Aula 16 - Fluxos em Redes

Luís Felipe

UFF

28 de Maio de 2026

Luís Felipe

28/05/26

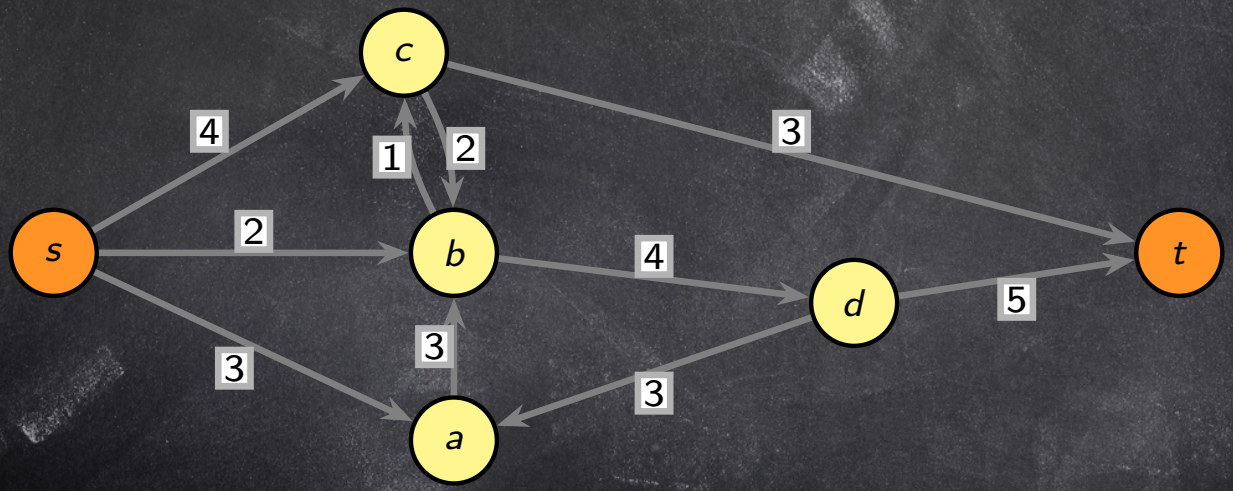
Fluxos em redes

Uma rede é um multidigrafo $D = (V, E)$ no qual cada aresta $e \in E$ possui uma capacidade real positiva: $c(e) > 0$.

Em geral, a rede possui dois vértices especiais:

- s: origem ou fonte;
- t: destino ou sumidouro.

Assumimos que s alcança todos os demais vértices, e que t é alcançado por todos os demais.



Cada número indica a capacidade da respectiva aresta.

Luís Felipe

28/05/26

Fluxo

Seja $D = (V, E)$ uma rede com capacidades $c : E \to \mathbb{R}_{>0}$.

Um fluxo é uma função

$$f : E \to \mathbb{R}_{\geq 0}$$

que satisfaz:

- Viabilidade:

$$0 \leq f(e) \leq c(e), \quad \forall e \in E.$$

- Conservação:

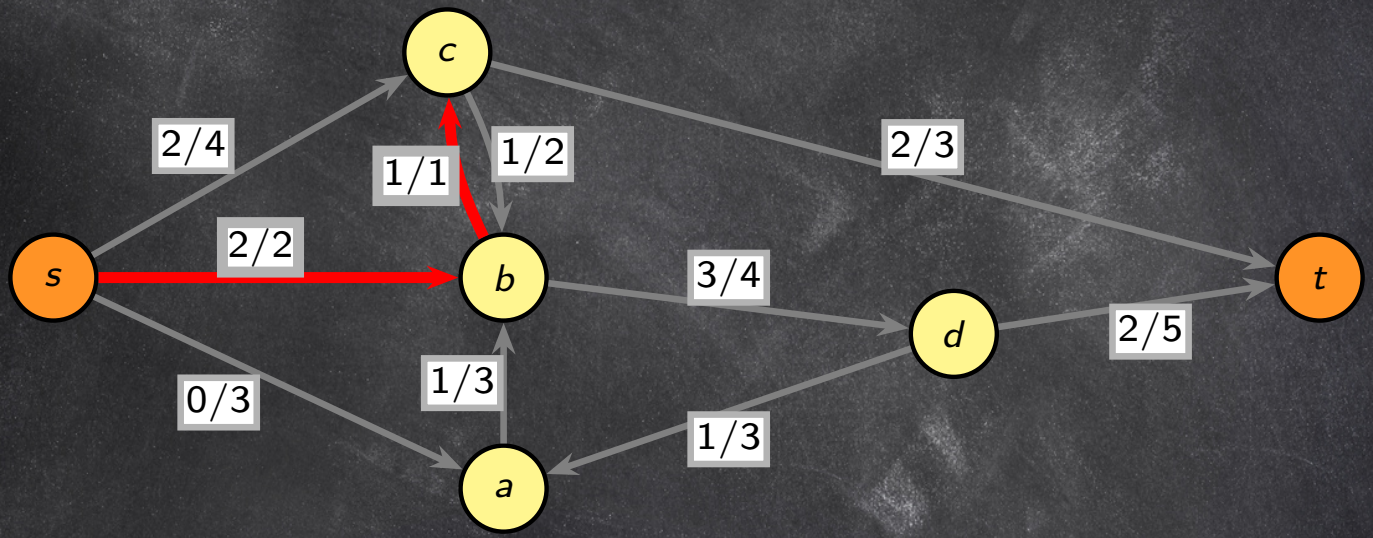
$$\sum_{x:(x,v)\in E} f(x, v) = \sum_{z:(v,z)\in E} f(v, z), \quad \forall v \in V \setminus \{s, t\}.$$

Ou seja, nenhum vértice intermediário cria nem destrói fluxo.

Luís Felipe

28/05/26

Fluxo em uma rede



O rótulo x/y em uma aresta representa: fluxo / capacidade.

Por exemplo, 2/4 significa que a aresta tem capacidade 4 e transporta fluxo 2.

Luís Felipe

28/05/26

Valor de um fluxo

O valor do fluxo na rede é denotado por $f(D)$ e definido por:

$$f(D) = \sum_{z:(s,z)\in E} f(s,z).$$

Isto é, o valor do fluxo é a quantidade total que sai da fonte s.

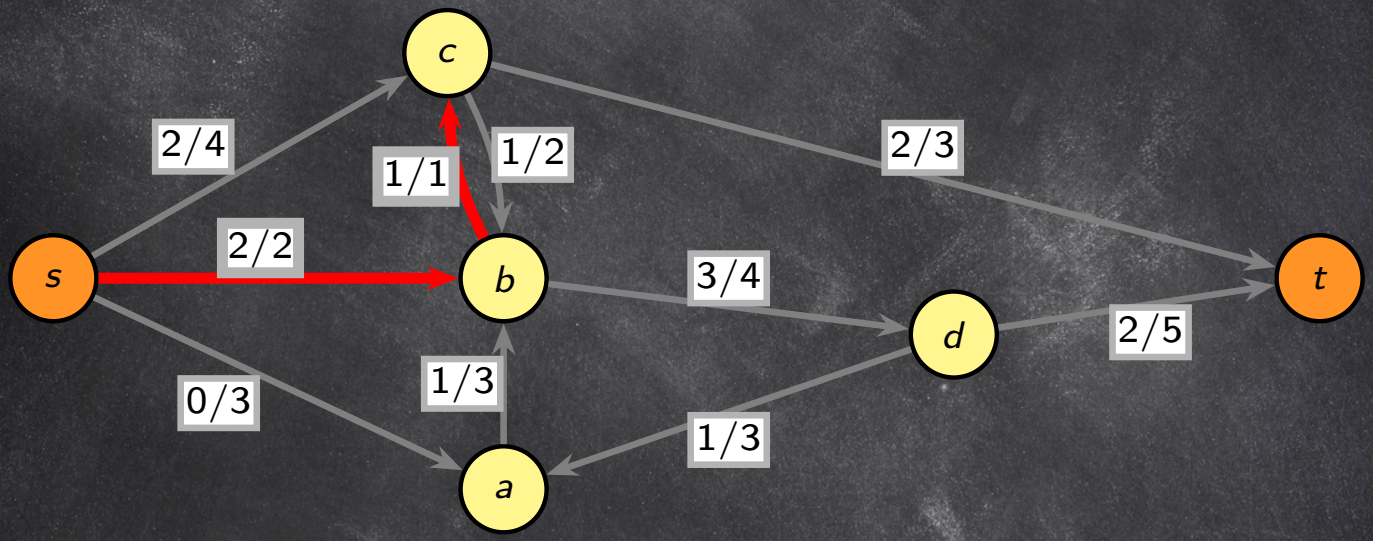
Pela conservação de fluxo, esse valor é igual à quantidade total que chega no destino t:

$$f(D) = \sum_{x:(x,t)\in E} f(x,t).$$

Luís Felipe

28/05/26

Exemplo



Neste exemplo:

$$f(D) = f(s,a) + f(s,b) + f(s,c) = 0 + 2 + 2 = 4.$$

Também:

$$f(D) = f(c,t) + f(d,t) = 2 + 2 = 4.$$

Luís Felipe

28/05/26

Problema do Fluxo Máximo

Problema do Fluxo Máximo

Entrada: uma rede $D = (V, E)$ com origem s, destino t e capacidades $c(e)$.

Objetivo: encontrar um fluxo f de maior valor possível.

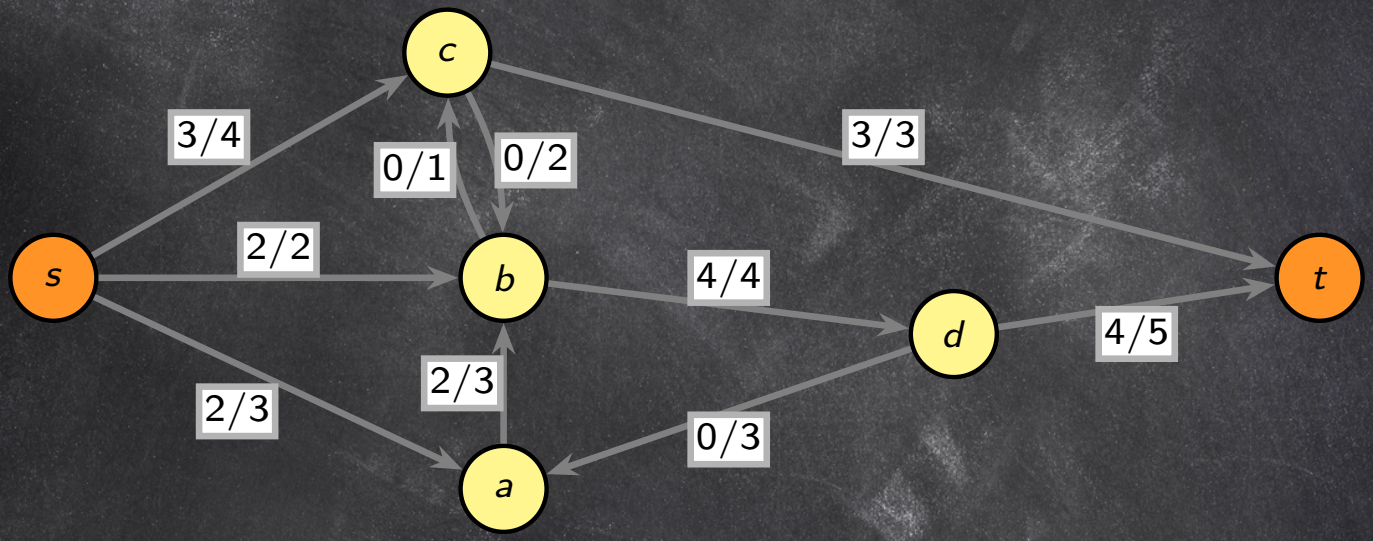
Ou seja, queremos maximizar:

$$f(D) = \sum_{z:(s,z)\in E} f(s,z).$$

Luís Felipe

28/05/26

Exemplo de fluxo máximo



Neste fluxo:

$$f(D) = 2 + 2 + 3 = 7.$$

Veremos depois que este fluxo é máximo.

