Fluxo em Grafos

SCC5900

Projeto de Algoritmos

Fluxo Máximo

- Podemos interpretar um grafo orientado como um fluxo em rede:
  - Existe uma origem que produz um material em uma taxa fixa
  - E um depósito que consome esse material na mesma taxa
- O fluxo em qualquer ponto do grafo é a taxa na qual o material se move
- Existem diversas aplicações como líquidos em tubos, peças em linhas de montagem e correntes elétricas

Fluxo Máximo

- Cada aresta orientada pode ser entendida como um canal:
  - E cada canal possui uma capacidade estabelecida
- Vértices são junções de canais, além da origem e do depósito:
  - Não há acumulação de material nos vértices
  - Ou seja, a taxa de entrada deve ser igual à taxa de saída de material
  - Chamamos essa propriedade de *conservação de fluxo*

Fluxo Máximo

- Em fluxo máximo, deseja-se calcular a maior taxa de fluxo de material:
  - Da origem até o depósito
  - Sem violar quaisquer restrições de capacidade
  - Garantindo a propriedade de conservação do fluxo.

Definições

- Um **fluxo em rede** é um grafo orientado $G = (V, A)$:
  - Cada aresta $(u, v) \in A$ tem uma capacidade não negativa $c\,(u, v) \geq 0$
  - Se c(u,v) = 0, aresta(u, v) não $\in A$
  - Dois vértices são especiais: origem s e depósito t
  - Por conveniência, assume-se que existe um caminho s □ v □ t

Definições

- Um **fluxo** em G é uma função de valor real $f : A$ □ $\Re$ que satisfaz:
  - Restrição de Capacidade: para todo $u, v \in A$, $0 \leq f(u, v) \leq c(u, v)$
  - Conservação de fluxo: para todo $u \in V - \{s, t\}$

$$\sum_{u \text{ entrando } V} f(u) = \sum_{u \text{ saindo } V} f(u)$$

Definições

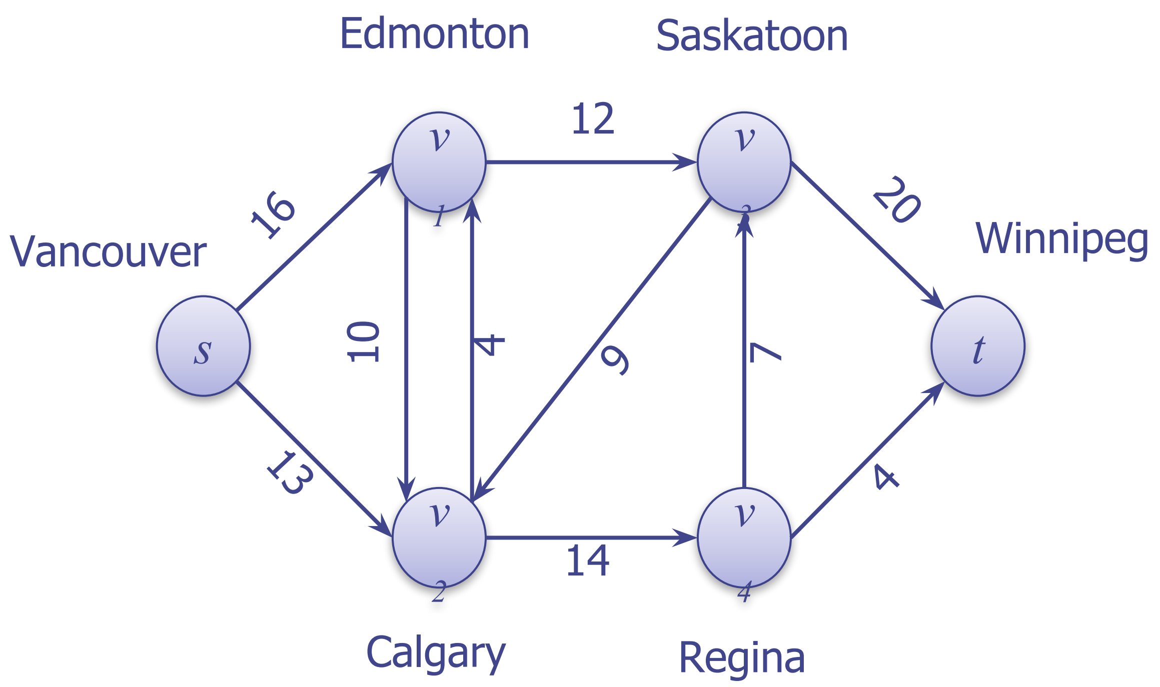
- O **valor de um fluxo** em G é definido como:

