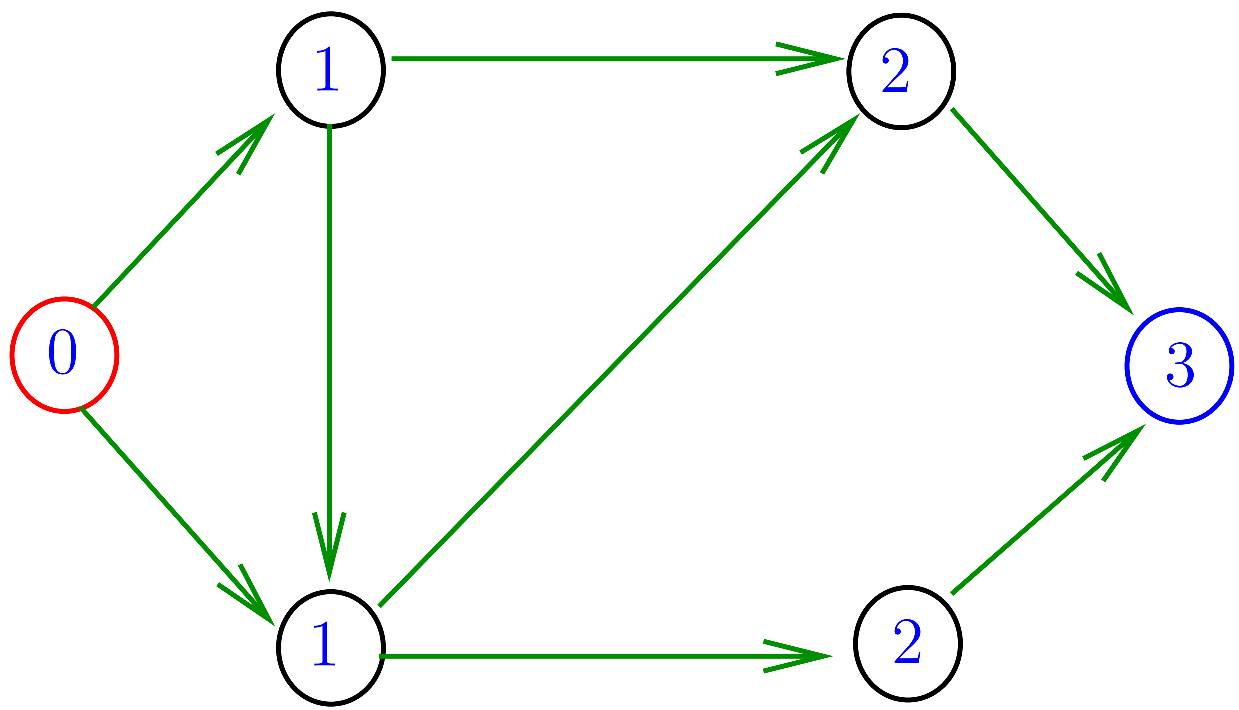
Arcos ausentes têm capacidade nula.

Potencial ótimo

Dado um nó s, um 1-potencial y é $(s,*)$-ótimo se:

- existe um caminho P de s a i tal que $|P| = y(i) - y(s)$ ou
- não existe um caminho de s a i e $y(i) - y(s) \geq n$.

Um arco ij é justo se $y(j) - y(i) = 1$.



Algoritmo de Edmonds e Karp

Recebe uma rede simétrica (N, A, u) com função-capacidade u e nós s e t e devolve um st-fluxo máximo x e um st-corte mínimo $\nabla(\overline{T}, T)$.

EDMONDS-KARP (N, A, u, s, t)

$\check{x} \leftarrow 0$

repita

  $A_{\check{x}} \leftarrow \{ij \in A : \check{x}(ij) < u(ij)\}$

  $\langle y, P\rangle \leftarrow$ CAMINHO-MÍNIMO$(N, A_{\check{x}}, s, t)$

  se $y(t) - y(s) < n$

    então $\check{x} \leftarrow$ INCREMENTE-FLUXO$(\check{x}, P)$

até que $y(t) - y(s) \geq n$

$x \leftarrow$ FLUXO$(\check{x})$

$T \leftarrow \{j \in N : y(j) - y(s) \geq n\}$

devolva x e T

Fases

As iterações do algoritmo (execuções da linhas 3–6) podem ser agrupadas em fases.

