# Fluxo em redes

## Algoritmos em Grafos

Marco A L Barbosa

# Conteúdo


Introdução

Problema do fluxo máximo

- O método de Ford-Fulkerson
- Algoritmo básico de Ford-Fulkerson
- Algoritmo de Edmonds-Karp

Referências

O estudo utilizando apenas este material **não é suficiente** para o entendimento do conteúdo. Recomendamos a leitura das referências no final deste material e a resolução (por parte do aluno) de todos os exercícios indicados.

# Introdução

# Introdução

- Uma **rede** $G = (V, E)$ é um grafo direcionado no qual cada aresta $(u, v) \in E$ tem uma capacidade $c(u, v) \geq 0$
- Se $G$ contém a aresta $(u, v)$ ele não pode conter $(v, u)$
- Se $(u, v) \notin E$, então $c(u, v) = 0$
- Destacamos dois vértice $s$ (**fonte**) e $t$ (**sumidouro**)
- Para cada vértice $v \in V$, temos $s \rightsquigarrow v \rightsquigarrow t$

# Introdução

- Um **fluxo** em $G$ é uma função $f : V \times V \to \mathbb{R}$ que satisfaz as seguintes propriedades
  - **Restrição de capacidade**: Para todo $u, v \in V$, $0 \leq f(u, v) \leq c(u, v)$
  - **Conservação do fluxo**: Para todo $u \in V - \{s, t\}$

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

A quantidade $f(u, v)$ é chamada de fluxo entre $u$ e $v$. Quando $(u, v) \notin E$, não pode haver fluxo de $u$ para $v$ e portanto $f(u, v) = 0$

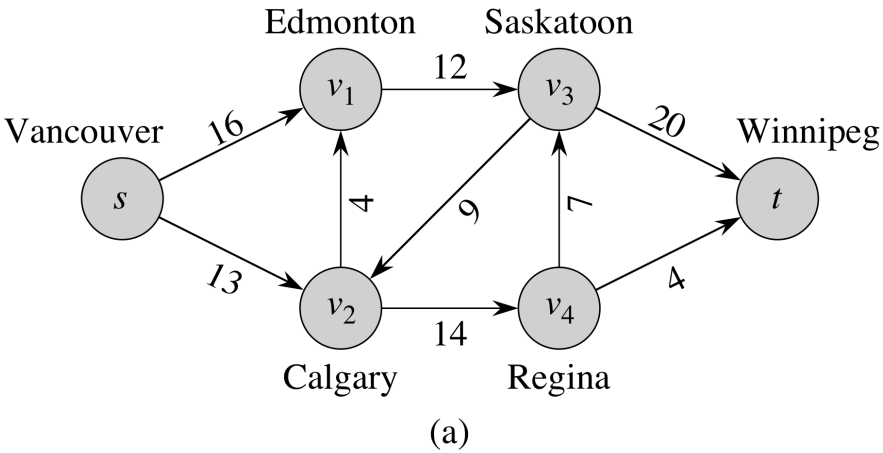
# Introdução

- O **valor** $|f|$ do fluxo $f$ é definido como

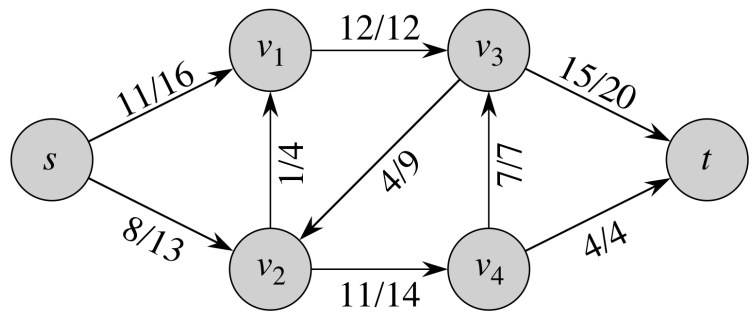
$$|f| = \sum_{v \in V} f(s, v) - \sum_{v \in V} f(v, s)$$

ou seja, o fluxo total que saí de $s$ menos o fluxo total que entra em $s$

# Exemplo


Edmonton
Saskatoon
Vancouver
Winnipeg
Calgary
Regina
$v_1$
$v_2$
$v_3$
$v_4$
$s$
$t$
16
13
12
14
4
9
7
20
4

(a)


$v_1$
$v_2$
$v_3$
$v_4$
$s$
$t$
11/16
8/13
12/12
11/14
1/4
4/9
7/7
15/20
4/4

(b)

# Problema do fluxo máximo

# Problema do fluxo máximo

Dado uma rede $G$, uma fonte $s$ e um sumidouro $t$, o **problema do fluxo máximo** consiste em encontrar um fluxo em $G$ de valor máximo.