$$|f| = \sum_{v \in V} f(s, v)$$

ou seja, o fluxo que sai da origem

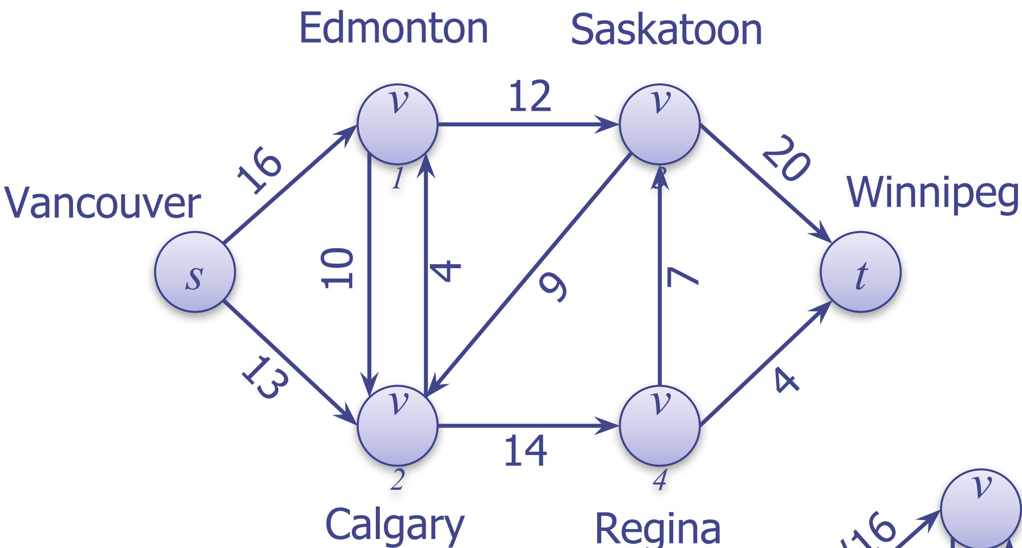
- No problema de fluxo máximo, queremos encontrar um fluxo de valor máximo de *s* a *t* sobre *G*

Exemplo

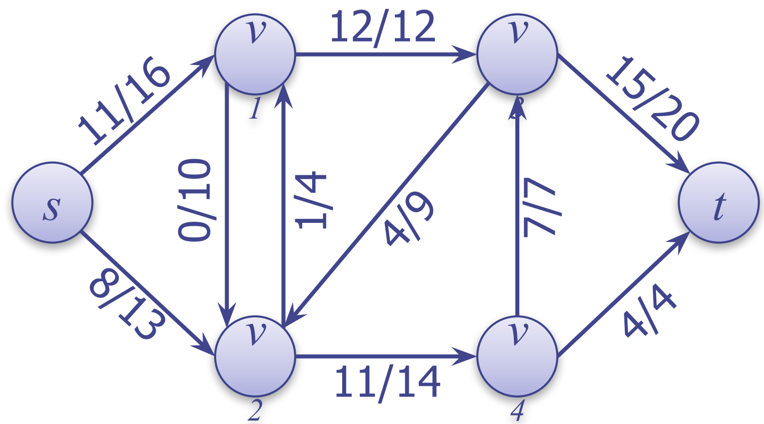


Edmonton
Saskatoon
Vancouver
Winnipeg
Calgary
Regina
12
16
20
10
4
9
7
13
14
4

Exemplo



Edmonton
Saskatoon
Vancouver
Winnipeg
Calgary
Regina
s
t
v
1
v
v
2
v
4
16
12
20
10
4
9
7
13
14
4



s
t
v
1
v
v
2
v
4
11/16
12/12
15/20
0/10
1/4
4/9
7/7
8/13
11/14
4/4

Método de Ford-Fulkerson

- Engloba diversas implementações com diferentes complexidades
- Utiliza três ideias importantes:
  - Redes residuais
  - Caminhos em ampliação, e
  - Cortes