Luís Felipe

28/05/26

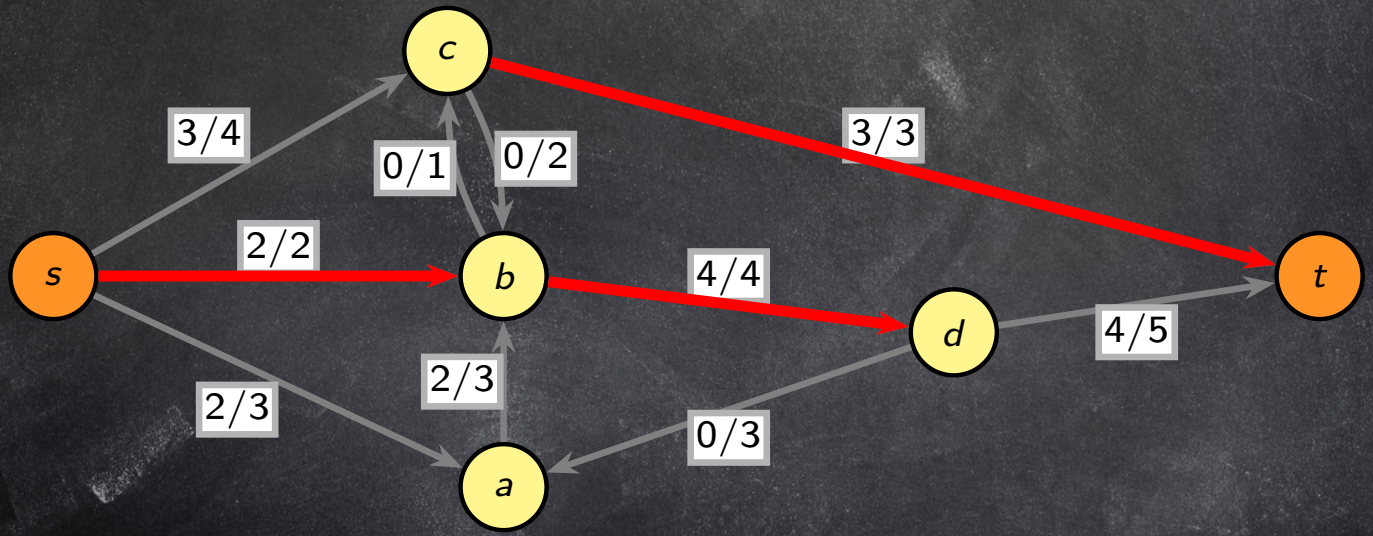
Aresta saturada

Uma aresta e é saturada quando:

$$f(e) = c(e).$$

Isto é, a aresta está usando toda a sua capacidade.

No exemplo abaixo, as arestas sb, bd e ct estão saturadas.



Luís Felipe

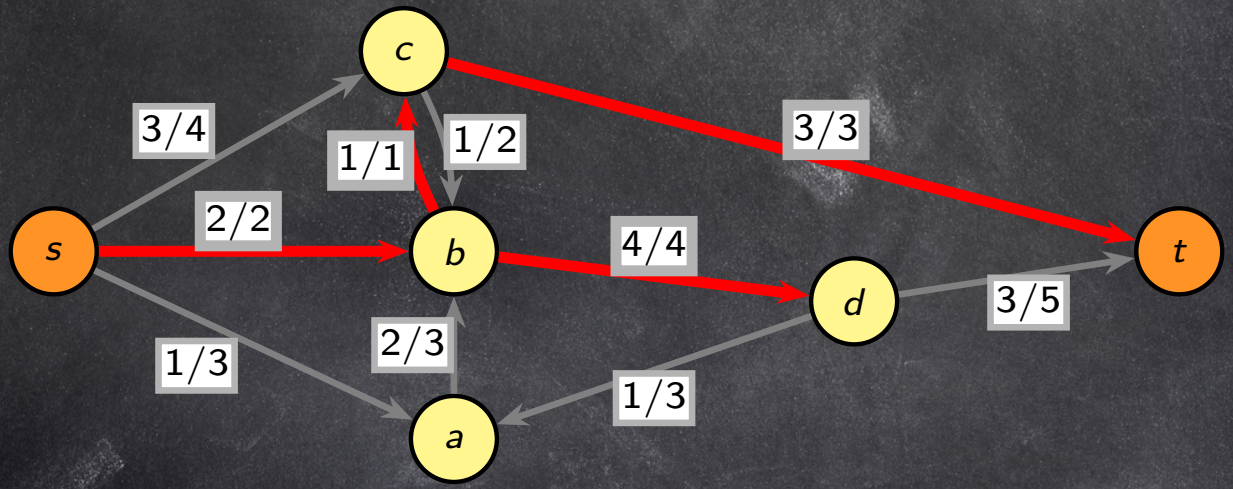
28/05/26

Fluxo maximal versus fluxo máximo

Um fluxo é maximal quando todo caminho direcionado de s a t contém alguma aresta saturada.

Intuitivamente: não conseguimos aumentar o valor do fluxo apenas colocando mais fluxo em algum caminho dirigido de s a t.

- Todo fluxo máximo é maximal.
- Nem todo fluxo maximal é máximo. Exemplo abaixo:



A razão é que às vezes precisamos diminuir fluxo em algumas arestas para conseguir aumentar o fluxo total.

Luís Felipe

28/05/26

Observação importante: Fluxo estático

Um fluxo não deve ser interpretado necessariamente como uma sequência temporal.

Ele é uma atribuição simultânea de valores às arestas da rede.

O que exigimos é:

$$f(e) \leq c(e)$$

para toda aresta e, e

$$\sum_x f(x, v) = \sum_z f(v, z)$$

para todo vértice interno v.

Assim, podem aparecer circulações internas, isto é, fluxo passando em ciclos.

Essas circulações não alteram o valor do fluxo da rede, pois o valor é medido pelo total que sai de s ou chega em t.

Luís Felipe

28/05/26

Cortes

Seja $S \subseteq V$ tal que:

$$s \in S \quad \text{e} \quad t \notin S.$$

Definimos $S' = V \setminus S$.

Um corte $[S, S']$ é o conjunto de arestas com um extremo em S e o outro extremo em S'.

A ideia é que todo caminho direcionado de s até t precisa atravessar o corte em algum momento.

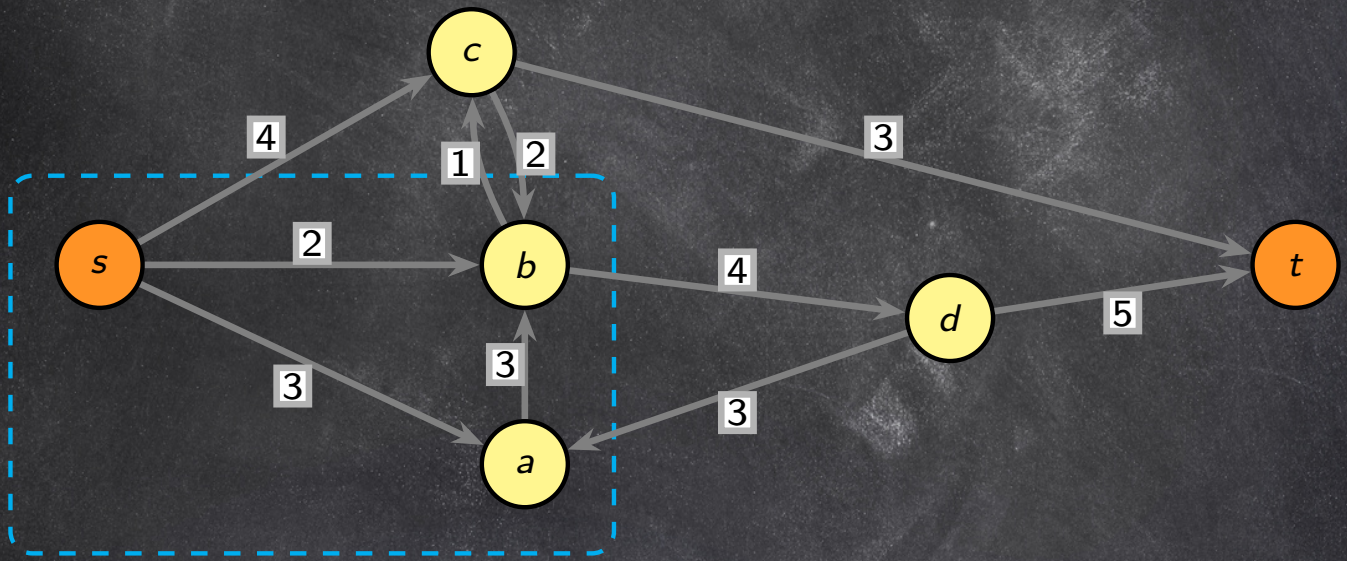
Luís Felipe

28/05/26

Exemplo de corte

Considere:

$$S = \{s, a, b\}, \quad S' = \{c, d, t\}.$$



As arestas do corte são as que cruzam de um lado para o outro.
Neste exemplo:

$$[S, S'] = \{sc, bc, cb, bd, da\}.$$

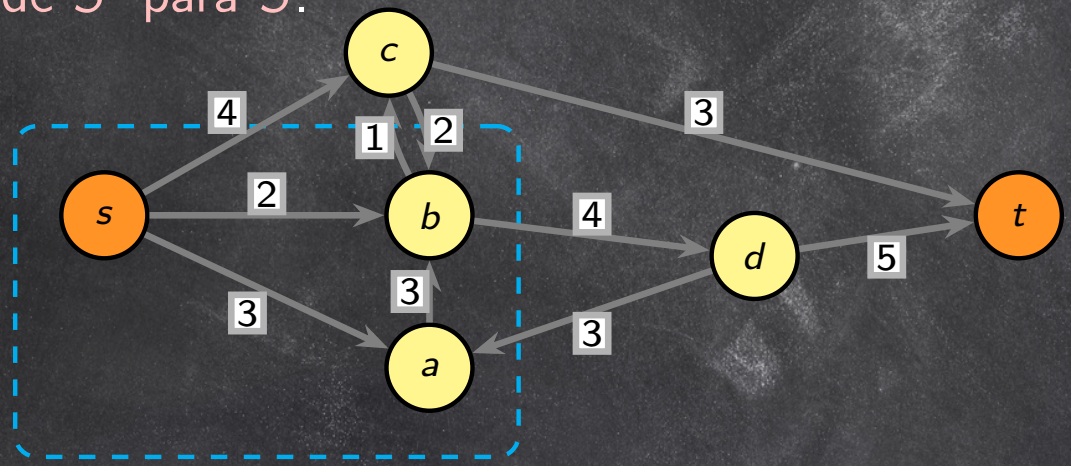
Luís Felipe

28/05/26

Arestas diretas e contrárias do corte

Para um corte $[S, S']$, definimos:

- $(S, S')^+$: conjunto das arestas diretas do corte, isto é, arestas que vão de S para S';
- $(S, S')^-$: conjunto das arestas contrárias do corte, isto é, arestas que vão de S' para S.



Para $S = \{s, a, b\}$ e $S' = V \setminus S = \{c, d, t\}$:

$$(S, S')^+ = \{sc, bc, bd\} \quad \text{arestas diretas do corte,}$$
$$(S, S')^- = \{cb, da\} \quad \text{arestas contrárias do corte.}$$

Atenção: aqui, "direta" e "contrária" são termos relativos ao corte, não à rede residual, que será visto daqui a pouco.

Luís Felipe

28/05/26

Capacidade de um corte

A capacidade de um corte $[S, S']$ é a soma das capacidades das arestas que vão de S para S':

$$c(S, S') = \sum_{e \in (S,S')^+} c(e).$$

Apenas as arestas de $(S, S')^+$ entram na soma.

As arestas de $(S, S')^-$ não contam para a capacidade do corte.