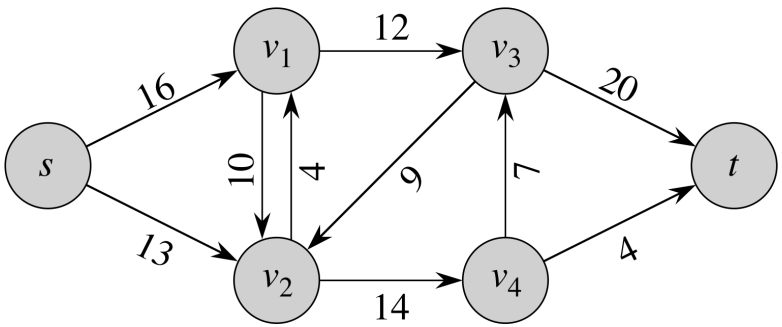
# Criando modelos

- Arestas antiparalelas
  - Suponha que a firma de caminhões oferecesse para Lucky Puck a oportunidade de transportar 10 engradados nos caminhões indo de Edmonton para Calgary
  - Parecesse uma boa oportunidade
  - O problema é que isto viola a restrição de que se $(v_1, v_2) \in E$, então $(v_2, v_1) \notin E$ (as arestas $(v_1, v_2)$ e $(v_2, v_1)$ são chamadas de **antiparalelas**)

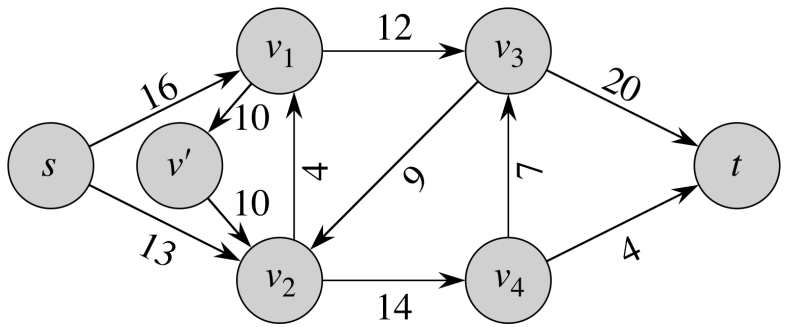
# Criando modelos

- Arestas antiparalelas
  - Suponha que a firma de caminhões oferecesse para Lucky Puck a oportunidade de transportar 10 engradados nos caminhões indo de Edmonton para Calgary
  - Parecesse uma boa oportunidade
  - O problema é que isto viola a restrição de que se $(v_1, v_2) \in E$, então $(v_2, v_1) \notin E$ (as arestas $(v_1, v_2)$ e $(v_2, v_1)$ são chamadas de **antiparalelas**)
  - Podemos transformar a rede em uma rede equivalente sem arestas antiparalelas
    - Escolhemos uma aresta, neste caso $(v_1, v_2)$, e a dividimos adicionando um $v'$ substituindo a aresta $(v_1, v_2)$ pelas arestas $(v_1, v')$ e $(v', v_2)$

# Criando modelos




(a)

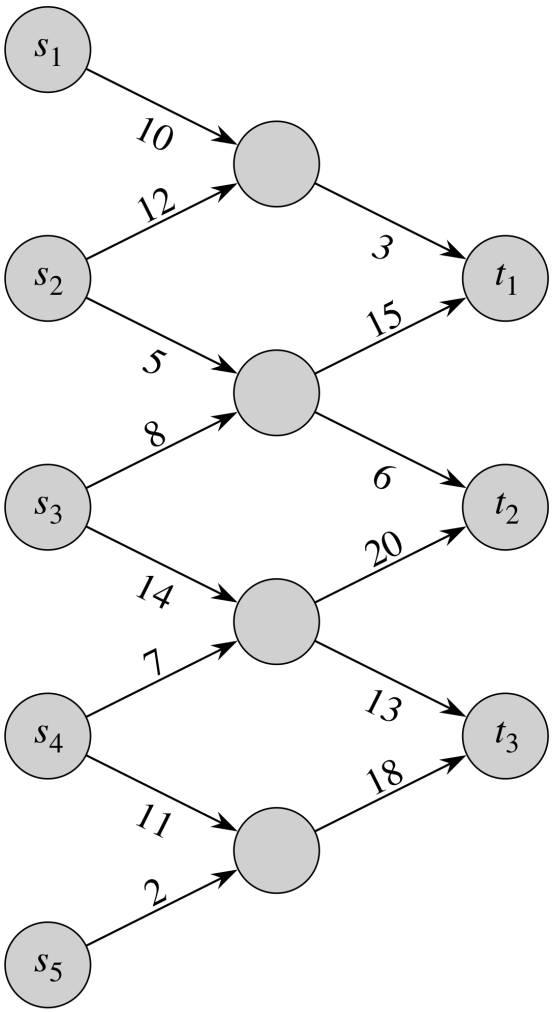



(b)