Redes residuais

- A rede residual consiste em arestas que podem admitir mais fluxo
- A capacidade residual de dois vértices *u*, *v* é dada por:
  - $c_f(u, v) = c\,(u, v) - f\,(u, v)$

Redes residuais

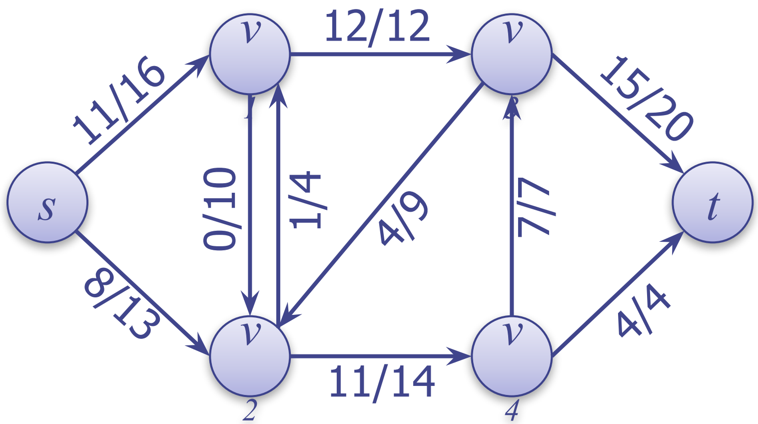
- Por exemplo, se
  - $c\,(u, v) = 16$ e $f\,(u, v) = 11$, então
  - $c_f(u, v) = 5$
  - Portanto pode aumentar a capacidade em 5 unidades antes de exceder a restrição
- Mas, se:
  - $c\,(u, v) = 16$ e $f\,(u, v) = -4$, então
  - $c_f(u, v) = 20$
  - Pois, pode-se cancelar o fluxo contrário em 4 unidades e empurrar mais 16 unidades

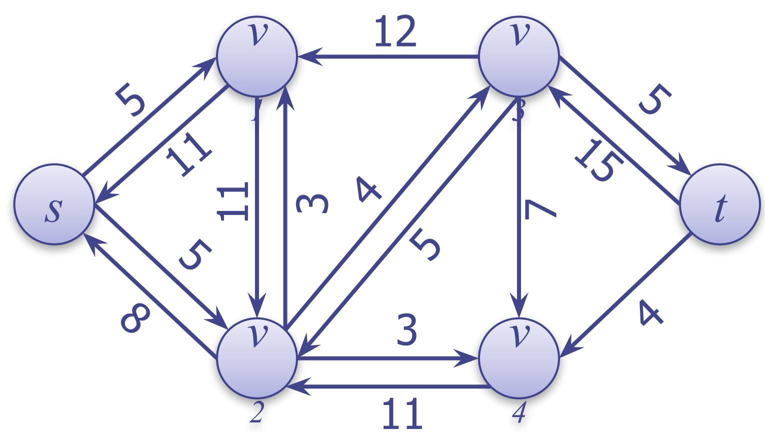
Redes residuais

- Dados um fluxo em rede $G = (V, A)$ e um fluxo f, a rede residual de G induzida por f é $G_f=(V, A_f)$, onde:

