Fluxo Máximo

Ricardo Dutra da Silva

Universidade Tecnológica Federal do Paraná

Fluxo Máximo

Entrada

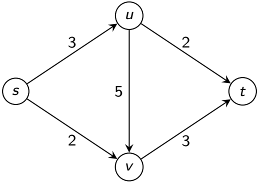
Uma grafo direcionado $G = (V, E)$, com custo/capacidade inteiro positivo c_e para cada aresta $e \in E$, e dois vértices distintos $s, t \in V$ tal que in-deg$(s) = 0$ e out-deg$(t) = 0$.

Saída

Fluxo $f(e) \geq 0$, para toda aresta $e \in E$, que maximiza o fluxo total $f = \sum_{e=(s,v) \in E} f(e)$, sujeito às restrições de:

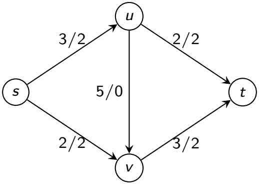
- capacidade: para toda aresta $e \in E$, $f(e) \leq c_e$;
- conservação: para todo vértice $v \in V \setminus \{s, t\}$, $\sum_{e=(u,v) \in E} f(e) = \sum_{e'=(v,w) \in E} f(e')$.

- Exemplo de rede de fluxo.

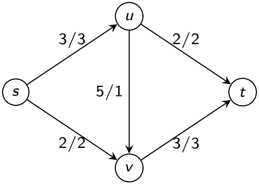


Fluxo Máximo

- Considere os rótulos $c_e/f(e)$.
- Fluxo total $f = \sum_{e=(s,v)\in E} c_e = 2 + 2 = 4$.
- Não é o fluxo máximo.



- Considere os rótulos $c_e/f(e)$.
- Fluxo total $f = \sum_{e=(s,v)\in E} c_e = 3 + 2 = 5$.
- Fluxo máximo.



- Tática:
  1. encontra um caminho de s a t ainda com capacidade residual;
  2. encontra o gargalo do caminho;
  3. incrementa o valor de toda as aresta no caminho com o valor do gargalo.

Fluxo Máximo

Algoritmo: $\textsc{Fluxo}(G = (V, E),\ s,\ t)$

`/* Retorna um caminho entre s e t com gargalo maior que 0. */`

$P \leftarrow \textsc{Busca}(G{,}s{,}t)$

enquanto $P \neq \emptyset$ **faça**

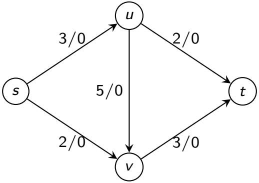
  $\Delta \leftarrow \min_{e \in P} \{c_e - f(e)\}$

  para $e \in P$ **faça**

    $f(e) = f(e) + \Delta$

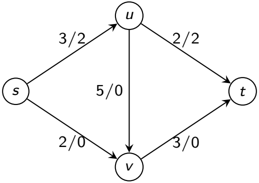
  $P \leftarrow \textsc{Busca}(G{,}s{,}t)$

- Exemplo.

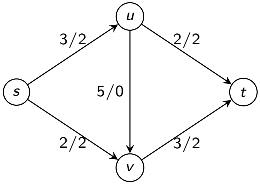


Fluxo Máximo

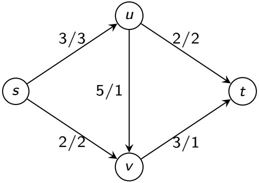
- Iteração 1: $P = s, u, t$; $\Delta = 2$.



- Iteração 2: $P = s, v, t$; $\Delta = 2$.

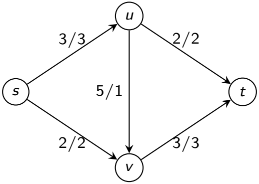


- Iteração 3: $P = s, u, v, t$; $\Delta = 1$.

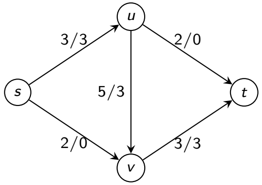


Fluxo Máximo

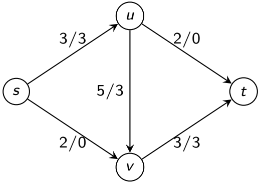
- Fluxo total $f = \sum_{e=(s,v)\in E} f(e) = 3 + 2 = 5$.
- Problema: outros caminhos podem levar a um fluxo não máximo.