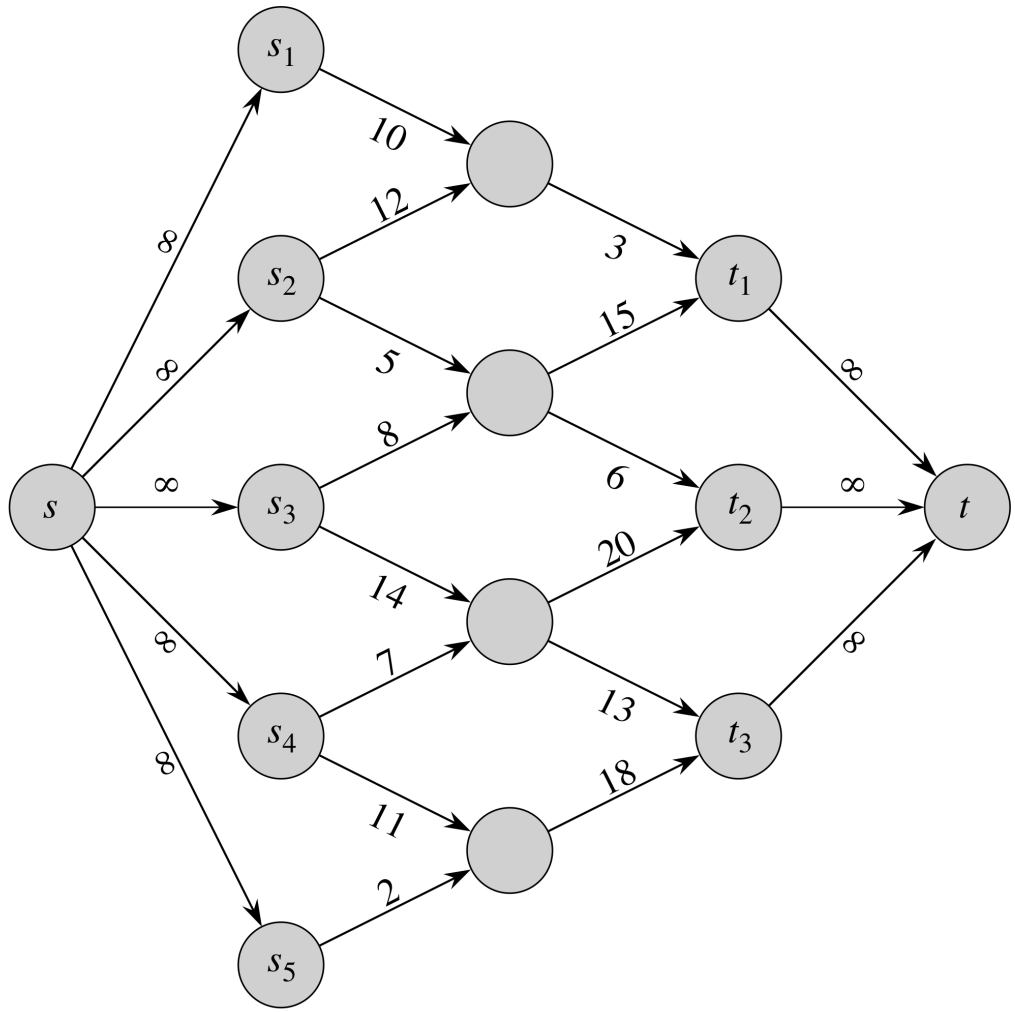
# Criando modelos

- Redes com múltiplas fontes e sumidouros
  - A empresa poderia ter mais que uma fábrica e mais que um depósito
  - Não está de acordo com a definição de rede
  - Transformamos em uma rede equivalente
    - Adicionamos uma super fonte $s$ e arestas com capacidade $\infty$ de $s$ para cada fonte original
    - Adicionamos um super sumidouro e arestas com capacidade $\infty$ de cada sumidouro original para o super sumidouro

# Criando modelos




(a)




(b)

# O método de Ford-Fulkerson

# O método de Ford-Fulkerson

- Chamamos de método e não algoritmo pois engloba diversas implementações com tempo de execução diferentes
- Utiliza os conceitos: rede residual, caminho aumentante e corte. Estes conceitos são importantes para muitos algoritmos e problemas de fluxo em rede
- Ideia
  - Incrementar iterativamente o valor do fluxo
  - Começamos com $f(u, v) = 0$ para todo $u, v \in G$, o que gera um fluxo de valor 0
  - A cada iteração, aumentamos o valor do fluxo encontrado um "caminho aumentante" na "rede residual" $G_f$ associada a $G$
  - O processo continua até que nenhum caminho aumentante é encontrado
  - O teorema do fluxo máximo e corte mínimo garante que este processo produz o fluxo máximo no término

# ford-fulkerson-method

```
ford-fulkerson-method(G, s, t)
1 Iniciar o fluxo f com 0
2 while existe um caminho aumentante p
            na rede residual G_f
3     aumente f ao longo de p
4 return f
```

- A fim de implementar e analisar o método de Ford-Fulkerson, precisamos de vários conceitos

# Redes residuais

- Intuitivamente, uma rede residual $G_f$ de uma rede $G$ e um fluxo $f$ consiste de arestas com capacidades que representam como o fluxo das arestas de $G$ podem ser alterados
  - O fluxo em uma aresta pode aumentar ou diminuir

# Redes residuais

- Intuitivamente, uma rede residual $G_f$ de uma rede $G$ e um fluxo $f$ consiste de arestas com capacidades que representam como o fluxo das arestas de $G$ podem ser alterados
  - O fluxo em uma aresta pode aumentar ou diminuir
- Seja $G = (V, E)$ uma rede com fonte $s$ e sumidouro $t$, $f$ um fluxo em $G$ e $u, v \in V$
- A **capacidade residual** $c_f(u, v)$ é definida como

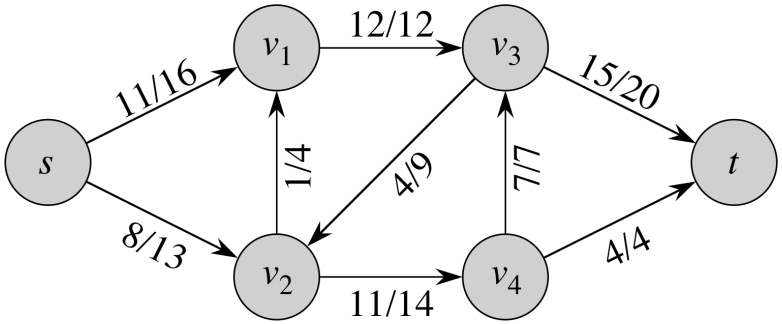
$$c_f(u,v) = \begin{cases} c(u,v) - f(u,v) & \text{se } (u,v) \in E \\ f(v,u) & \text{se } (v,u) \in E \\ 0 & \text{caso contrário} \end{cases}$$