$$A_f = \{(u, v) \in V \times V : c_f(u, v) > 0\}$$

Redes residuais





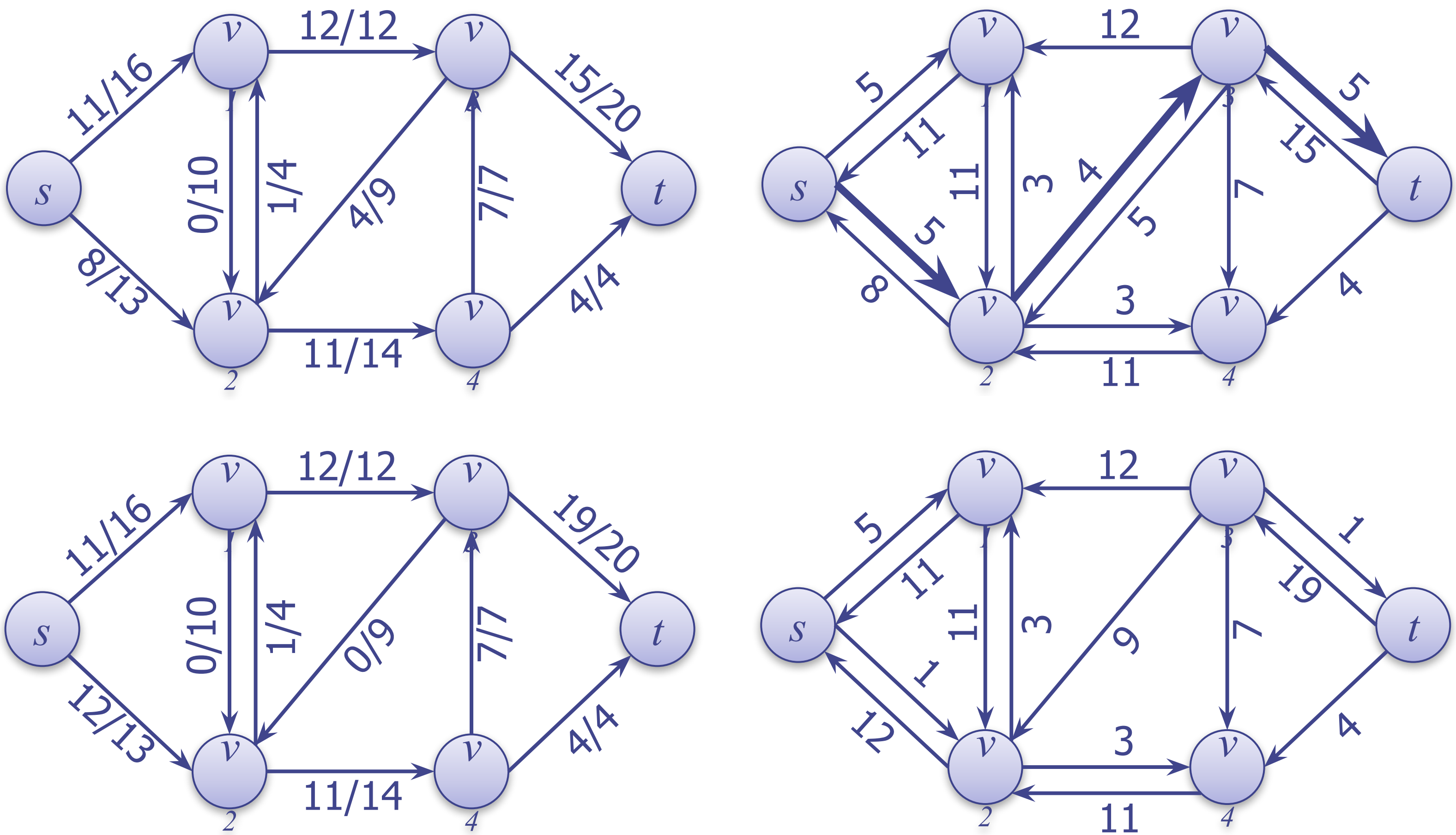
Redes residuais

- As arestas em A_f são as arestas em A ou suas inversas:
  - Se $f(u, v) < c(u, v)$ então $c_f(u, v) = c(u, v) - f(u, v) > 0$
  - Se $f(v, u) < 0$, então $c_f(v, u) = c(v, u) - f(v, u) < 0$
  - Em ambos os casos (u, v) e $(v, u) \in A_f$

Caminhos em ampliação

- Dados um fluxo em rede $G = (V, A)$, e um fluxo f, um caminho em ampliação p é um caminho simples de s a t na rede residual G_f
- Um caminho de ampliação admite algum fluxo positivo adicional de s a t sem violar as capacidades das arestas