- Iteração 1: $P = s, u, v, t$; $\Delta = 3$.



- Fluxo total $f = \sum_{e=(s,v)\in E} f(e) = 3 + 0 = 3$.



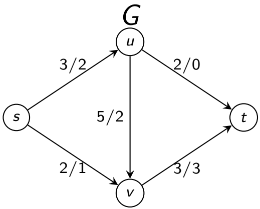
- Solução: fazer buscas em um grafo residual.

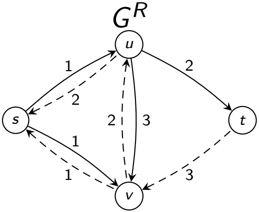
Definição

O grafo residual $G^R = (V', E')$ *para uma rede de fluxos* $G = (V, E)$ *é tal que* $V' = V$ *e para toda aresta* $(u, v) \in E$*:*

- *a aresta residual* $(u, v) \in E'$ *se seu peso, dado por* $c_{u,v} - f(u, v)$*, é maior que* 0*;*
- *a aresta de retorno* $(v, u) \in E'$ *se seu peso, dado por* $f(u, v)$*, é maior que* 0*.*

- Exemplo de grafo residual.





Fluxo Máximo

Algoritmo: FORDFULKERSON$(G = (V, E),\ s,\ t)$

$G^R \leftarrow$ GRAFORESIDUAL(G)

$P \leftarrow$ BUSCA(G^R, s, t)

enquanto $P \neq \emptyset$ **faça**

    $\Delta \leftarrow min_{e \in P} \{c_e\}$

    para $e \in P$ **faça**

        se TIPO$(e) =$ ARESTARESIDUAL **então**

            $f(e) = f(e) + \Delta$

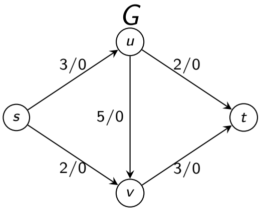
        senão

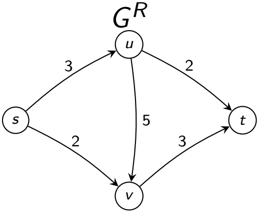
            $f(e) = f(e) - \Delta$

    $G^R \leftarrow$ GRAFORESIDUAL(G)

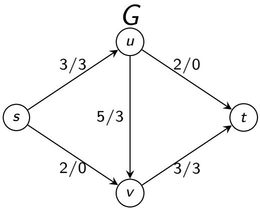
    $P \leftarrow$ BUSCA(G^R, s, t)

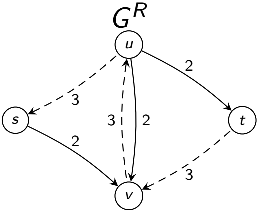
- Exemplo.



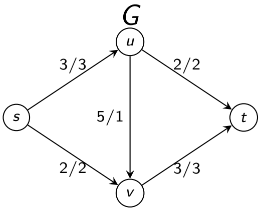


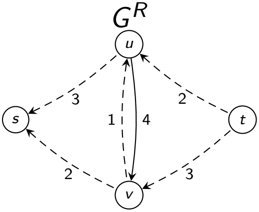
- Iteração 1. $P = s, u, v, t$; $\Delta = 3$.



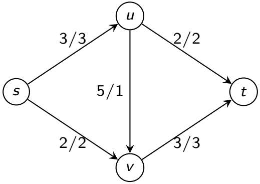


- Iteração 2. $P = s, v, u, t$; $\Delta = 2$.





- Fluxo total $f = \sum_{e=(s,v)\in E} f(e) = 3 + 2 = 5.$



Fluxo Máximo

Teorema

No algoritmo de FORDFULKERSON, *os fluxos* $f(e)$ *e as capacidades residuais são inteiras.*

Demonstração.

O invariante vale antes da primeira iteração pela natureza da entrada. Em uma iteração qualquer, os custos do grafo residual são inteiros, logo o gargalo é inteiro e os novos fluxos e capacidades residuais são inteiros. $\square$

Fluxo Máximo

Teorema

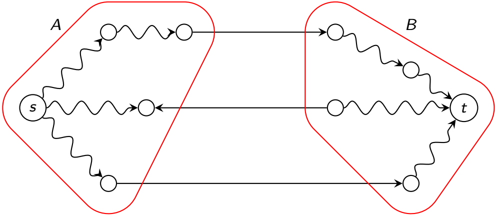
O fluxo $\sum_{e \in \delta^+(s)} f(e)$ *aumenta a cada iteração do algoritmo* FordFulkerson *com o caminho* P *encontrado.*

Demonstração.