Luís Felipe

28/05/26

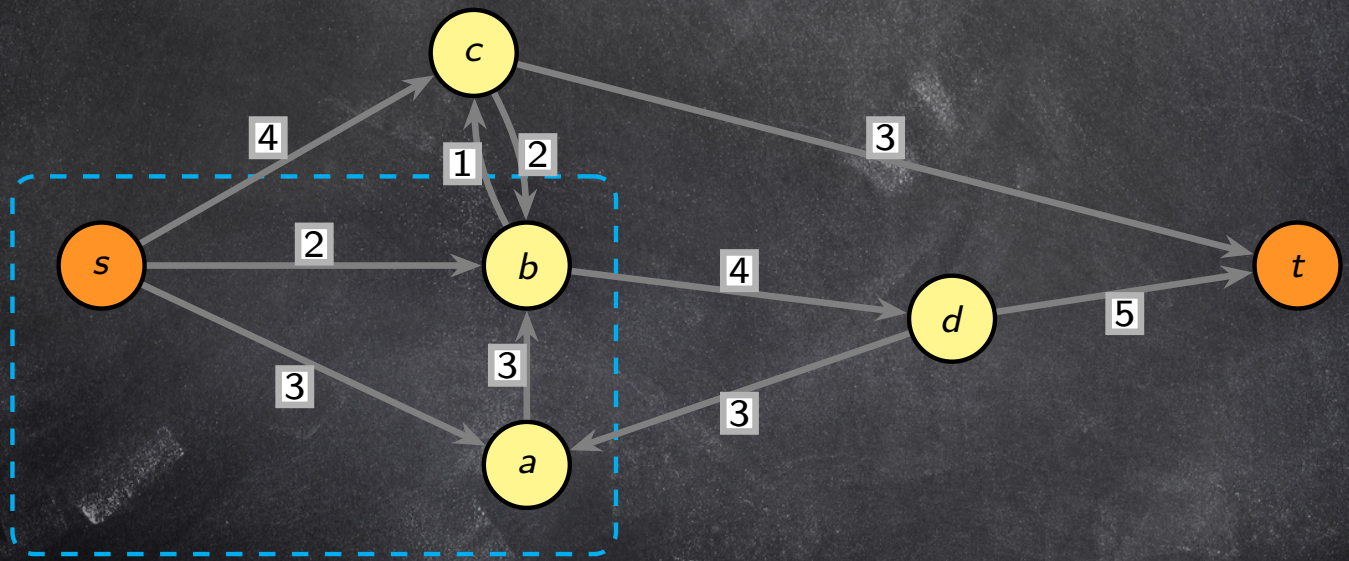
Exemplo: capacidade do corte

Para $S = \{s, a, b\}$, temos:

$$(S, S')^+ = \{sc, bc, bd\}.$$

Logo:

$$c(S, S') = c(sc) + c(bc) + c(bd) = 4 + 1 + 4 = 9.$$



Luís Felipe

28/05/26

Problema do Corte Mínimo

Problema do Corte Mínimo

Entrada: uma rede $D = (V, E)$ com origem s, destino t e capacidades $c(e)$.

Objetivo: encontrar um corte $[S, S']$ com menor capacidade possível.

Isto é, queremos minimizar:

$$c(S, S') = \sum_{e \in (S,S')^+} c(e).$$

Luís Felipe

28/05/26

Fluxo através de um corte

O fluxo através de um corte $[S, S']$ é:

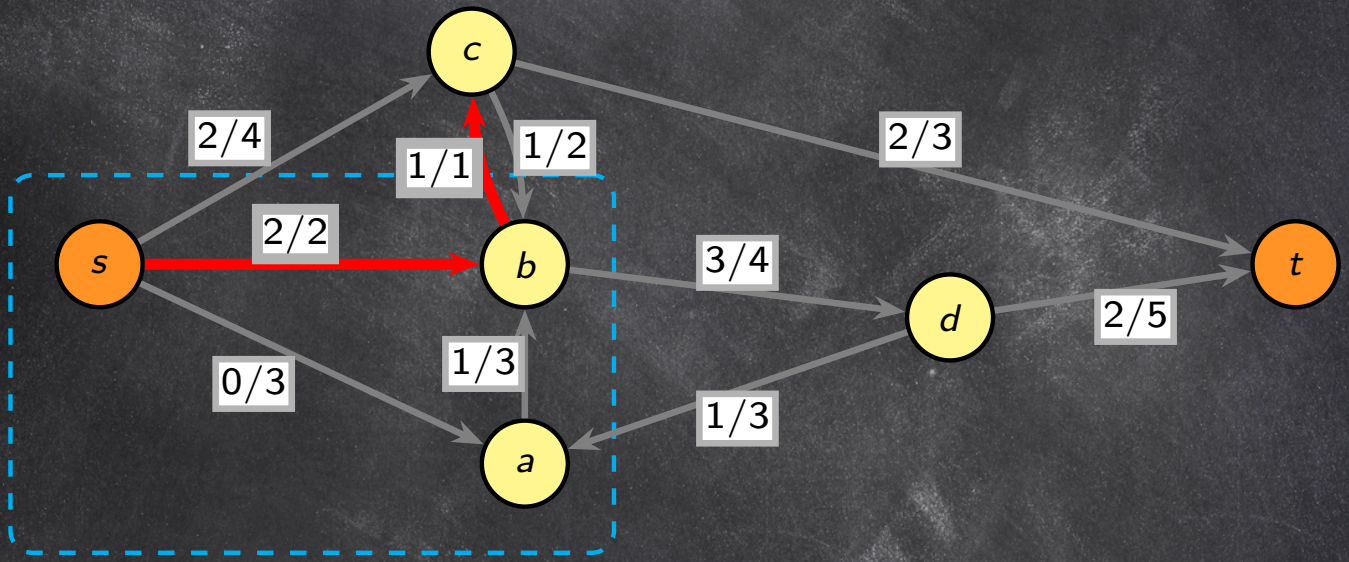
$$f(S, S') = \sum_{e \in (S,S')^+} f(e) - \sum_{e \in (S,S')^-} f(e).$$

Ou seja:

- somamos o fluxo que sai de S para S';
- subtraímos o fluxo que volta de S' para S.

Exemplo: fluxo no corte

Considere o fluxo abaixo e $S = \{s, a, b\}$.



Temos:

$$f(S, S') = f(sc) + f(bc) + f(bd) - f(cb) - f(da).$$

Logo:

$$f(S, S') = 2 + 1 + 3 - 1 - 1 = 4 = f(D).$$

Atenção: $f(D)$ é o valor do fluxo f considerado, não necessariamente o valor máximo possível da rede.

Luís Felipe

28/05/26

Fluxo em qualquer corte

Teorema. Seja f um fluxo em uma rede D e seja $[S, S']$ um corte qualquer. Então:

$$f(S, S') = f(D).$$

Ideia da prova:
Ao somar as equações de conservação para todos os vértices de $S \setminus \{s\}$, os fluxos internos a S se cancelam.

O que sobra é exatamente:

$$\text{fluxo que sai de } S - \text{fluxo que entra em } S.$$

Esse valor é igual ao fluxo que sai da fonte s, isto é, $f(D)$.

Luís Felipe

28/05/26

Todo corte limita todo fluxo

Teorema. Seja f um fluxo qualquer em D e seja $[S, S']$ um corte qualquer. Então:

$$f(D) \leq c(S, S').$$

Prova:

$$f(D) = f(S, S') = \sum_{e \in (S,S')^+} f(e) - \sum_{e \in (S,S')^-} f(e).$$

Como $f(e) \geq 0$:

$$f(D) \leq \sum_{e \in (S,S')^+} f(e).$$

Como $f(e) \leq c(e)$:

$$f(D) \leq \sum_{e \in (S,S')^+} c(e) = c(S, S').$$

Luís Felipe

28/05/26

Consequência importante

Para qualquer fluxo f e qualquer corte $[S, S']$:

$$f(D) \leq c(S, S').$$

Logo, podemos interpretar essa desigualdade de duas formas:

1. todo corte fornece um limite superior para o valor de qualquer fluxo;
2. todo fluxo fornece um limite inferior para a capacidade de qualquer corte.

Corolário: Seja f um fluxo em uma rede D e seja $[S, S']$ um corte. Se

$$f(D) = c(S, S'),$$

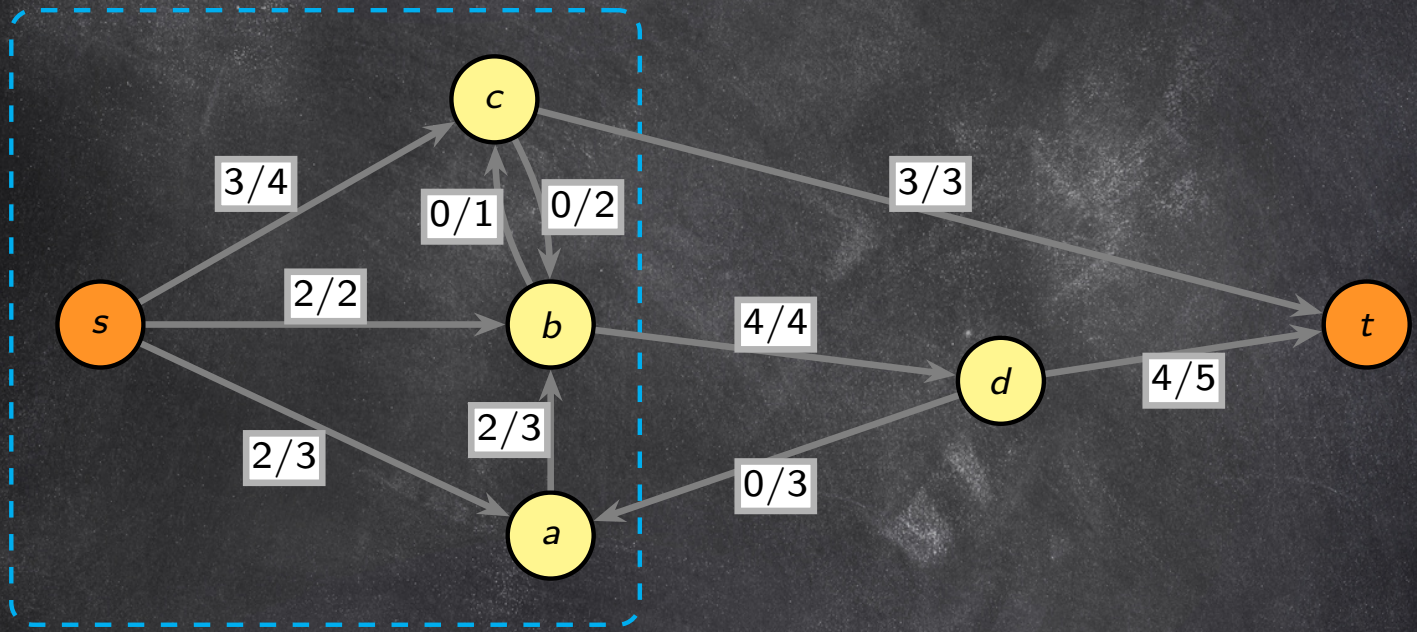
então f é um fluxo máximo e $[S, S']$ é um corte mínimo.

Luís Felipe

28/05/26

Fluxo máximo e corte mínimo

No exemplo abaixo, o fluxo tem valor 7.



Para $S = \{s, a, b, c\}$:

$$c(S, S') = c(bd) + c(ct) = 4 + 3 = 7.$$

Como $f(D) = 7 = c(S, S')$, o fluxo é máximo e o corte é mínimo.

Luís Felipe

28/05/26

Arestas diretas e contrárias

Dado um fluxo f em uma rede D, uma aresta $e = (x, y)$ é:

- direta se ainda podemos aumentar fluxo nela:

$$c(e) - f(e) > 0;$$

- contrária se podemos reduzir fluxo nela:

$$f(e) > 0.$$

Uma mesma aresta pode ser simultaneamente direta e contrária quando:

$$0 < f(e) < c(e).$$

Luís Felipe

28/05/26

Rede residual

Dada uma rede D e um fluxo f, a rede residual D_f representa todas as possíveis variações de fluxo.