Caminhos em ampliação



Caminhos em ampliação

- **Capacidade residual** é a quantidade máxima que se pode aumentar o fluxo em cada aresta de um caminho de ampliação *p*:

$$c_f(p) = min\{c_f(u, v): (u, v) \text{ está em } p\}$$

Ford-Fulkerson

- O algoritmo básico de Ford-Fulkerson consiste em encontrar caminhos de ampliação, e iterativamente aumentar o fluxo em *G*

Ford-Fulkerson

```
Ford-Fulkerson(G, s, t)
   for cada aresta (u, v) ∈ A
      do f[u,v] = 0
         f[v,u] = 0
   while existe p de s até t na rede residual G_f
      do c_f(p) = min{c_f(u,v): (u,v) está em p}
         for cada aresta (u,v) em p
            do f[u,v] = f[u,v] + c_f(p)
               f[v,u] = -f[u,v]
```

Edmonds-Karp

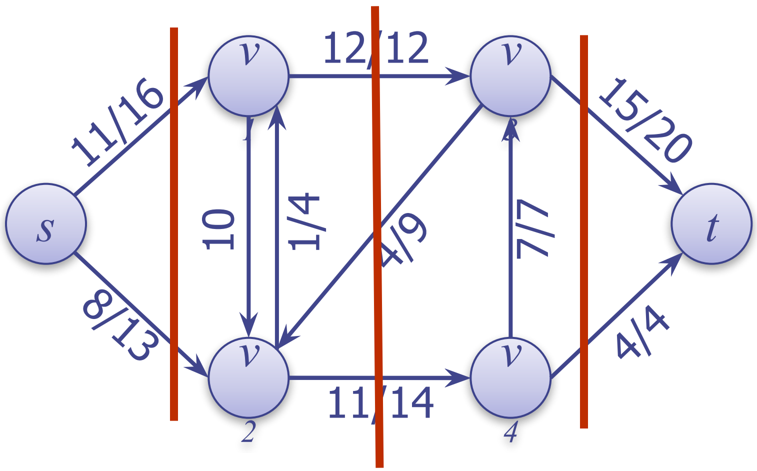
- Um ponto em aberto do algoritmo de Ford-Fulkerson é como encontrar os caminhos em ampliação *p*
- O algoritmo de Edmonds-Karp utiliza uma busca em largura para encontrar esses caminhos:
  - Portanto *p* é o caminho mínimo entre *s* e *t* considerando o número de arestas
  - O algoritmo de Edmonds-Karp possui complexidade $O(VA^2)$
- Existem outras variações mais eficientes...

Cortes

- Um *corte* em um grafo é a partição dos vértices em dois conjuntos disjuntos. Uma aresta de cruzamento conecta dois vértices em partições diferentes
- Um *corte-st* é um corte que coloca o vértice *s* em uma aresta e o vértice *t* em outra

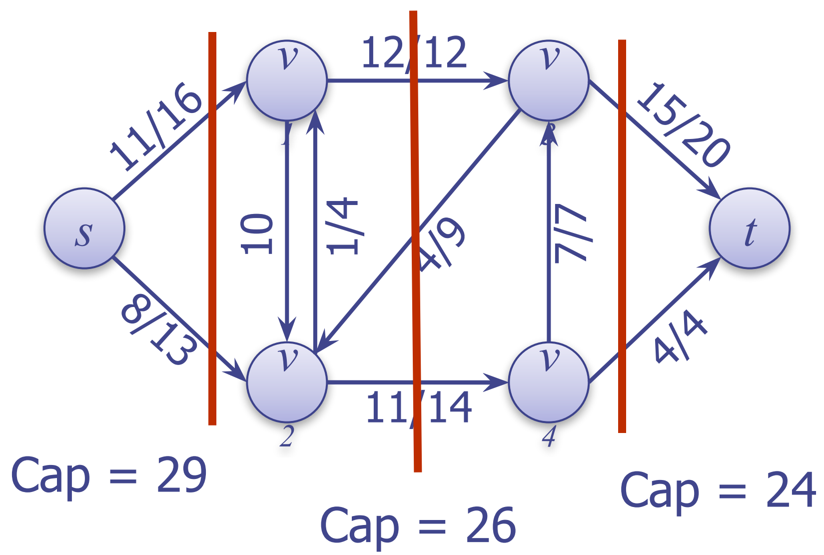
Cortes

- Um corte-*st* divide um fluxo em rede em dois componentes conexos, interrompendo o fluxo de *s* para *t*