- A **rede residual** de $G$ induzida por $f$ é $G_f = (V, E_f)$, onde

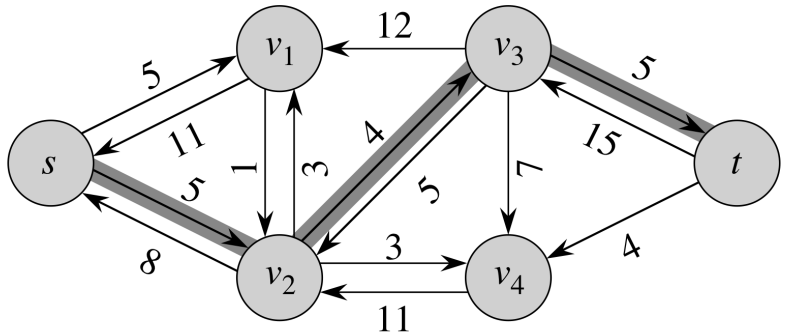
$$E = \{(u,v) \in V \times V : c_f(u,v) > 0\}$$

e $|E_f| \leq 2|E|$

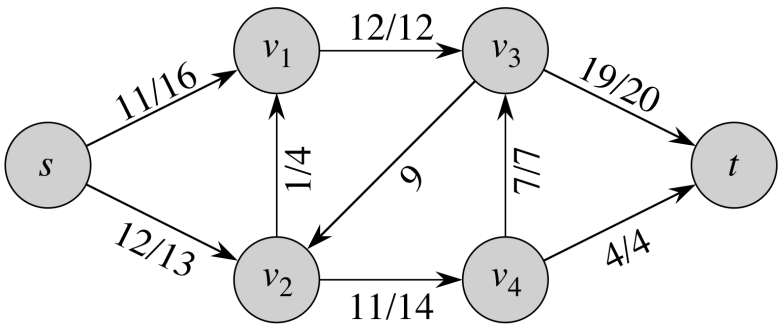
# Exemplo de rede residual



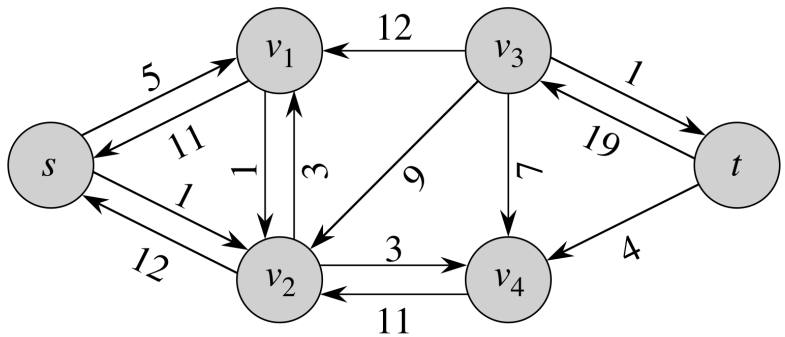

(a)




(b)




(c)




(d)

# Redes residuais

- Se $f$ é um fluxo em $G$ e $f'$ é um fluxo na rede residual $G_f$ correspondente, definimos $f \uparrow f'$, o **aumento** do fluxo $f$ por $f'$, como sendo a função $V \times V \to \mathbb{R}$

$$(f \uparrow f')(u,v) = \begin{cases} f(u,v) + f'(u,v) - f'(v,u) & \text{se } (u,v) \in E \\ 0 & \text{caso contrário} \end{cases}$$

# Redes residuais

- Lema 26.1
  - Seja $G = (V, E)$ um rede com fonte $s$ e sumidouro $t$ e seja $f$ um fluxo em $G$. Seja $G_f$ uma rede residual de $G$ induzida por $f$ e seja $f'$ um fluxo em $G_f$. Então a função $f \uparrow f'$ definida na equação (26.4) é um fluxo em $G$ com valor $|f \uparrow f'| = |f| + |f'|$
  - Prova vista em sala (veja o livro)

# Caminho aumentante

- Dado uma rede $G = (V, E)$ e um fluxo $f$, um **caminho aumentante** $p$ é um caminho simples de $s$ para $t$ na rede residual $G_f$
- O valor máximo que pode ser aumentado no fluxo de cada aresta no caminho aumentante $p$ é chamado **capacidade residual** de $p$, e é dado por

$$c_f(p) = \min\{c_f(u, v) : (u, v) \in p\}$$

# Caminho aumentante

- Lema 26.2
  - Seja $G = (V, E)$ um rede, $f$ um fluxo em $G$ e $p$ um caminho aumentante em $G_f$. Seja a função $f_p : V \times V \to \mathbb{R}$, definida como