Para cada aresta $(x, y) \in E$:

- se $c(x, y) - f(x, y) > 0$, adicionamos a aresta residual (x, y) com capacidade:

$$c_f(x, y) = c(x, y) - f(x, y);$$

- se $f(x, y) > 0$, adicionamos a aresta residual (y, x) com capacidade:

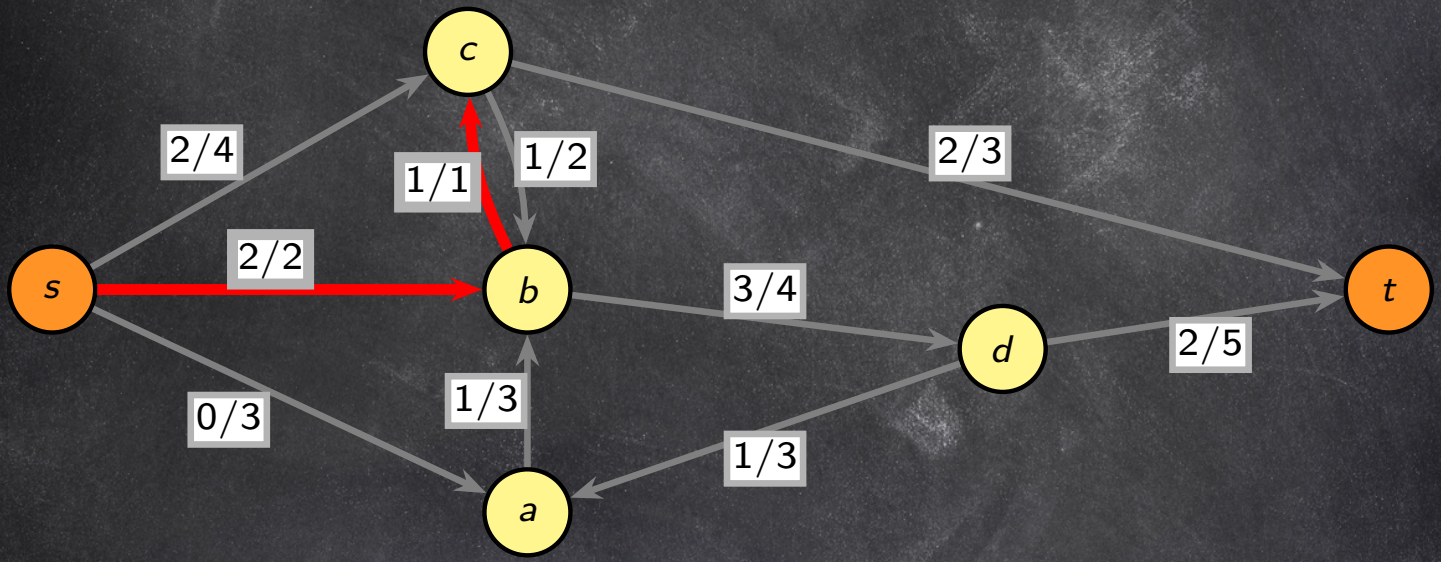
$$c_f(y, x) = f(x, y).$$

Luís Felipe

28/05/26

Rede residual: vamos construir passo a passo

Considere a rede D abaixo com um fluxo f de valor 4.



A rede residual D_f mostra quanto ainda podemos aumentar fluxo e quanto podemos desfazer do fluxo já enviado.

Luís Felipe

28/05/26

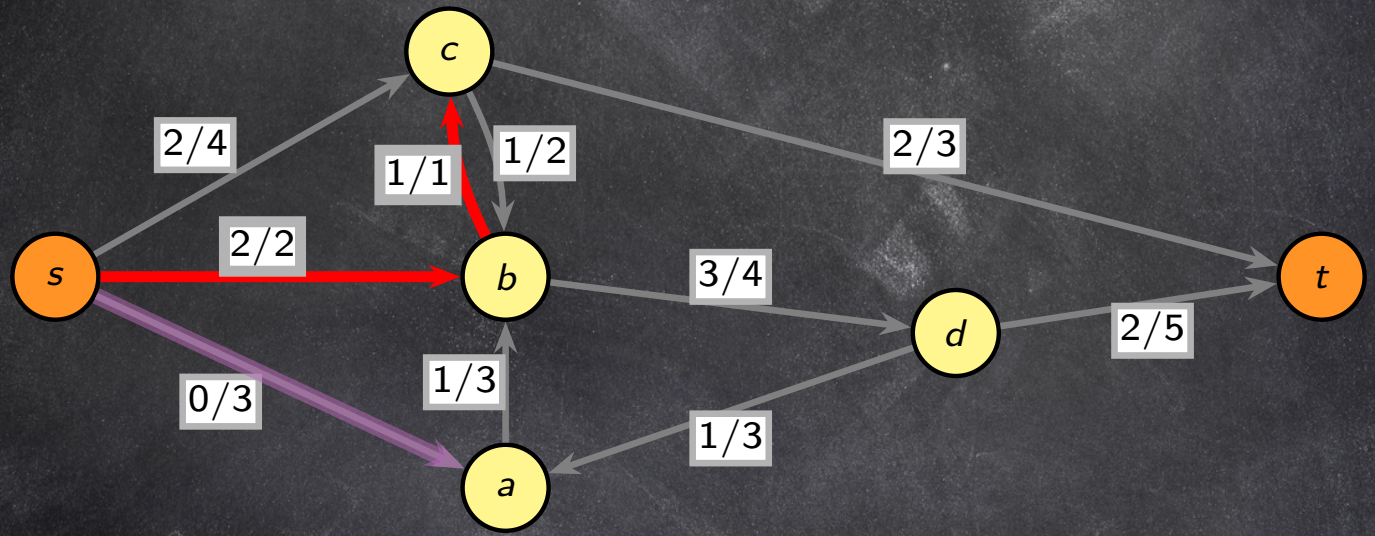
Passo 1: aresta direta

Se uma aresta (x, y) ainda tem capacidade sobrando, isto é,

$$c(x, y) - f(x, y) > 0,$$

então colocamos em D_f uma aresta residual no mesmo sentido:

$$(x, y) \quad \text{com capacidade residual} \quad c_f(x, y) = c(x, y) - f(x, y).$$



Na aresta (s, a), temos $f(s, a) = 0$ e $c(s, a) = 3$. Logo:

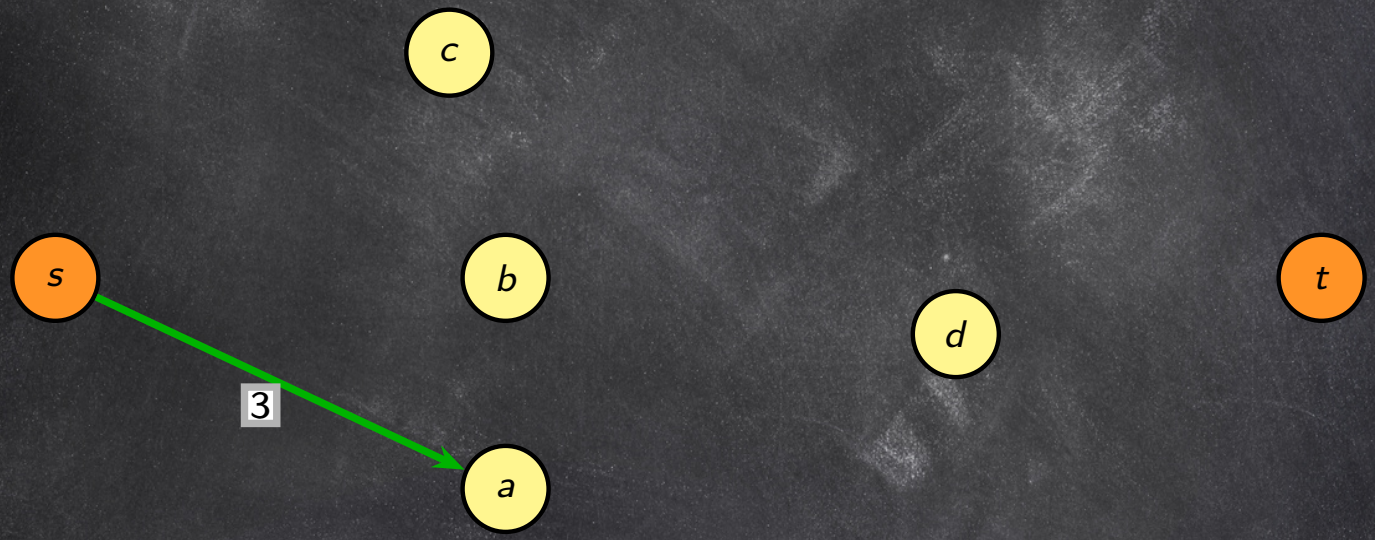
$$c_f(s, a) = 3 - 0 = 3.$$

Luís Felipe

28/05/26

Passo 1: aresta direta em D_f

A aresta (s, a) gera a seguinte aresta residual:



Interpretação:

Ainda podemos enviar até 3 unidades adicionais de fluxo pela aresta original (s, a).

Luís Felipe

28/05/26

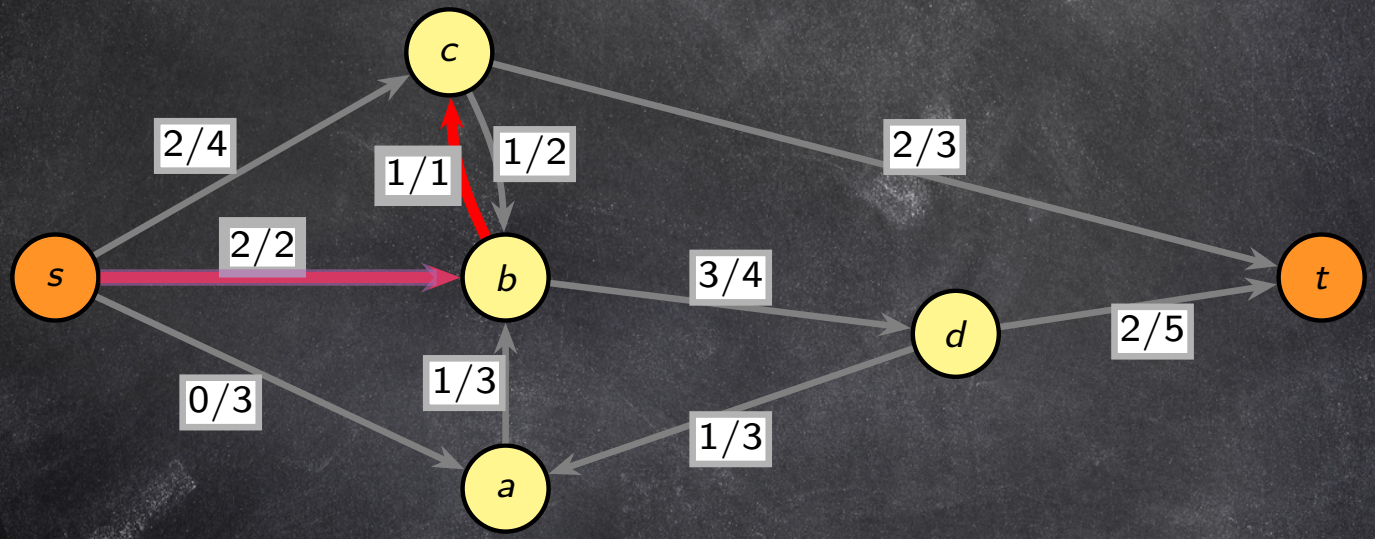
Passo 2: aresta contrária

Se uma aresta (x, y) já tem fluxo positivo, isto é,

$$f(x, y) > 0,$$

então colocamos em D_f uma aresta residual no sentido contrário:

$$(y, x) \quad \text{com capacidade residual} \quad c_f(y, x) = f(x, y).$$



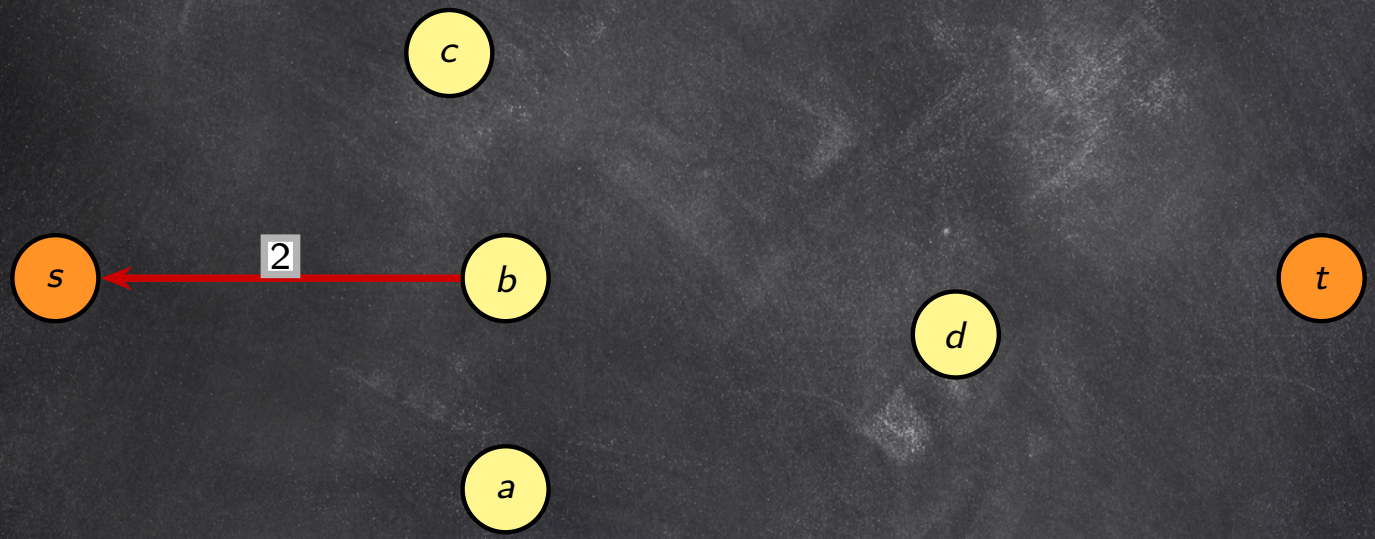
Na aresta (s, b), temos $f(s, b) = 2$. Logo, podemos desfazer até 2 unidades desse fluxo.