Cortes

- A *capacidade* de um corte-*st* é a soma das capacidades das arestas de cruzamento *st* (*forward*)

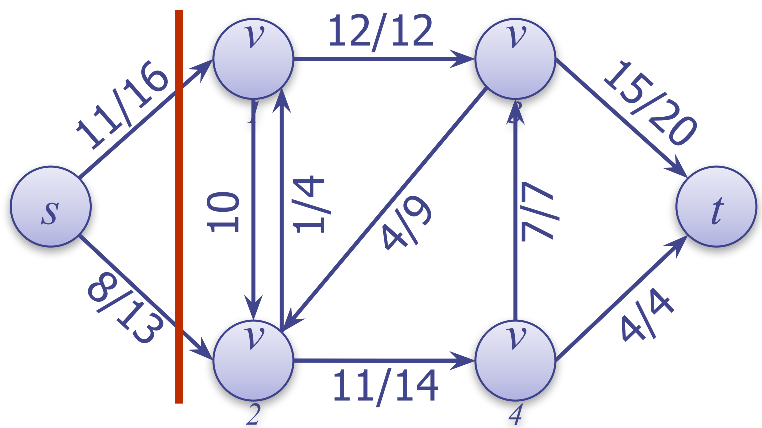


Cortes

- O *fluxo* através de um corte-*st* é a diferença entre a soma do fluxo das arestas de cruzamento *st* menos o fluxo das arestas de cruzamento *ts* (*forward – backward*)

Cortes

- Para qualquer fluxo de rede, o fluxo através de qualquer corte-*st* é igual ao valor do fluxo*

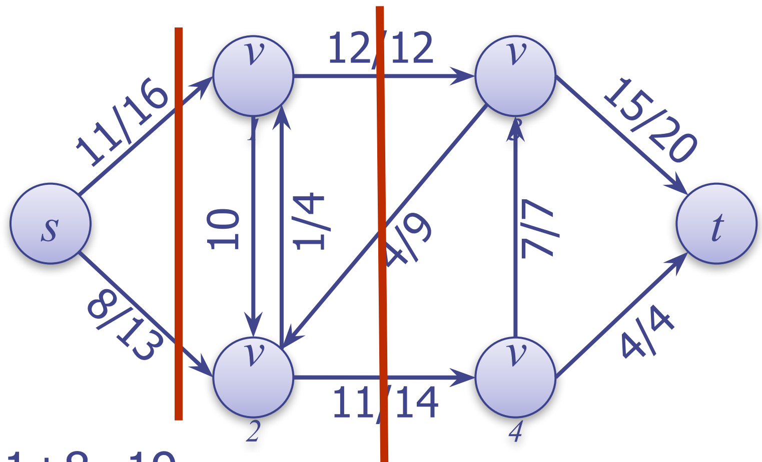


11+8=19

*Observe que este fluxo em rede não é máximo

Cortes

- Para qualquer fluxo de rede, o fluxo através de qualquer corte-*st* é igual ao valor do fluxo*



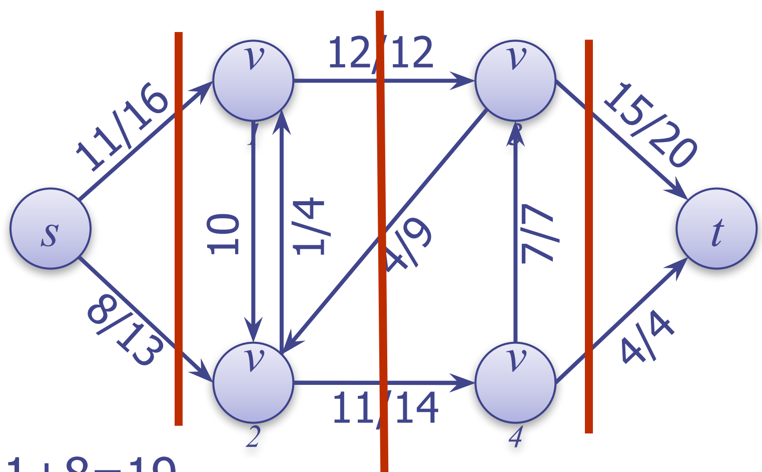
11+8=19

12-4+11=19

*Observe que este fluxo em rede não é máximo

Cortes

- Para qualquer fluxo de rede, o fluxo através de qualquer corte-*st* é igual ao valor do fluxo*