Duas iterações estão na mesma fase se os correspondentes caminhos de incremento tem o mesmo número de arcos.

É evidente que o número de fases é $\leq n - 1$.

Número de iterações

- (N, A_k) := rede residual construída da iteração k (linha 3);
- y_k := 1-potencial $(s, *)$-ótimo obtido na iteração k (linha 4). Podemos supor que $y_k(s) = 0$.
- α_k := número de arcos que pertencem a algum st-caminho mínimo em (N, A_k)

Para $k = 1, \ldots, r-1$ (r = número de iterações) vale que

(1A) $y_{k+1}(t) \geq y_k(t)$;
(1B) $y_{k+1}(t) = y_k(t) \Rightarrow \alpha_{k+1} < \alpha_{k+1}$.

(1A)+(1B) $\Rightarrow$ número execuções das linhas 3–6 $\leq nm$

Antes ... um observação que ajuda

Para cada arco ij em A_{k+1} vale que

(2A) $y_k(j) - y_k(i) \leq \ \ 1$ se $ij \in A_k$;
(2B) $y_k(j) - y_k(i) = -1$ se $ij \notin A_k$;

Demonstração de (2A) (rascunho).

Segue do fato de y_k ser um 1-potencial.

Demonstração de (2B) (rascunho).

Se $ij \notin A_k$, então $ji \in A_k$ e ji pertence ao caminho de incremento P devolvido na linha 4 durante a iteração k.

Logo, $y_k(i) - y_k(j) = 1 \Rightarrow y_k(j) - y_k(i) = -1$.

Rascunho da demonstração de (1A)

Seja $P = \langle s = i_0, i_1, i_2, \ldots, i_{q-1}, i_q = t \rangle$ um st-caminho mínimo em (N, A_{k+1}).

Temos que

$$\begin{aligned} y_{k+1}(t) &= |P| \\ &\geq (y_k(t) - y_k(i_{q-1})) + \cdots + (y_k(i_1) - y_k(s)) \\ &= y_k(t) - y_k(s) \\ &= y_k(t), \end{aligned}$$

onde a desigualdade é devida a (2A) e (2B).

Rascunho da demonstração de (1B)

Por (2B) temos que se

$$y_{k+1}(t) = y_k(t)$$

então todo arco em um st-caminho mínimo em (N, A_{k+1}) é um arco em (N, A_k).

Deste fato segue que: $y_{k+1}(t) = y_k(t) \Rightarrow \alpha_{k+1} \leq \alpha_k$.

Na iteração k, pelo menos um arco do st-caminho de incremento em (N, A_k) foi saturado e não está em (N, A_{k+1}). Portanto, $\alpha_{k+1} < \alpha_k$.

Consumo de tempo

O número de iterações das linhas 3–7 é $\leq nm$.

linha	consumo de todas as execuções da linha
1	$\mathrm{O}(m)$
3	$nm\,\mathrm{O}(m) = \mathrm{O}(nm^2)$
4	$nm\,\mathrm{O}(n+m) = \mathrm{O}(n^2m + nm^2)$
5	$nm\,\mathrm{O}(1) = \mathrm{O}(nm)$
6	$nm\,\mathrm{O}(n) = \mathrm{O}(n^2m)$
7	$nm\,\mathrm{O}(1) = \mathrm{O}(nm)$
8–10	$\mathrm{O}(n+m)$

total = $\mathrm{O}(nm^2)$ (para redes conexas $n \leq m$)

Conclusão

O algoritmo EDMONDS-KARP faz não mais que nm passos de incremento.

O consumo de tempo do algoritmo EDMONDS-KARP é $O(nm^2)$.

Este consumo de tempo é fortemente polinomial.