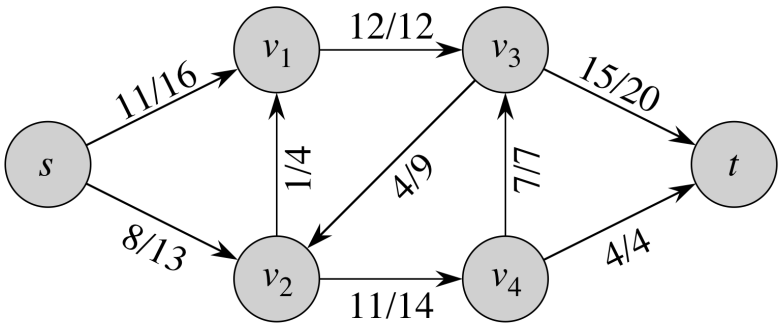
$$f_p(u, v) = \begin{cases} c_f(p) & \text{se } (u, v) \in p \\ 0 & \text{caso contrário} \end{cases}$$

  Então, $f_p$ é um fluxo em $G_f$ com valor $|f_p| = c_f(p) > 0$

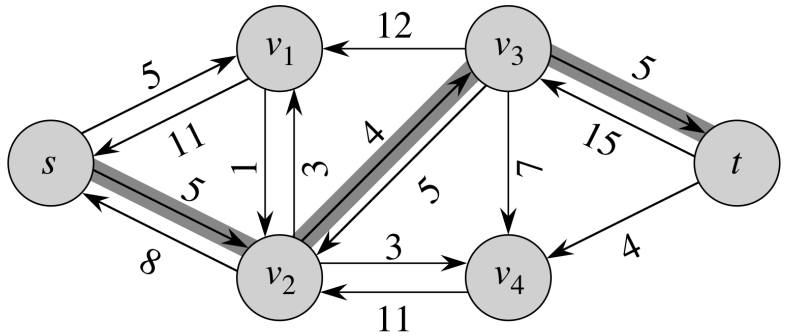
# Caminho aumentante

- Corolário 26.3
  - Seja $G = (V, E)$ um rede, $f$ um fluxo em $G$ e $p$ um caminho aumentante em $G_f$. Seja a função $f_p$ como definido na equação (26.8) e suponha que nós aumentamos $f$ por $f_p$. Então a função $f \uparrow f_p$ é um fluxo em $G$ com valor $|f \uparrow f_p| = |f| + |f_p| > |f|$
  - Prova: a partir dos lemas 26.1 e 26.2

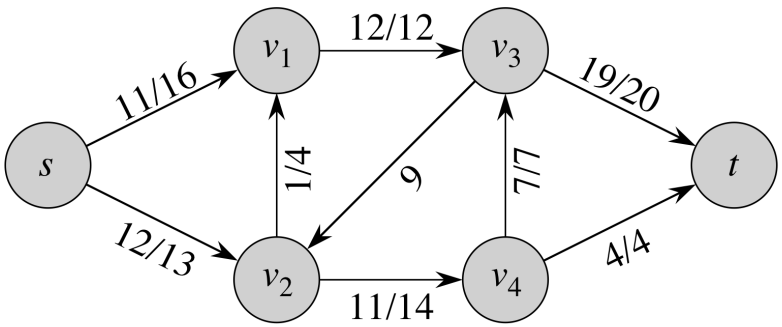
# Exemplo de caminho aumentante



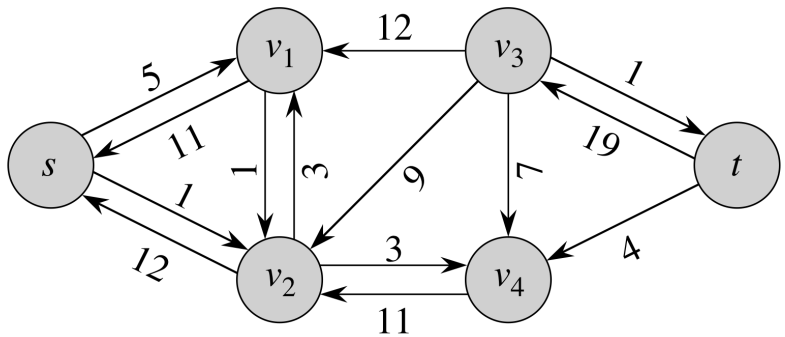

(a)




(b)




(c)




(d)

# Corte de rede

- Um corte $(S, T)$ de uma rede $G = (V, E)$ é uma partição de $V$ em $S$ e $T = V - S$, tal que $s \in S$ e $t \in T$
- Se $f$ é um fluxo, então o **fluxo líquido** $f(S, T)$ através do corte $(S, T)$ é definido como