11+8=19

12-4+11=19

15+4=19

*Observe que este fluxo em rede não é máximo

Corte mínimo

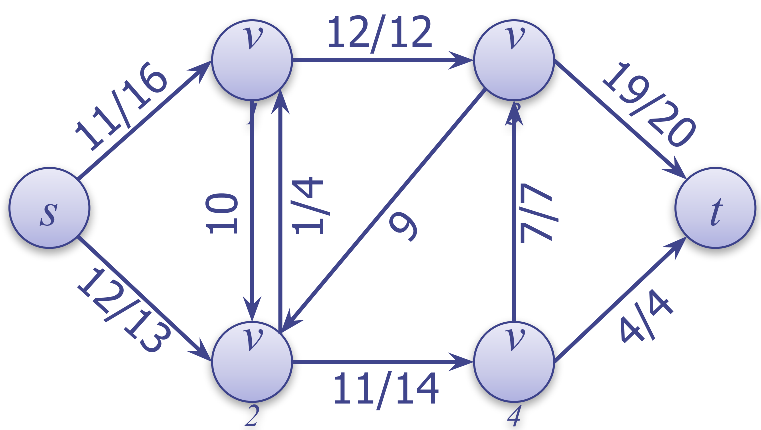
- No problema de corte mínimo, deve-se encontrar um corte-*st* no qual a capacidade do corte é menor do que a capacidade de qualquer outro corte-*st*
- Na verdade, resolver esse problema é igual a resolver o problema de fluxo máximo

Corte mínimo

- O fluxo máximo em um fluxo em rede é igual a capacidade mínima de um corte-*st*
- Porque? É o corte mínimo que limita o fluxo na rede, e consequentemente define o fluxo máximo

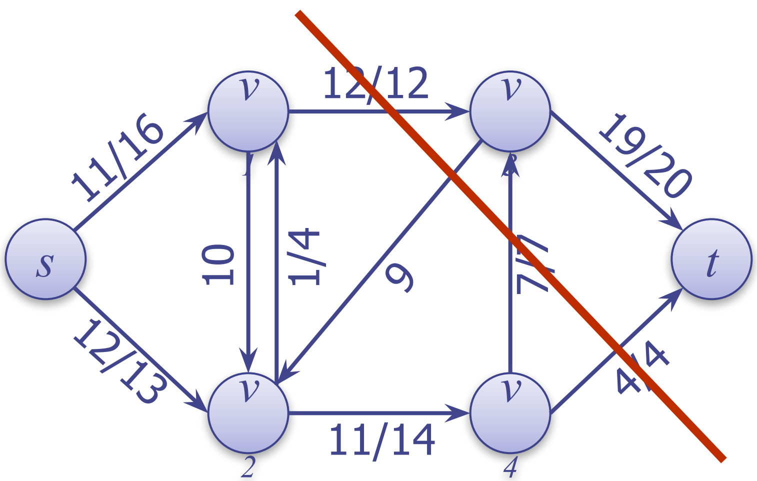
Corte mínimo

- Observe a seguinte rede com fluxo máximo igual a 23. Existe um conjunto de arestas *forward* cheias que limitam o fluxo



Corte mínimo

- Observe a seguinte rede com fluxo máximo igual a 23. Existe um conjunto de arestas *forward* cheias que limitam o fluxo