Luís Felipe

28/05/26

Passo 2: aresta contrária em D_f

A aresta original (s, b) gera a aresta residual contrária (b, s):



Interpretação:

No residual, usar (b, s) significa reduzir o fluxo atual em (s, b).

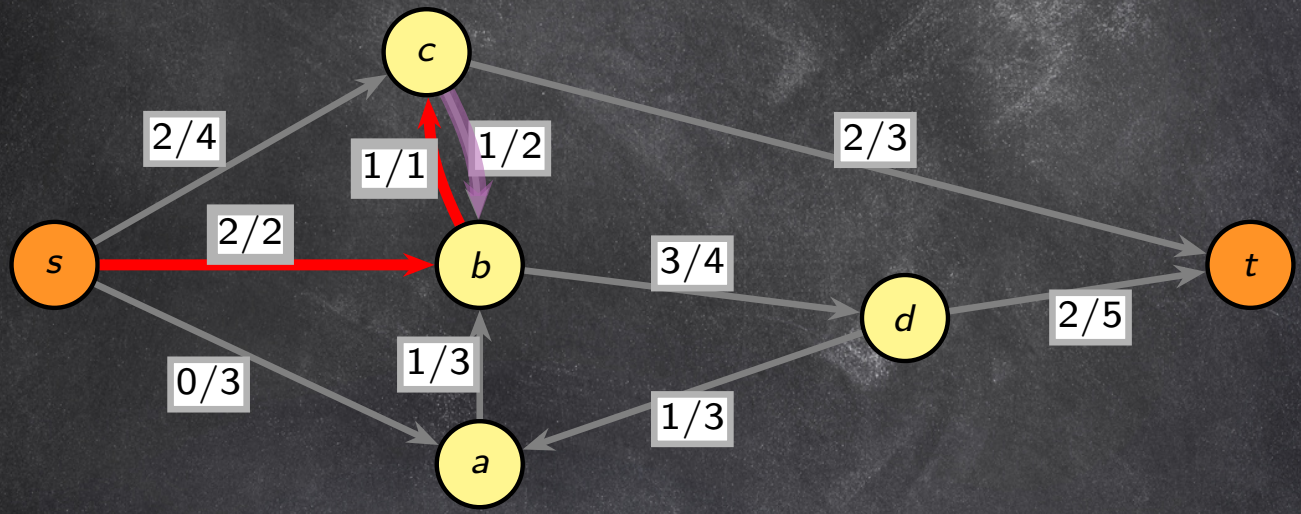
Luís Felipe

28/05/26

Passo 3: aresta direta e contrária ao mesmo tempo

Uma mesma aresta original pode gerar duas arestas residuais.
Isso acontece quando:

$$0 < f(x, y) < c(x, y).$$



Na aresta (c, b), temos:

$$f(c, b) = 1, \qquad c(c, b) = 2.$$

Logo, há capacidade para aumentar e também para reduzir fluxo.

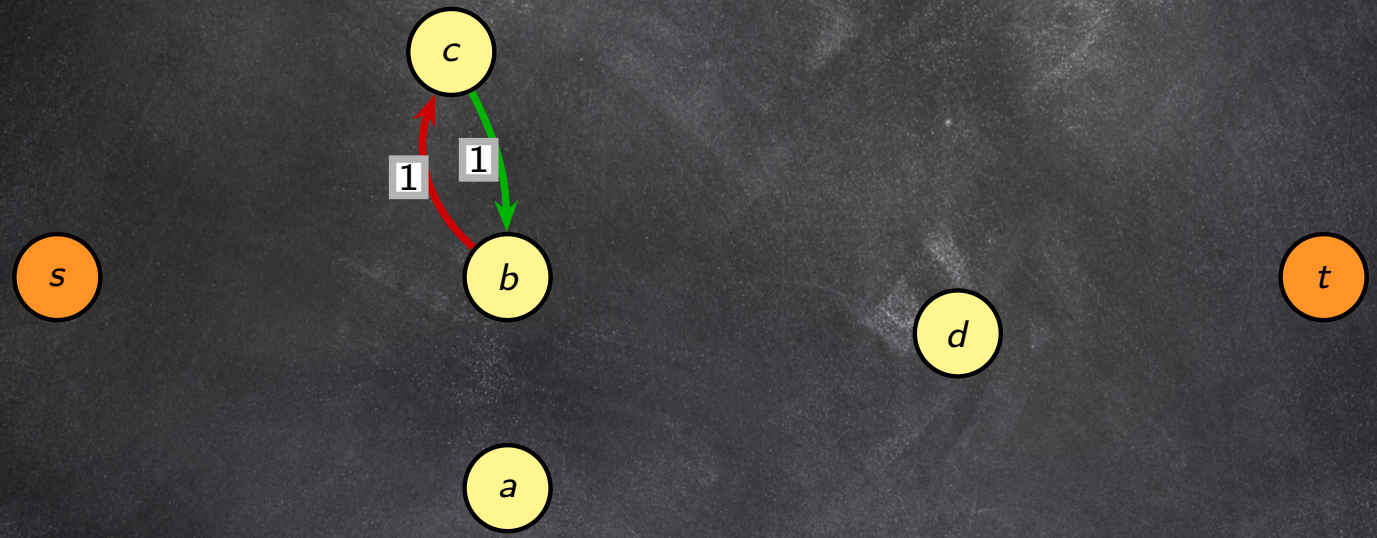
Luís Felipe

28/05/26

Passo 3: duas arestas residuais

A aresta original (c, b) gera:

- uma aresta residual direta (c, b) com capacidade $2 - 1 = 1$;
- uma aresta residual contrária (b, c) com capacidade 1.



A aresta direta representa aumento de fluxo; a contrária representa redução.

Luís Felipe

28/05/26

Aplicar a regra a todas as arestas

Aresta em D	f/c	Residual direta	Residual contrária
(s, a)	$0/3$	(s, a) com 3	–
(s, b)	$2/2$	–	(b, s) com 2
(s, c)	$2/4$	(s, c) com 2	(c, s) com 2
(a, b)	$0/3$	(a, b) com 3	–
(b, c)	$1/1$	–	(c, b) com 1
(c, b)	$1/2$	(c, b) com 1	(b, c) com 1
(b, d)	$3/4$	(b, d) com 1	(d, b) com 3
(d, a)	$1/3$	(d, a) com 2	(a, d) com 1
(c, t)	$2/3$	(c, t) com 1	(t, c) com 2
(d, t)	$2/5$	(d, t) com 3	(t, d) com 2

Cada linha da tabela vem diretamente das duas regras:

$$c_f(x, y) = c(x, y) - f(x, y), \qquad c_f(y, x) = f(x, y).$$

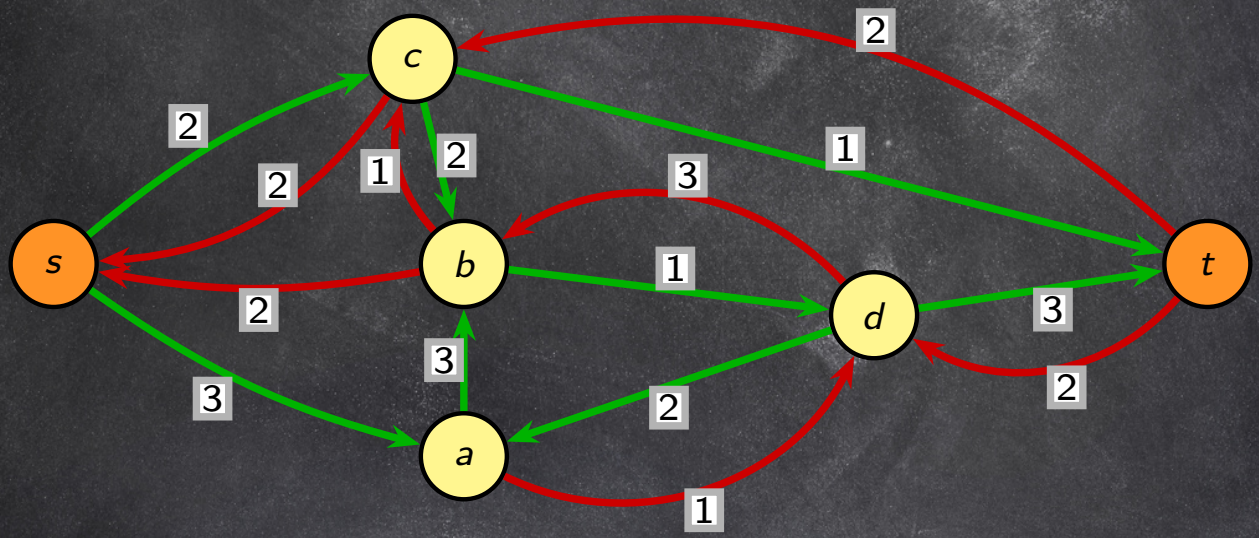
Observação: se duas regras geram arestas residuais com o mesmo sentido, suas capacidades devem ser somadas na rede residual final.

Luís Felipe

28/05/26

Rede residual completa

Aplicando as duas regras a todas as arestas, para o dado fluxo atual f, obtemos D_f.



Verde: arestas residuais de aumento de fluxo.
Vermelho: arestas residuais de redução de fluxo.

Luís Felipe

28/05/26

Interpretação da rede residual

A rede residual é um mapa das possíveis variações de fluxo.

- Uma aresta residual no mesmo sentido de uma aresta original indica que podemos aumentar fluxo nessa aresta.
- Uma aresta residual no sentido contrário indica que podemos reduzir fluxo na aresta original.

Essa possibilidade de reduzir fluxo é essencial.

Ela permite desfazer escolhas feitas anteriormente e redirecionar o fluxo por caminhos melhores.

Luís Felipe

28/05/26

Caminho aumentante e gargalo

Um caminho aumentante na rede residual é um caminho de s a t na rede residual D_f.

Se existe um caminho aumentante, então ainda há uma forma de aumentar o valor do fluxo.

O aumento possível ao longo desse caminho é limitado pela menor capacidade residual entre suas arestas.

O gargalo de um caminho aumentante P em D_f é:

$$g(P) = \min_{e \in E(P)} c_f(e).$$

Ou seja, $g(P)$ é a menor capacidade residual ao longo do caminho.

Podemos aumentar o fluxo total exatamente em $g(P)$ unidades usando esse caminho.

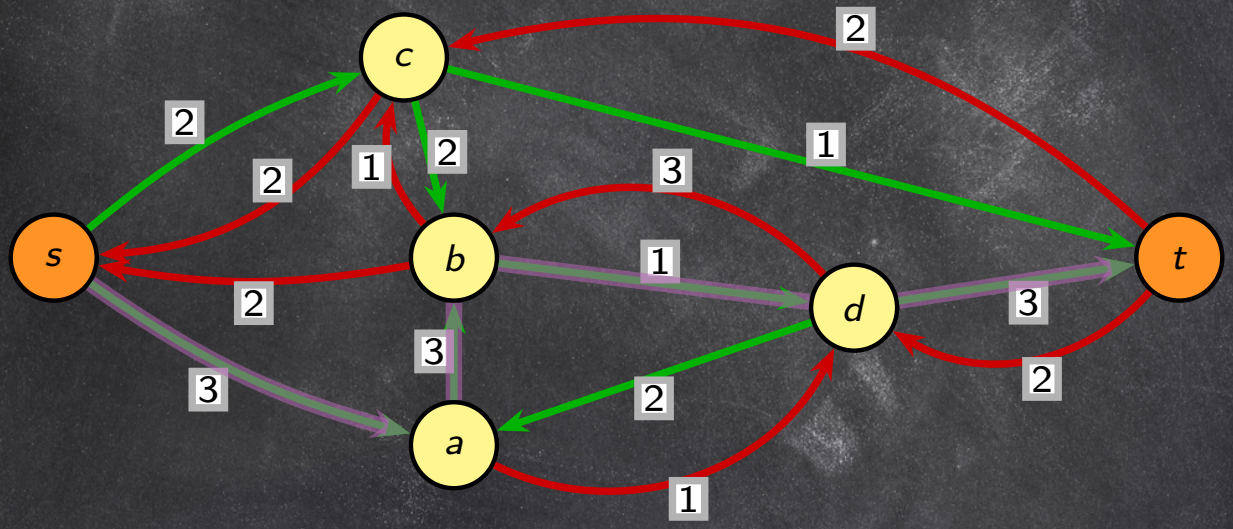
Luís Felipe

28/05/26

Exemplo de caminho aumentante

Na rede residual abaixo, considere o caminho:

$$P = s, a, b, d, t.$$



As capacidades residuais ao longo do caminho P são:

$$3, 3, 1, 3.$$

Logo, o gargalo do caminho P é:

$$g(P) = 1.$$

Luís Felipe

28/05/26

Lema do caminho aumentante

Lema. Dados uma rede D e um fluxo f, se há um caminho aumentante em D_f com gargalo g, então existe um novo fluxo f_{novo} tal que:

$$f_{novo}(D) = f(D) + g.$$

Atualização:

- se a aresta residual corresponde a aumentar fluxo em uma aresta original, somamos g;
- se a aresta residual corresponde a reduzir fluxo em uma aresta original, subtraímos g.