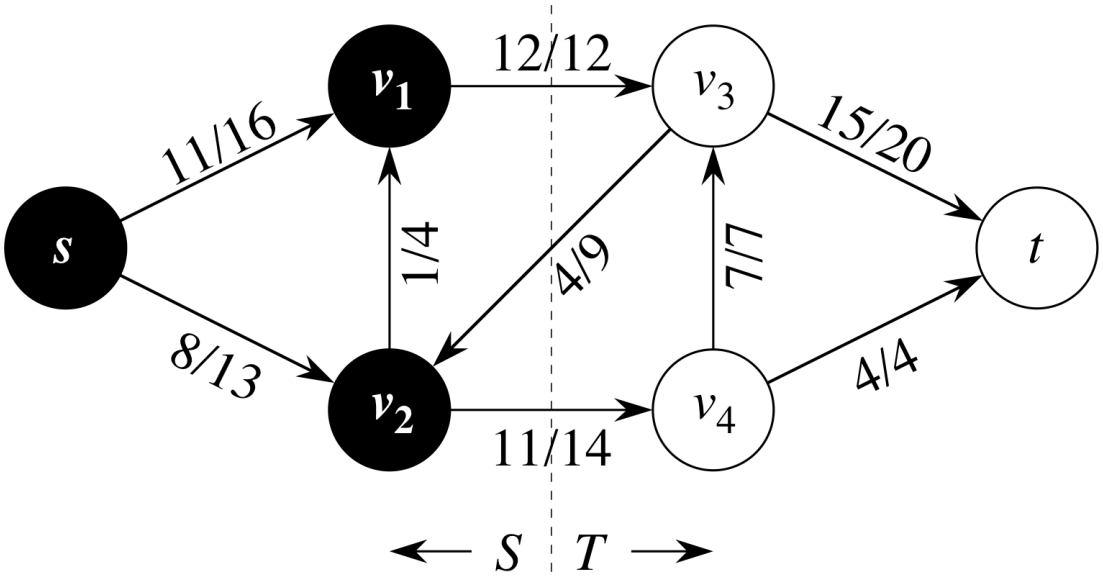
$$f(S, T) = \sum_{u \in S} \sum_{v \in T} f(u, v) - \sum_{u \in S} \sum_{v \in T} f(v, u)$$

- A **capacidade** do corte $(S, T)$ é

$$c(S, T) = \sum_{u \in S} \sum_{v \in T} c(u, v)$$

- Um **corte mínimo** de uma rede é um corte que tem capacidade mínima entre todos os cortes da rede

# Exemplo de corte



# Corte de rede

- Lema 26.4
  - Seja $f$ um fluxo em uma rede $G$ com fonte $s$ e sumidouro $t$, e seja $(S, T)$ qualquer corte de $G$. Então o fluxo líquido através do corte $(S, T)$ é $f(S, T) = |f|$
  - Prova vista em sala (veja o livro)

# Corte de rede

- Lema 26.4
  - Seja $f$ um fluxo em uma rede $G$ com fonte $s$ e sumidouro $t$, e seja $(S, T)$ qualquer corte de $G$. Então o fluxo líquido através do corte $(S, T)$ é $f(S, T) = |f|$
  - Prova vista em sala (veja o livro)
- Corolário 26.5
  - O valor do fluxo $f$ em uma rede $G$ é limitado superiormente pela capacidade de qualquer corte de $G$
  - Prova vista em sala (veja o livro)

# Teorema do fluxo máximo e corte mínimo

- Teorema 26.6
  - O valor do fluxo máximo é igual a capacidade de um corte mínimo.
    Seja $f$ um fluxo em uma rede $G = (V, E)$ com fonte $s$ e sumidouro $t$, então as seguintes condições são equivalentes:
    1. $f$ é um fluxo máximo em $G$
    2. A rede residual $G_f$ não contém nenhum caminho aumentante
    3. $|f| = c(S, T)$ para algum corte $(S, T)$ de $G$
  - Prova vista em sala (veja o livro)

# Algoritmo básico de Ford-Fulkerson

# Algoritmo básico de Ford-Fulkerson

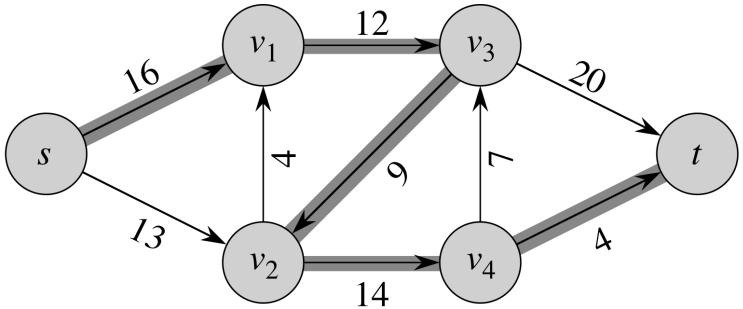
- Em cada iteração do método de Ford-Fulkerson é encontrado algum caminho aumentante $p$ que é utilizado para modificar o fluxo $f$
- Como o Lema 26.2 e o Corolário 26.3 sugerem, o fluxo $f$ pode ser substituído por $f \uparrow f_p$, gerando um novo fluxo com valor $|f| + |f_p|$
- Vamos ver uma implementação
  - Cada aresta residual em $p$ é uma aresta na rede original ou uma aresta contrária na rede original
  - Fluxo é adicionado se a aresta é a original
  - Fluxo é removido se a aresta é contrária
  - Quando não existe mais caminho aumentante, $f$ é máximo

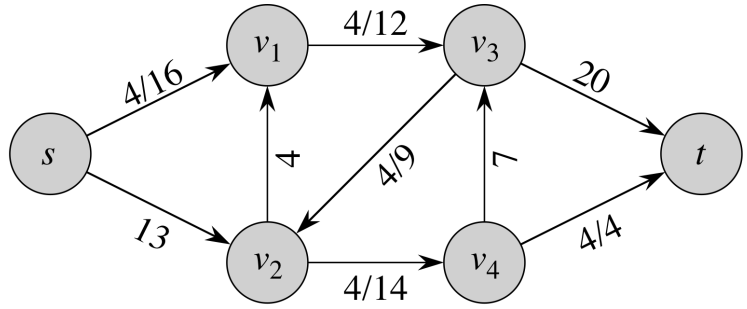
# Algoritmo básico de Ford-Fulkerson

```
ford-fulkerson(G, s, t)
1 for cada aresta (u, v) ∈ G.E
2     (u, v).f = 0
3 while existe um caminho p de s até t na
          rede residual G_f
4     c_f(p) = min{c_f(u, v) : (u, v) ∈ p}
5     for cada aresta (u, v) ∈ p
6         if (u, v) ∈ E
7             (u, v).f = (u, v).f + c_f(p)
8         else
9             (v, u).f = (v, u).f − c_f(p)
```