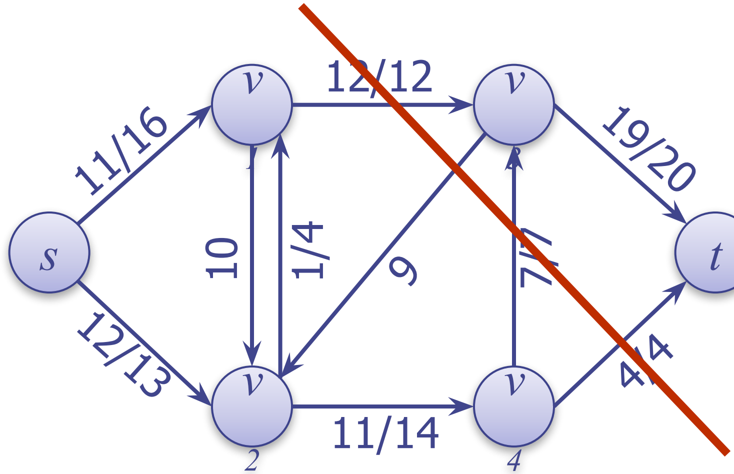


Corte mínimo

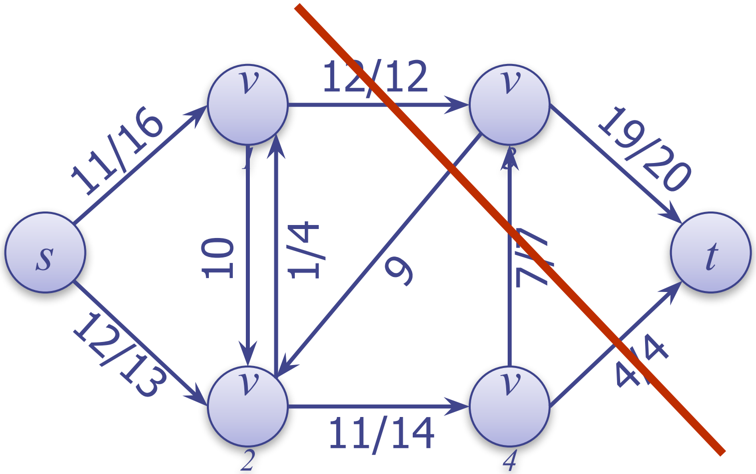
- Como achar as arestas (ou vértices) que fazem parte de um corte mínimo?
- Segundo Sedgewick:

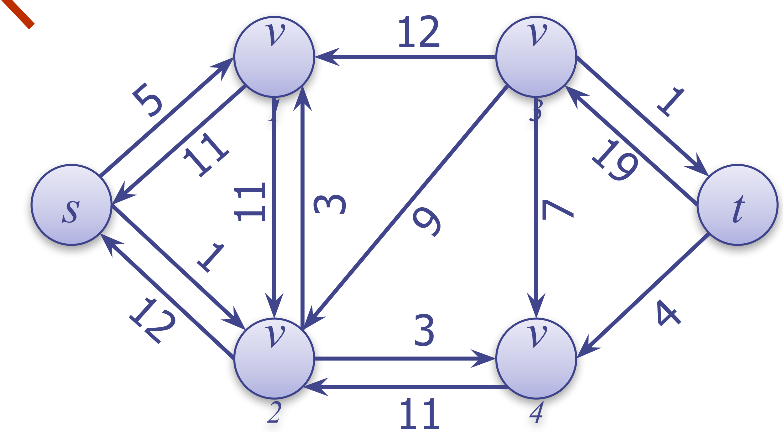
"The Ford-Fulkerson algorithms gives precisely such a flow and cut: When the algorithm terminates, identify the first full forward or empty backward edge on every path from *s* to *t* in the graph. Let C_s be the set of all vertices that can be reached from *s* with an undirected path that does not contain a full forward or empty backward edge and let C_t be the remaining vertices. Then, *t* must be in C_t, so (C_s, C_t) is na *st*-cut, whose cut consists entirely of full forward or empty backward edges."

Corte mínimo



Corte mínimo





Corte Mínimo

- Dessa forma, podemos utilizar uma busca em largura sobre a rede residual a partir de *s*
- Os vértices *u*-*v* que pertencem ao corte mínimo são aqueles que *u* foi marcado na busca em largura e *v* não foi marcado
- Veja que não é preciso fazer uma chamada adicional a busca em largura. Pode-se utilizar o resultado da última busca realizada pelo algoritmo de Edmonds-Karp