Luís Felipe

28/05/26

Atualizando o fluxo

Se o caminho aumentante na rede residual é:

$$P = e'_1, e'_2, \ldots, e'_k$$

com gargalo g, então atualizamos o fluxo considerando todas as arestas residuais do caminho.

Para cada aresta residual e'_j de P:

- se $e'_j = (x, y)$ veio da aresta original (x, y), então:

$$f(x, y) \leftarrow f(x, y) + g;$$

- se $e'_j = (y, x)$ veio da aresta original (x, y), então:

$$f(x, y) \leftarrow f(x, y) - g.$$

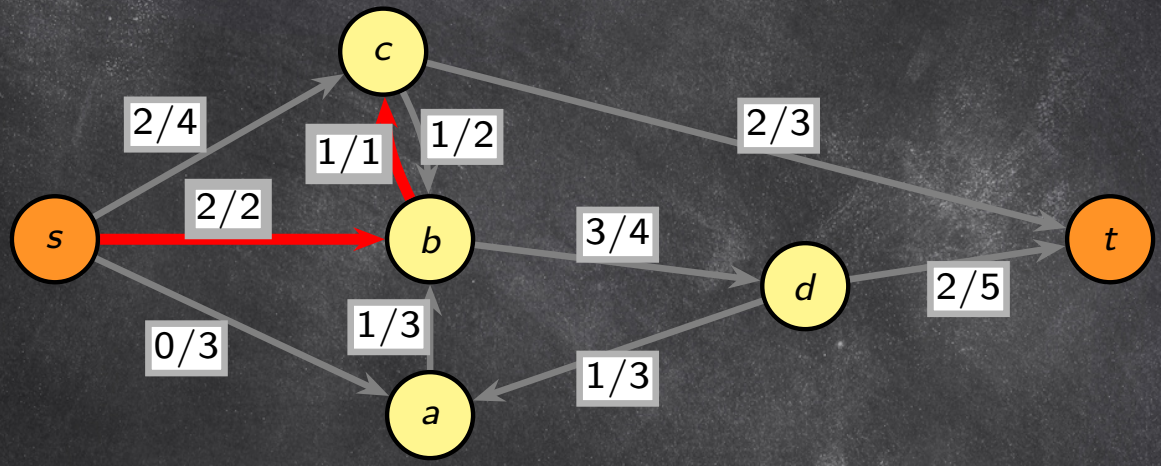
As arestas que não correspondem a nenhuma aresta de P mantêm o mesmo valor de fluxo.

Luís Felipe

28/05/26

Exemplo de aumento

Fluxo inicial com valor 4:



Considere o caminho aumentante na residual:

$$P = s, a, b, d, t, \qquad g = 1.$$

Depois da atualização, o novo fluxo terá valor:

$$4 + 1 = 5.$$

Luís Felipe

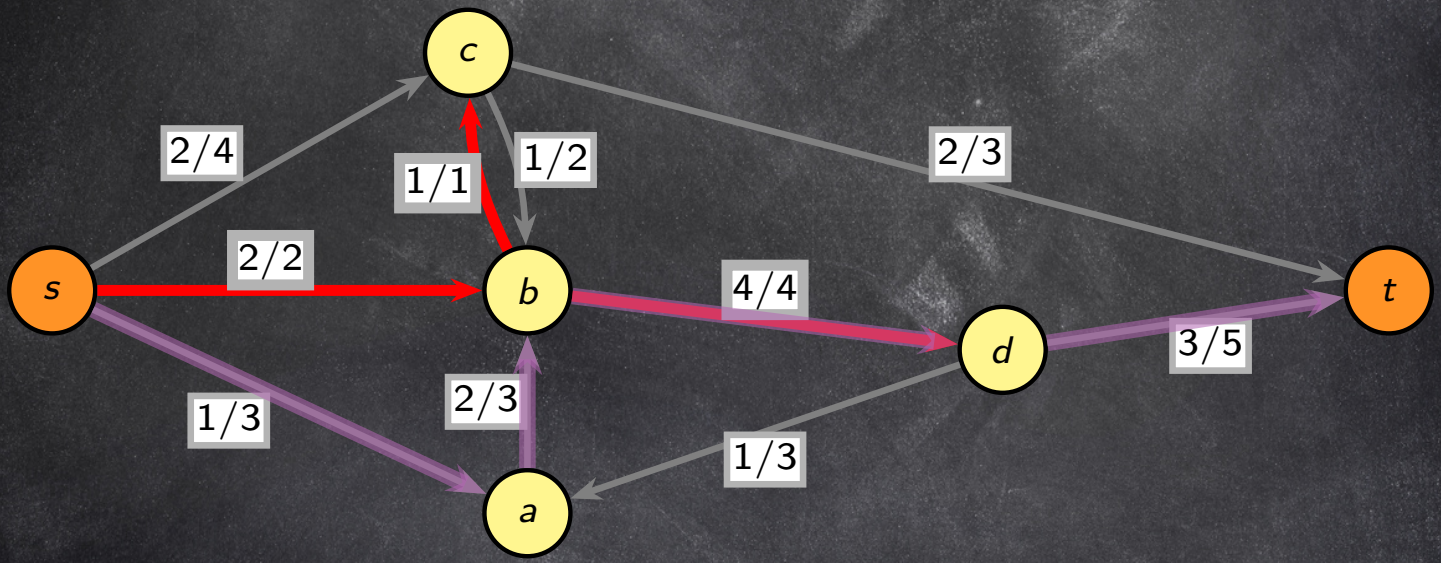
28/05/26

Exemplo de aumento: novo fluxo

Após aumentar $g = 1$ unidade pelo caminho residual

$$P = s, a, b, d, t,$$

obtemos o seguinte fluxo:



$$f_{\text{novo}}(D) = f(D) + g = 4 + 1 = 5.$$

Luís Felipe

28/05/26

Caracterização por caminhos aumentantes

Teorema. Dados uma rede D e um fluxo f:

$$f \text{ é máximo} \iff \text{não existe caminho aumentante em } D_f.$$

Esse é o análogo, para fluxos, do Teorema de Berge para emparelhamentos.

Enquanto existir caminho aumentante, ainda podemos aumentar o valor do fluxo.

Quando não existe caminho aumentante, o fluxo é máximo.

Luís Felipe

28/05/26

Fluxo máximo - corte mínimo

Teorema (Fluxo Máximo - Corte Mínimo).

Em toda rede, o valor de um fluxo máximo é igual à capacidade de um corte mínimo:

$$\max_{f} f(D) = \min_{(S,S')} c(S, S').$$

Ou seja:

o maior valor que conseguimos enviar de s para t é exatamente a menor capacidade que separa s de t.

Luís Felipe

28/05/26

Prova: fluxo máximo - corte mínimo

Já sabemos que, para qualquer fluxo f e qualquer corte $[S, S']$:

$$f(D) \leq c(S, S').$$

Logo:

$$\max_{f} f(D) \leq \min_{(S,S')} c(S, S').$$

Falta mostrar que existe um fluxo e um corte com igualdade.

Para isso, tomamos um fluxo máximo f e olhamos a rede residual D_f.

Luís Felipe

28/05/26

Prova: construindo o corte

Se f é máximo, então não há caminho aumentante em D_f.

Seja S o conjunto dos vértices alcançáveis a partir de s em D_f.

Como não há caminho aumentante, temos:

$$t \notin S.$$

Logo, $(S, V \setminus S)$ é um corte.

Vamos mostrar que:

$$f(D) = c(S, V \setminus S).$$

Luís Felipe

28/05/26

Prova: saturação no corte

Se $u \in S$ e $v \notin S$, então não pode existir aresta residual (u, v).

Portanto, toda aresta original $(u, v) \in (S, S')^+$ deve estar saturada:

$$f(u, v) = c(u, v).$$

Além disso, se $v \notin S$ e $u \in S$, então uma aresta original $(v, u) \in (S, S')^-$ não pode ter fluxo positivo.

Caso contrário, existiria a aresta residual (u, v).

Logo, para toda aresta em $(S, S')^-$:

$$f(v, u) = 0.$$

Luís Felipe

28/05/26

Prova: concluindo

Como toda aresta de $(S, S')^+$ está saturada e toda aresta de $(S, S')^-$ tem fluxo zero:

$$\begin{aligned} f(D) &= f(S, S') \\ &= \sum_{e \in (S,S')^+} f(e) - \sum_{e \in (S,S')^-} f(e) \\ &= \sum_{e \in (S,S')^+} c(e) - 0 \\ &= c(S, S'). \end{aligned}$$

Assim, f é máximo e $[S, S']$ é mínimo.

Luís Felipe

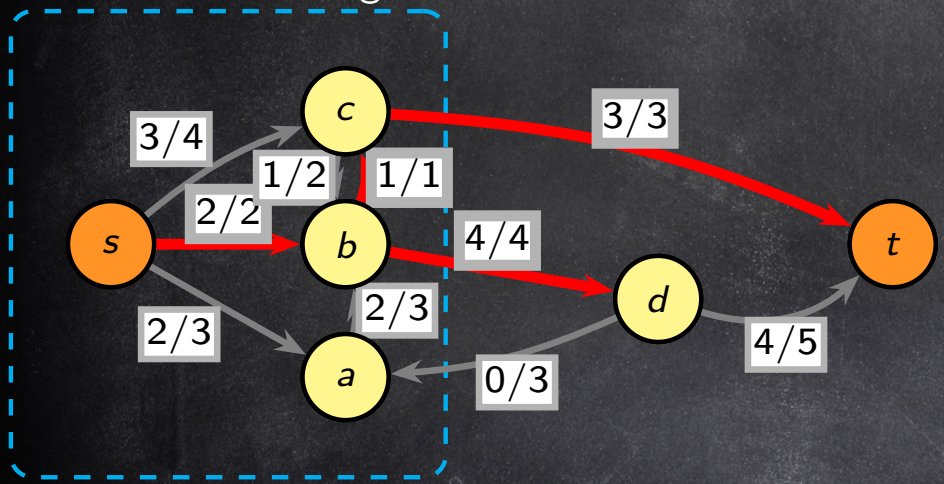
28/05/26

Como obter um corte mínimo pelo residual

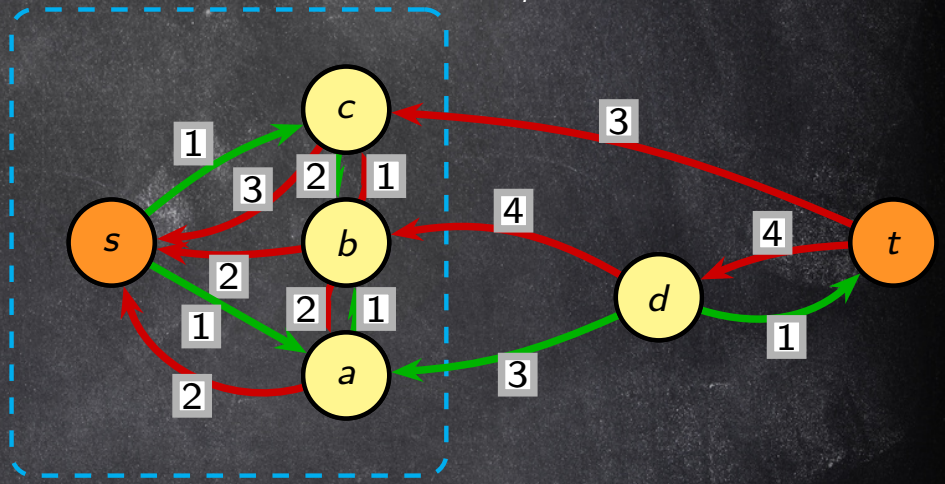
Depois de obter um fluxo máximo f, i.e., não existe mais caminho de s para t em D_f:

- construa a rede residual D_f;
- faça uma busca a partir de s em D_f;
- seja S o conjunto dos vértices alcançados;
- então $(S, V \setminus S)$ é um corte mínimo.