A primeira aresta e, de P, é uma aresta saindo de s e não pode haver uma segunda aresta de retorno diminuindo o valor de s, dado o conceito de caminho. Portanto e é uma aresta residual que incrementa o valor do fluxo saindo de s. $\square$

Definição

Uma (s, t)-corte de um grafo $G = (V, E)$ é uma partição de V em conjuntos A e B com $s \in A$ e $t \in B$.



Definição

A capacidade de um (s, t)-corte é a dada pela soma dos custos das arestas saindo de A.

$$\sum_{e \in \delta^+(A)} c_e$$

Definição

Um (s, t)-corte mínimo é um (s, t)-corte com a menor capacidade dentre todos os (s, t)-cortes possíveis.

Teorema

Se f é um fluxo de um grafo G, então as seguintes afirmações são equivalentes:

1. *f é um fluxo máximo de G;*
2. *existe em G um (s, t)-corte com capacidade igual a f;*
3. *não há um caminho entre s-t no grafo residual G^R de G.*

Demonstração.

1 implica 3.
Suponha que existe um caminho s-t em G^R. Então conseguiríamos um fluxo maior que f. Contradição. □

Demonstração.

3 implica 2.

Começamos mostrando que para qualquer corte (A, B), o fluxo saindo de A é igual a f.

O fluxo de f é

$$f = \sum_{e \in \delta^+(s)} f(e) = \sum_{e \in \delta^+(s)} f(e) - \sum_{e \in \delta^-(s)} f(e).$$

Pela restrição de conservação, para todo $v \in V \setminus \{s, t\}$,

$$\sum_{e \in \delta^+(v)} f(e) - \sum_{e \in \delta^-(v)} f(e) = 0.$$

$\square$

Demonstração.

Podemos somar para todo $v \in A$

$$f = \sum_{v \in A} \left(\sum_{e \in \delta^+(v)} f(e) - \sum_{e \in \delta^-(v)} f(e) \right)$$

e a única soma diferente de zero é aquela quando $v = s$. $\square$

Demonstração.

Olhando para a soma de forma alternativa, dada uma aresta e qualquer de G, temos:

- se e está contida em B ela não contribui com a soma acima;
- se e está contida em A, então $f(e)$ aparece uma vez e $-f(e)$ aparece uma vez;
- se e sai de A para B ela contribui com $f(e)$;
- se e sai de B para A ela contribui com -$f(e)$.

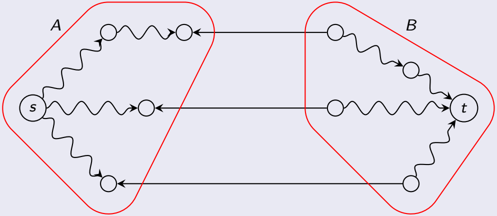
Portanto, temos que f é igual ao fluxo em (A, B).

$$f = \sum_{e \in \delta^+(A)} f(e) - \sum_{e \in \delta^-(A)} f(e).$$

$\square$

Demonstração.

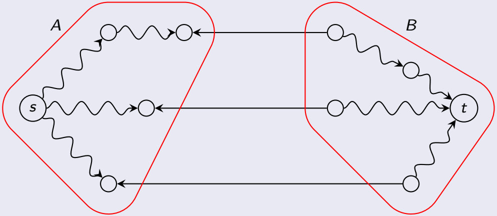
Vamos definir A como
$A = \left\{ v \in V : \text{existe um caminho } s \rightsquigarrow v \text{ em } G^R \right\}$. Como não há caminho s-t, teríamos um (s,t)-corte apenas com arestas chegando em A.



□

Demonstração.

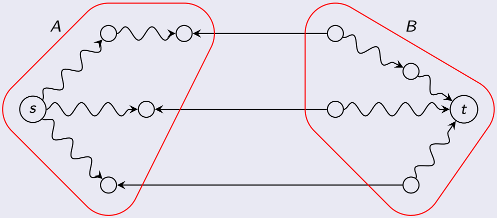
Toda aresta e saindo de A para B em G é tal que $f(e) = c_e$, senão haveria uma aresta residual de A para B em G^R.