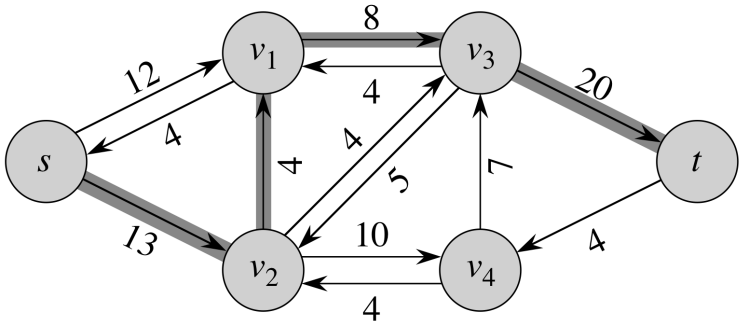
# Exemplo de execução

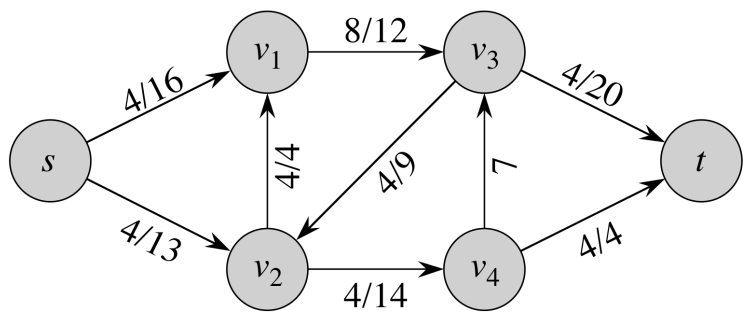
(a)



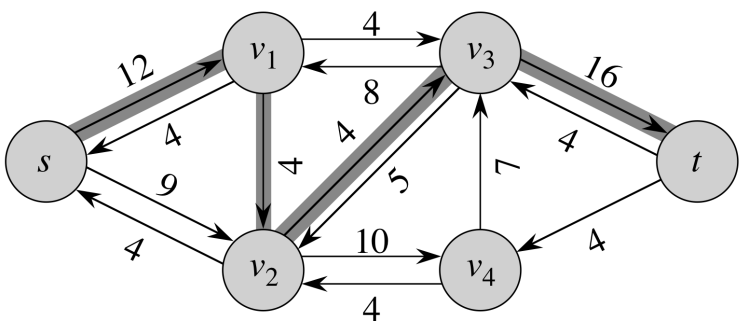



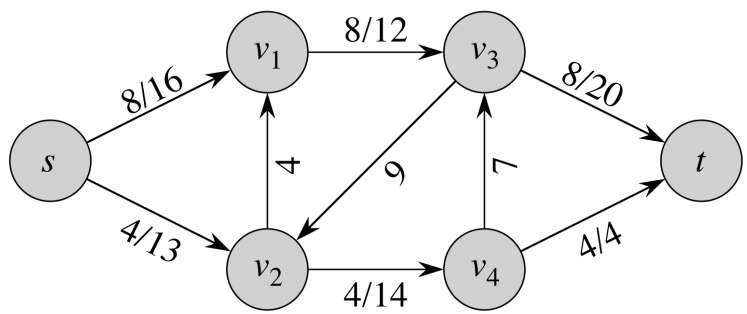

(b)



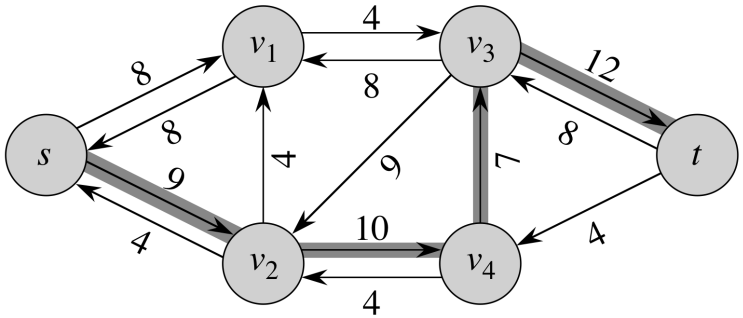

(c)