Grafo original com o fluxo máximo f



Rede residual D_f



No residual do fluxo máximo, os vértices alcançáveis a partir de s são:
$S = \{s, a, b, c\}$, $S' = \{d, t\}$.
As arestas do corte que saem de S para S' são: $(S, S')^+ = \{bd, ct\}$.
Logo: $c(S, S') = c(bd) + c(ct) = 4 + 3 = 7$.

Luís Felipe

28/05/26

Ford-Fulkerson

```
1  funcao FordFulkerson(D, s, t):
2
3      f <- fluxo zero
4
5      enquanto existe caminho P de s a t na rede residual D_f faca
6
7          g <- gargalo de P
8
9          para cada aresta residual e em P faca
10
11             se e eh aresta de aumento entao
12                 aumente em g o fluxo na aresta original
13             senao
14                 reduza em g o fluxo na aresta original
15
16     retornar f
```

Luís Felipe

28/05/26

Complexidade de Ford-Fulkerson

Se todas as capacidades são inteiras, cada aumento incrementa o valor do fluxo em pelo menos 1.

Se F é o valor do fluxo máximo, então há no máximo F aumentos.

Encontrar um caminho aumentante e atualizá-lo custa $O(m)$.

Portanto, a complexidade é:

$$O(mF).$$

Essa cota depende do valor numérico F, e não apenas de n e m.

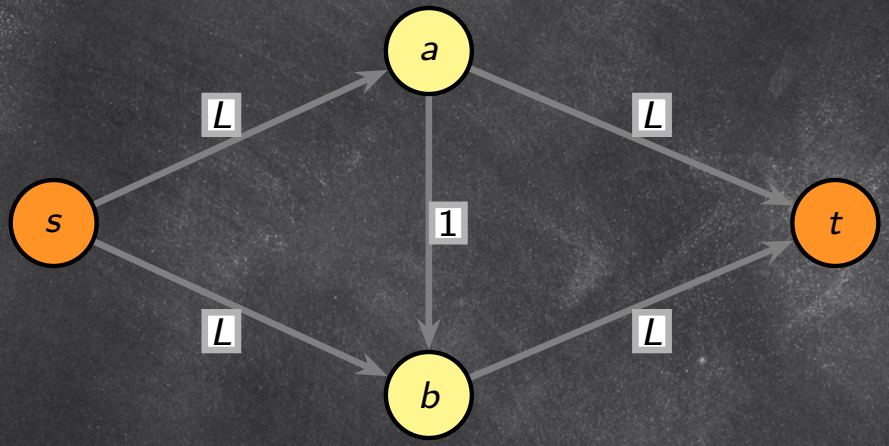
Essa complexidade é pseudo-polinomial, pois depende do valor numérico de F, e não apenas do tamanho da entrada em bits.

Luís Felipe

28/05/26

Um caso simples, mas ruim

Considere a rede:



Há duas rotas largas:

$$s, a, t \quad \text{e} \quad s, b, t,$$

ambas com capacidade L.

Mas existe também a aresta estreita:

$$ab$$

com capacidade 1.

Luís Felipe

28/05/26

Escolhas ruins de caminhos

Se Ford-Fulkerson escolher alternadamente caminhos que usam a aresta ab, por exemplo:

$$s, a, b, t$$

e depois um caminho que desfaz parte dessa escolha, cada aumento pode ter gargalo 1.

Nesse caso, o algoritmo pode precisar de:

$$2L$$

iterações.

Mesmo que o fluxo máximo tenha valor $2L$ e possa ser obtido em apenas duas boas escolhas.

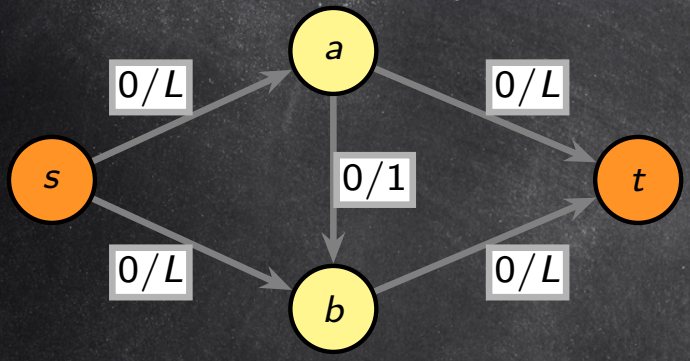
Luís Felipe

28/05/26

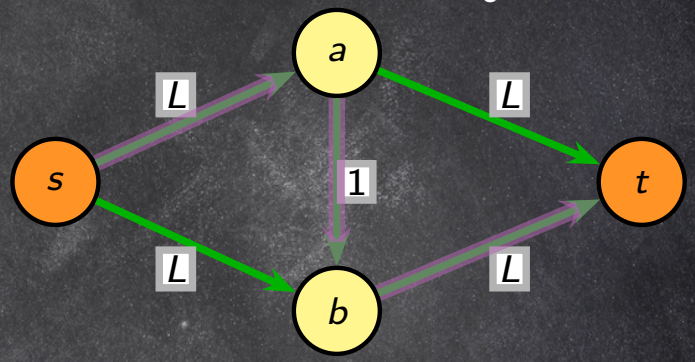
Ford–Fulkerson: primeira escolha ruim

Considere o fluxo inicial f_0 de valor zero.

Fluxo atual f_0



Rede residual D_{f_0}



Suponha que o algoritmo escolha:

$$P_1 = s, a, b, t.$$

O gargalo é:

$$g = \min\{L, 1, L\} = 1.$$

Atualizamos:

$$f(sa) \leftarrow 1, \qquad f(ab) \leftarrow 1, \qquad f(bt) \leftarrow 1.$$

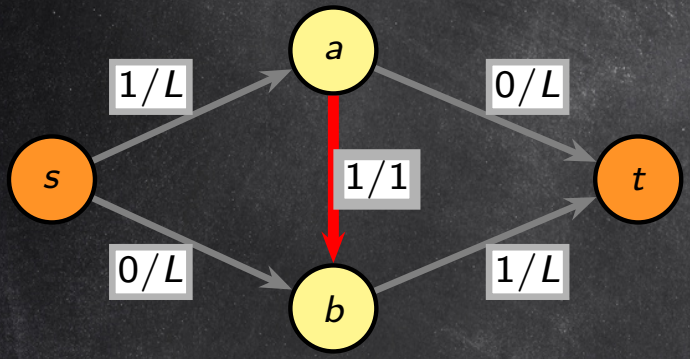
Luís Felipe

28/05/26

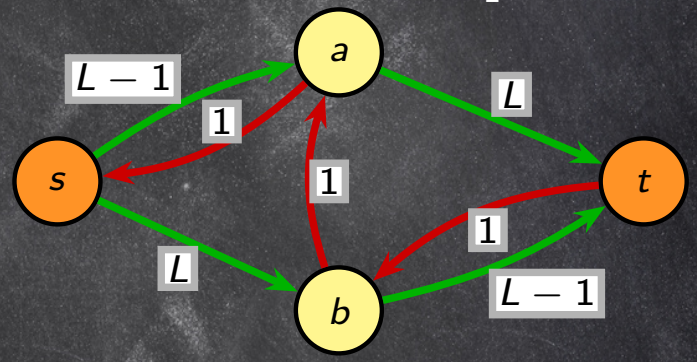
Depois da primeira atualização

Após aumentar 1 unidade por s, a, b, t, temos:

Novo fluxo f_1



Rede residual D_{f_1}



Aparece a aresta residual contrária:

$$b \to a,$$

pois há fluxo positivo em $a \to b$.

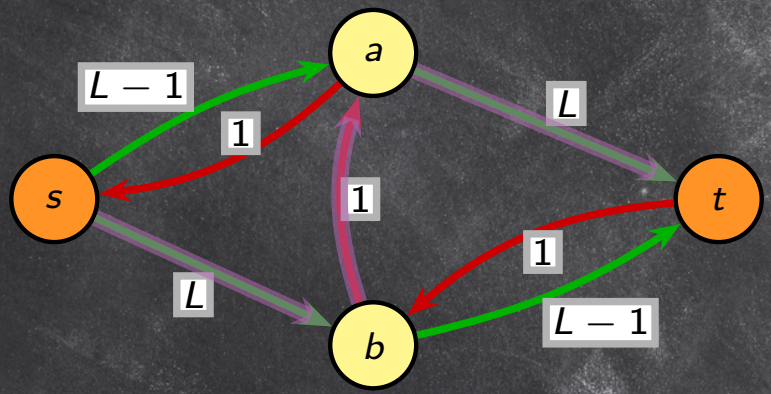
Luís Felipe

28/05/26

Ford–Fulkerson: segunda escolha ruim

Na residual D_{f_1}, o algoritmo pode escolher:

$$P_2 = s, b, a, t.$$



O gargalo novamente é:

$$g = 1.$$

Agora:

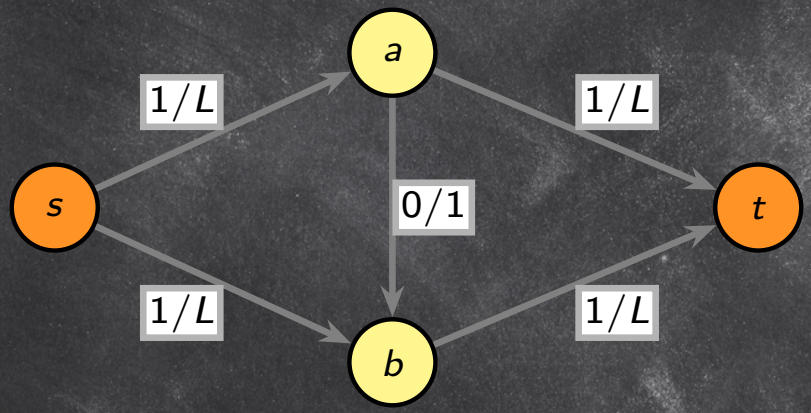
$$f(sb) : 0 \to 1, \qquad f(ab) : 1 \to 0, \qquad f(at) : 0 \to 1.$$

Luís Felipe

28/05/26

Depois da segunda atualização

Após aumentar pelo caminho residual s, b, a, t, obtemos:



O valor do fluxo agora é:

$$2.$$

Em duas iterações, aumentamos o fluxo em 2, mas sempre com gargalo 1.

Esse padrão pode se repetir até atingir o valor máximo $2L$.

Luís Felipe

28/05/26

Como evitar escolhas ruins?

A ideia é escolher caminhos aumentantes mais curtos.

Em vez de escolher qualquer caminho aumentante, escolhemos um caminho com menor número de arestas na rede residual.

Isso pode ser feito com busca em largura.

Essa versão é chamada de Edmonds-Karp.

Luís Felipe

28/05/26

Edmonds-Karp

```
1  funcao EdmondsKarp(D, s, t):
2
3      f <- fluxo zero
4
5      enquanto existe caminho P de s a t em D_f encontrado por BFS faca
6
7          g <- gargalo de P
8
9          aumente o fluxo ao longo de P em g
10
11         atualize a rede residual D_f
12
13     retornar f
```

A diferença para Ford-Fulkerson é a escolha do caminho aumentante: sempre usamos BFS.

Luís Felipe

28/05/26

Complexidade de Edmonds-Karp

Em Edmonds-Karp, cada caminho aumentante é um caminho mínimo em número de arestas na rede residual.

Cada BFS custa:

$$O(m).$$

O número total de aumentos é:

$$O(nm).$$

Logo, a complexidade total é:

$$O(nm^2).$$

Essa cota é polinomial e não depende do valor numérico das capacidades.