□

Demonstração.

Toda aresta e saindo de B para A em G é tal que $f(e) = 0$, senão haveria uma aresta de retorno indo de A para B.



Portanto, o fluxo saindo de (A, B) é

$$\sum_{e \in \delta^+(A)} f(e) - \sum_{e \in \delta^-(A)} f(e) = \sum_{e \in \delta^+(A)} c_e - \sum_{e \in \delta^-(A)} 0,$$

a capacidade de (A, B). □

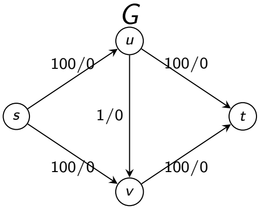
Demonstração.

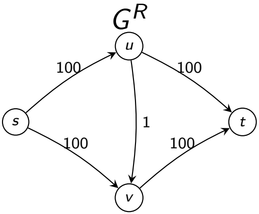
2 implica 1.
Para todo corte (A, B), $f \leq$ capacidade de (A, B), pois, pela restrição de capacidade:

$$\begin{aligned} f &= \sum_{e \in \delta^+(A)} \underbrace{f(e)}_{\leq c_e} - \sum_{e \in \delta^-(A)} \underbrace{f(e)}_{\geq 0} \\ &\leq \sum_{e \in \delta^+(A)} c_e \\ &= \text{capacidade de } (A, B). \end{aligned}$$

Logo, se para um corte (A, B), $f =$ capacidade de (A, B), então f é máximo. $\square$

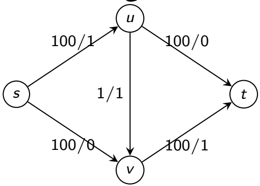
- Problema: dependendo dos caminhos retornados, podem ser realizadas f iterações do método.



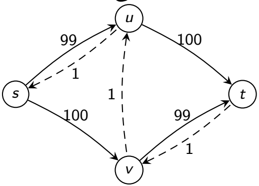


- Iteração 1. $P = s, u, v, t$; $\Delta = 1$.

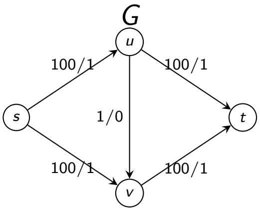
G

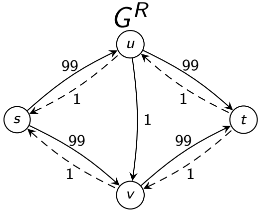


G^R

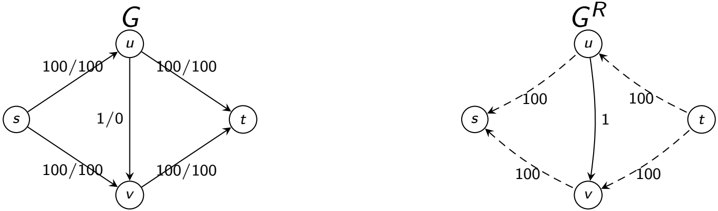


- Iteração 2. $P = s, v, u, t$; $\Delta = 1$.





- Se continuar o zigue-zague.
- Iteração 200. $P = s, v, u, t$; $\Delta = 1$.



Fluxo Máximo

Algoritmo: FordFulkerson($G = (V, E)$, s, t)

```
G^R ← GrafoResidual(G)
P ← Busca(G^R, s, t)
enquanto P ≠ ∅ faça
    Δ ← min_{e∈P} {c_e}
    para e ∈ P faça
        se Tipo(e) = ArestaResidual então
            f(e) = f(e) + Δ
        senão
            f(e) = f(e) − Δ
    G^R ← GrafoResidual(G)
    P ← Busca(G^R, s, t)
```

Teorema

O FORDFULKERSON *termina em no máximo* $C = \sum_{e\in\delta^+(s)} c_e$ *iterações.*

Demonstração.

Pelos resultados anteriores sabemos que o fluxo inicial 0 aumenta de pelos menos 1 a cada iteração, como o fluxo máximo f é menor ou igual a C, então em no máximo C iterações o algoritmo termina com um fluxo. $\square$

- Tempo, lembrando que em grafo conexo $m \in \Omega(n)$:
  - linhas 1 e 2: $\mathcal{O}(m)$
  - linha 3: $\mathcal{O}(C)$;
  - linha 4: $\mathcal{O}(Cn)$;
  - linhas 5 a 9: $\mathcal{O}(Cn)$;
  - linhas 10 e 11: $\mathcal{O}(Cm)$.
- Portanto, $\mathcal{O}(Cm)$.