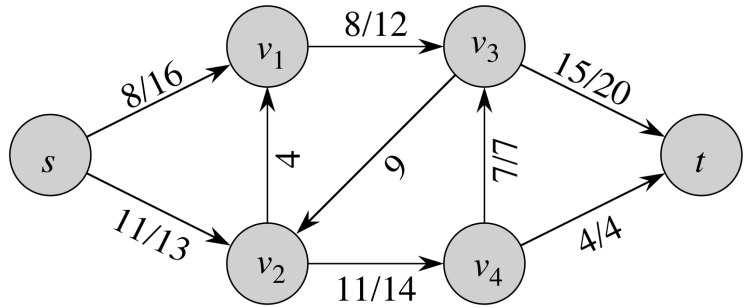


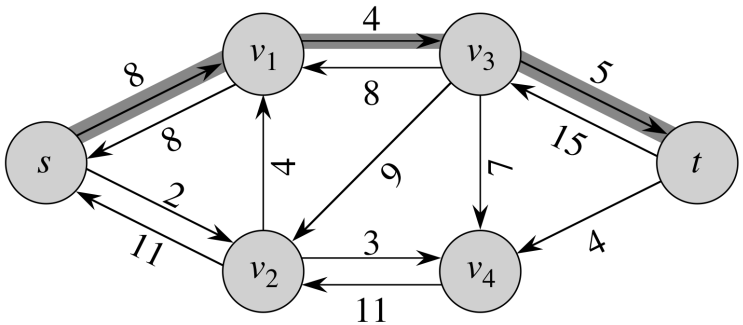
# Exemplo de execução

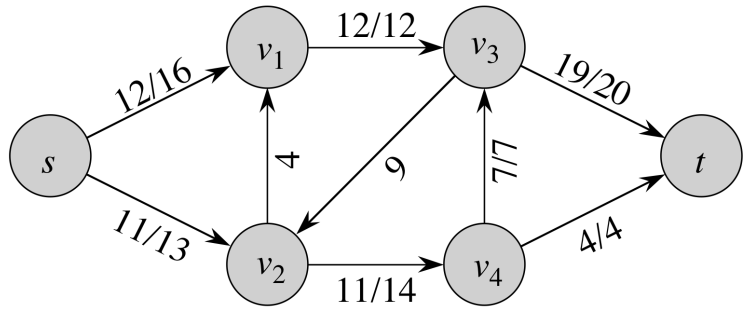
(d)



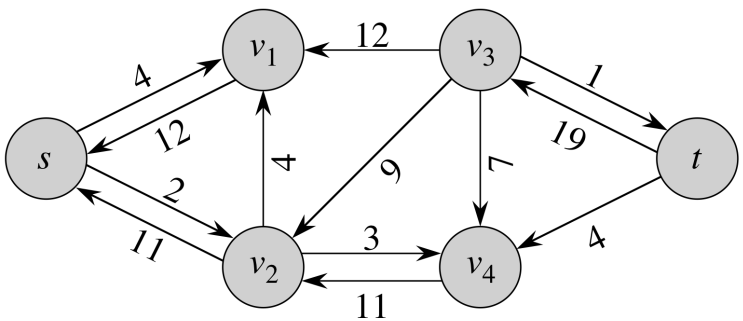




(e)



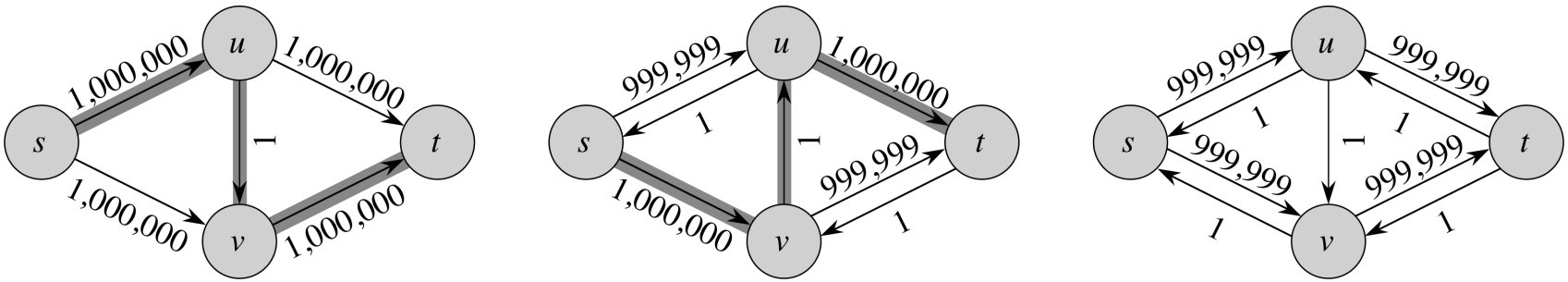

(f)



# Algoritmo básico de Ford-Fulkerson

- Análise do tempo de execução
  - Depende de como o caminho $p$ é escolhido
  - Vamos supor que todas as capacidades sejam inteiras
  - Seja $f^*$ o fluxo máximo na rede residual
  - Então, o laço `while` das linhas 3-8 executa no máximo $|f^*|$, isto porque o valor do fluxo aumenta em pelo menos uma unidade em cada iteração
  - O conteúdo dentro do `while` pode ser executado de forma eficiente se escolhermos a estrutura correta para representar a rede e se o caminho aumente for encontrado em tempo linear
    - Manter um grafo $G' = (V, E')$, onde $E' = \{(u, v) : (u, v) \in E \text{ ou } (v, u) \in E\}$
    - Encontrar o caminho aumente com dfs ou bfs, tempo $O(V + E') = O(E)$
    - Cada iteração demora $O(E)$
  - Portanto, o tempo de execução do algoritmo é $O(E|f^*|)$

# Exemplo ruim



# Algoritmo de Edmonds-Karp

# Algoritmo de Edmonds-Karp

- Encontrar o caminho aumentante $p$ com a busca em largura
- Escolher o menor caminho entre $s$ e $t$, sendo que o tamanho do caminho é o número de arestas no caminho
- Executa em $O(VE^2)$

# Referências

# Referências


- Thomas H. Cormen et al. Introduction to Algorithms. 3rd edition. Capítulo 26.