Luís Felipe

28/05/26

Python: BFS na rede residual

```python
from collections import deque

def bfs(res, s, t):
    pai = {s: None}
    fila = deque([s])

    while fila:
        u = fila.popleft()

        for v, cap in res[u].items():            # v e vizinho de u; cap e a capacidade residual de u->v
            if cap > 0 and v not in pai:
                pai[v] = u

                if v == t:
                    P = []
                    x = t
                    while x != s:
                        P.append((pai[x], x))
                        x = pai[x]
                    P.reverse()
                    return P

                fila.append(v)

    return None
```

Luís Felipe

28/05/26

Python: Edmonds-Karp

```python
def edmonds_karp(G, s, t):
    # G[u][v] = capacidade da aresta uv

    res = {u: {} for u in G}

    for u in G:
        for v, cap in G[u].items():
            res[u][v] = res[u].get(v, 0) + cap
            res.setdefault(v, {})
            res[v].setdefault(u, 0)

    valor = 0

    while True:
        P = bfs(res, s, t)
        if P is None:
            return valor, res

        g = min(res[u][v] for u, v in P)
        valor += g

        for u, v in P:
            res[u][v] -= g      # usa capacidade residual direta
            res[v][u] += g      # cria/aumenta capacidade residual reversa
```

Luís Felipe

28/05/26

Python: recuperando os fluxos

```python
def fluxos_a_partir_residual(G, res):
    fluxo = {}

    for u in G:
        for v, cap in G[u].items():
            # Se res[v][u] aumentou, isto representa fluxo em uv.
            fluxo[(u, v)] = res[v].get(u, 0)

    return fluxo
```

Observação: esta função assume que começamos com fluxo zero e que a rede residual foi atualizada pelo algoritmo anterior.

Luís Felipe

28/05/26

Python: corte mínimo

```
59  def corte_minimo(G, res, s):
60      visitado = {s}
61      fila = deque([s])
62
63      while fila:
64          u = fila.popleft()
65
66          for v, cap in res[u].items():
67              if cap > 0 and v not in visitado:
68                  visitado.add(v)
69                  fila.append(v)
70
71      S = visitado
72      T = set(res) - S
73
74      corte = []
75      capacidade = 0
76
77      for u in S:
78          for v, cap in G.get(u, {}).items():
79              if v in T:
80                  corte.append((u, v))
81                  capacidade += cap
82
83      return S, T, corte, capacidade
```

Luís Felipe

28/05/26

Python: exemplo completo

```python
G = {
    "s": {"a": 3, "b": 2, "c": 4},
    "a": {"b": 3},
    "b": {"c": 1, "d": 4},
    "c": {"b": 2, "t": 3},
    "d": {"a": 3, "t": 5},
    "t": {}
}

valor, res = edmonds_karp(G, "s", "t")
fluxo = fluxos_a_partir_residual(G, res)
S, T, corte, cap = corte_minimo(G, res, "s")

print(valor)      # 7
print(S)          # vertices alcancaveis de s na residual final
print(corte)      # arestas do corte minimo
print(cap)        # 7
```

Luís Felipe

28/05/26

Aplicação: emparelhamento bipartido

Uma aplicação clássica de fluxo máximo é encontrar um emparelhamento máximo em grafos bipartidos.

Seja $G = (X \cup Y, E)$ um grafo bipartido.

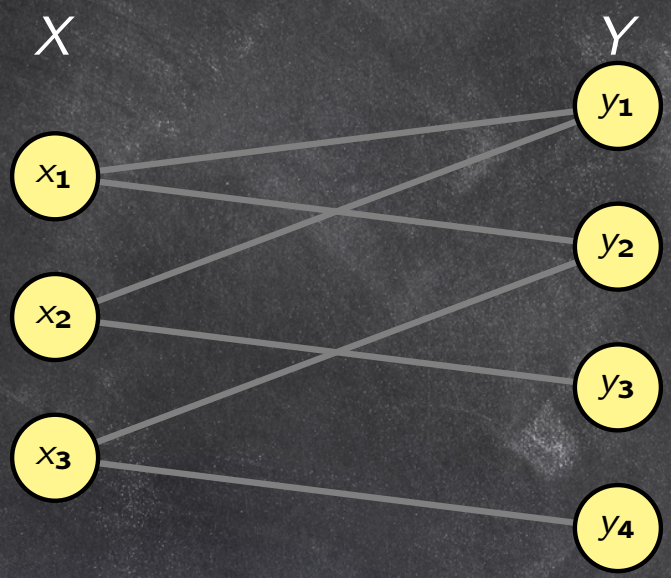
- Cada aresta liga um vértice de X a um vértice de Y.
- Queremos escolher o maior número possível de arestas sem repetir vértices.
- Isto é exatamente o problema de encontrar um emparelhamento máximo.

A ideia é transformar G em uma rede de fluxo.

Luís Felipe

28/05/26

Grafo bipartido de entrada



Queremos encontrar um emparelhamento máximo neste grafo.

Luís Felipe

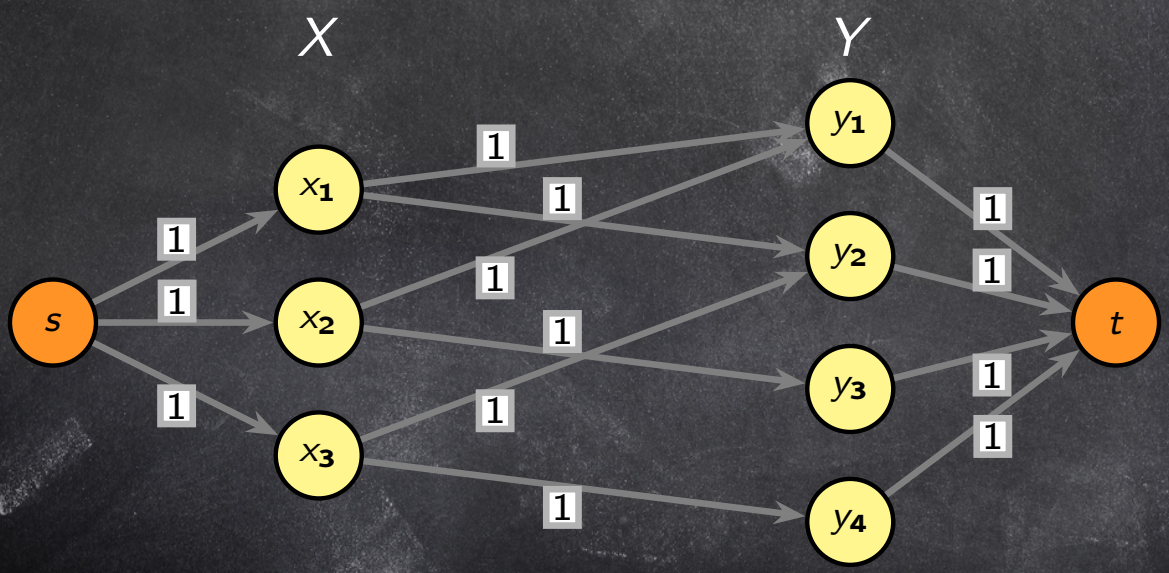
28/05/26

Transformando em rede de fluxo

Construímos uma rede D a partir de $G = (X \cup Y, E)$.

- Adicionamos uma origem s e um destino t.
- Para cada $x \in X$, adicionamos uma aresta sx com capacidade 1.
- Para cada aresta $xy \in E(G)$, adicionamos uma aresta dirigida xy com capacidade 1.
- Para cada $y \in Y$, adicionamos uma aresta yt com capacidade 1.

Assim, todo caminho de s até t tem a forma: s, x, y, t.



Todas as capacidades são iguais a 1.

Por que isso funciona?

- Como cada aresta sx tem capacidade 1, cada vértice $x \in X$ pode ser usado por no máximo uma unidade de fluxo.
- Como cada aresta yt tem capacidade 1, cada vértice $y \in Y$ pode ser usado por no máximo uma unidade de fluxo.
- Cada unidade de fluxo que vai de s até t escolhe um caminho:

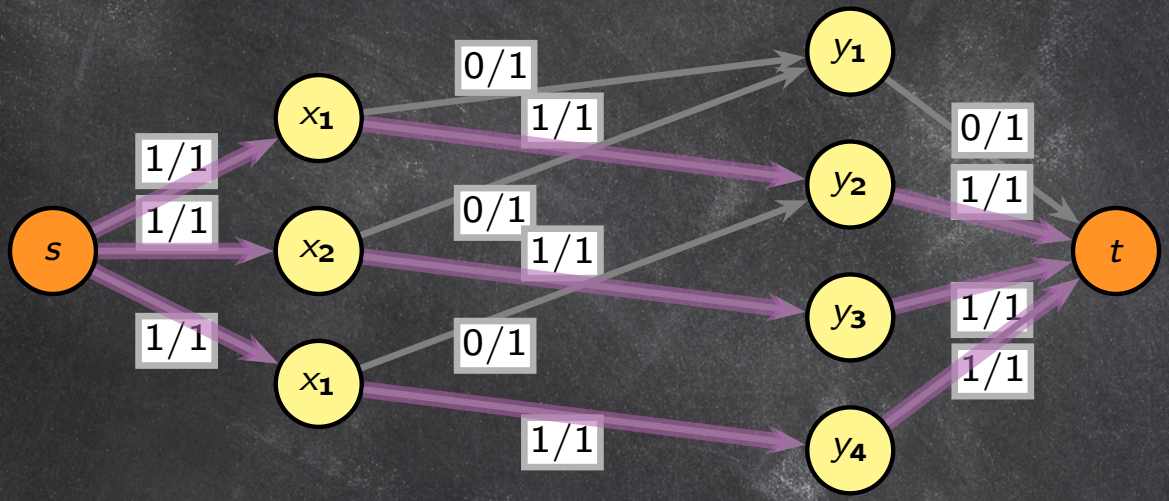
$$s, x, y, t.$$

- Esse caminho representa a escolha da aresta xy no emparelhamento.
- Portanto, um fluxo de valor k corresponde a um emparelhamento de tamanho k.

Luís Felipe

28/05/26

Exemplo: fluxo e emparelhamento



Este fluxo tem valor 3 e corresponde ao emparelhamento:

$$M = \{x_1y_2,\ x_2y_3,\ x_3y_4\}.$$

Luís Felipe

28/05/26

Do fluxo para o emparelhamento

Depois de calcular um fluxo máximo f, obtemos o emparelhamento:

$$M = \{xy \in E(G) : f(x, y) = 1\}.$$

Ou seja, olhamos apenas para as arestas entre X e Y que receberam uma unidade de fluxo.

Como as capacidades das arestas que saem de s e entram em t são iguais a 1, nenhum vértice de X ou de Y pode aparecer em duas arestas escolhidas.

Logo, M é de fato um emparelhamento.

Luís Felipe

28/05/26

Do emparelhamento para o fluxo

Reciprocamente, dado um emparelhamento M em G, construímos um fluxo em D da seguinte forma:

Para cada aresta $xy \in M$, enviamos uma unidade pelo caminho:

$$s, x, y, t.$$

Isto define:

$$f(s,x) = 1, \qquad f(x,y) = 1, \qquad f(y,t) = 1.$$

Para as demais arestas, o fluxo é zero.

Assim, um emparelhamento de tamanho k gera um fluxo de valor k.

Portanto:
tamanho do emparelhamento máximo em G =
valor do fluxo máximo em D.

Luís Felipe

28/05/26

Algoritmo

```
1  funcao emparelhamento_bipartido_por_fluxo(G, X, Y):
2
3      construa uma rede D
4
5      adicione origem s e destino t
6
7      para cada x em X faca
8          adicione aresta sx com capacidade 1
9
10     para cada xy em E(G), com x em X e y em Y faca
11         adicione aresta xy com capacidade 1
12
13     para cada y em Y faca
14         adicione aresta yt com capacidade 1
15
16     f <- fluxo_maximo(D, s, t)
17
18     M <- {xy em E(G): f(x,y) = 1}
19
20     retornar M
```

Luís Felipe

28/05/26

Python: construindo a rede

```python
def rede_para_emparelhamento_bipartido(G, X, Y):
    # G: lista de adjacencia do grafo bipartido
    # X, Y: biparticao

    s = "s"
    t = "t"

    C = {}

    for x in X:
        C[(s, x)] = 1

    for x in X:
        for y in G[x]:
            if y in Y:
                C[(x, y)] = 1

    for y in Y:
        C[(y, t)] = 1

    return C, s, t
```

Luís Felipe

28/05/26

Python: recuperando o emparelhamento

```python
def recupera_emparelhamento(fluxo, X, Y):
    M = set()

    for x in X:
        for y in Y:
            if fluxo.get((x, y), 0) == 1:
                M.add((x, y))

    return M
```

Depois de rodar fluxo máximo na rede construída, basta selecionar as arestas xy com fluxo igual a 1.