Fluxo Máximo

Algoritmo: $\textsc{EdmondsKarp}(G = (V, E),\ s,\ t)$

$G^R \leftarrow \textsc{GrafoResidual}(G)$

$P \leftarrow \textsc{BuscaLargura}(G^R, s, t)$

enquanto $P \neq \emptyset$ **faça**

  $\Delta \leftarrow \min_{e \in P} \{c_e\}$

  para $e \in P$ **faça**

    se $\textsc{Tipo}(e) = \textsc{ArestaResidual}$ **então**

      $f(e) = f(e) + \Delta$

    senão

      $f(e) = f(e) - \Delta$

  $G^R \leftarrow \textsc{GrafoResidual}(G)$

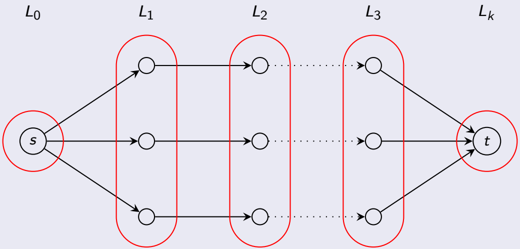
  $P \leftarrow \textsc{BuscaLargura}(G^R, s, t)$

Teorema

O algoritmo EdmondsKarp *possui tempo* $\mathcal{O}(nm^2)$.

Demonstração.

Um caminho na busca em largura conecta uma camada i com uma camada $i+1$. Vamos assumir um grafo em que os menores caminhos atravessam k camadas.



$\square$

Demonstração.

Após uma iteração do algoritmo EdmondsKarp, nenhum menor caminho é diminuído.
Basta perceber que uma iteração vai criar arestas de retorno e estas podem apenas criar caminhos maiores, pois vão ligar uma camada com a anterior. Aumentaria dois pulos.
Uma próxima iteração escolheria então um outro menor caminho daqueles atravessando k camadas. □

Demonstração.

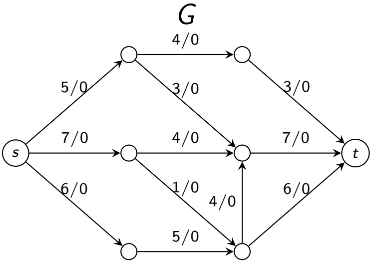
Em cada iteração, pelo menos uma das arestas atravessando as camadas é removida (a do gargalo).
Esta aresta não é incluída novamente. Para isto acontecer o caminho escolhido deveria passar por uma aresta de retorno e, como já argumentamos, seria uma caminho maior que k. □

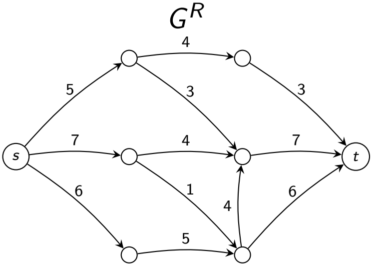
Demonstração.

Perceba que o número l de arestas atravessando entre camadas é tal que $l \leq m$. Como pelo menos uma aresta é removida a cada iteração, temos no máximo m iterações até todos os caminhos s-t se tornarem maiores que k.
Como todo caminho s-t no grafo tem no máximo $n-1$ arestas, k pode variar de 1 a $n-1$.
Teríamos então no máximo $\mathcal{O}(nm)$ iterações no algoritmo, o que leva a um tempo $\mathcal{O}(nm^2)$ para o algoritmo. □

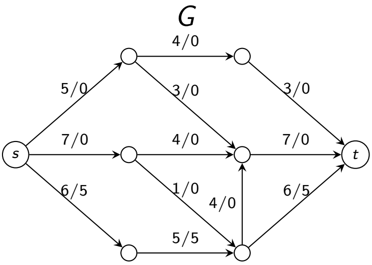
- Pelas provas acima vemos que o problema do fluxo máximo está relacionado com o problema de um corte mínimo (s, t).
- Podemos achar o corte mínimo usando o algoritmo de fluxo máximo.

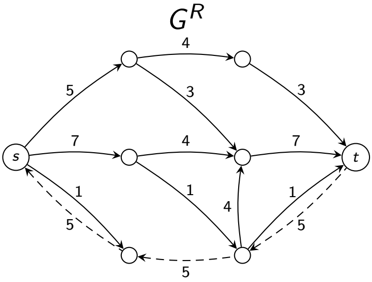
Fluxo Máximo

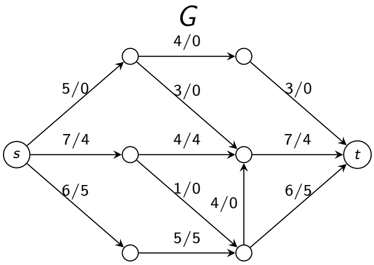




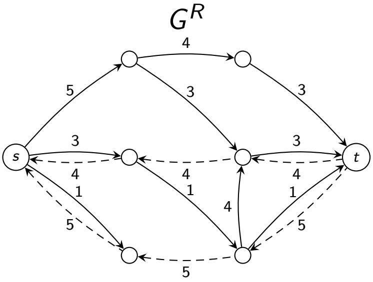
Fluxo Máximo





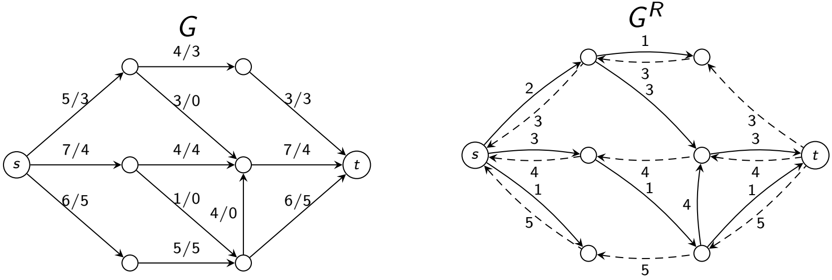


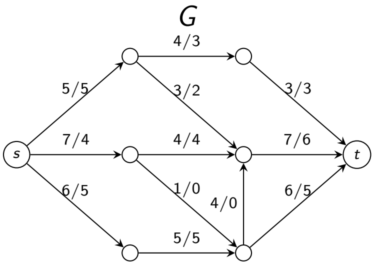
G
4/0
5/0
3/0
3/0
7/4
4/4
7/4
s
t
6/5
1/0
4/0
6/5
5/5

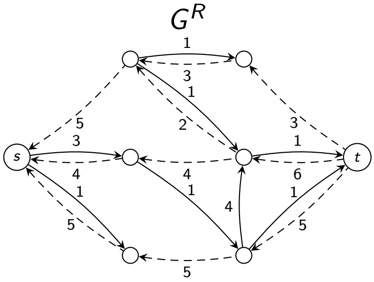


G^R
4
5
3
3
3
3
s
t
4
1
4
1
4
4
1
5
5
5

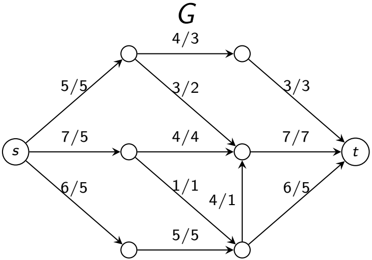
Fluxo Máximo

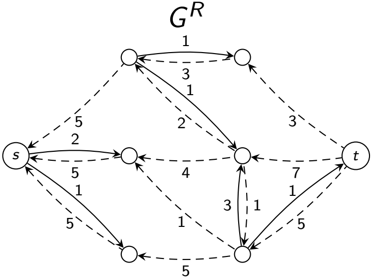



G
4/3
5/5
3/2
3/3
s
7/4
4/4
7/6
t
6/5
1/0
4/0
6/5
5/5


G^R
1
3
1
5
2
3
3
1
s
t
4
4
6
1
1
1
4
5
5
5

Fluxo Máximo

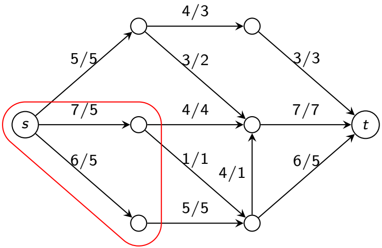




Fluxo Máximo

- Fluxo máximo $f = 15$ e corte mínimo tem capacidade 15.

G



G^R