

Min-Cut/Max-Flow Algorithm

Paulo A. V. de Miranda

pavm@ic.unicamp.br

Laboratório de Informática Visual (LIV), Instituto de Computação (IC),
Universidade Estadual de Campinas (UNICAMP)

Introdução

- Imagine um material fluindo através de um sistema a partir de uma fonte, onde o material é produzido, até um destino, onde ele é consumido.
- O fluxo do material em qualquer ponto do sistema é dado pela taxa com que o material se move.
- Redes de fluxo podem ser usadas para modelar:
 - líquidos fluindo ao longo de tubulações.
 - peças através de linhas de montagem.
 - corrente através de redes elétricas.
 - informação através de redes de comunicação.

Introdução

Problema de fluxo máximo:

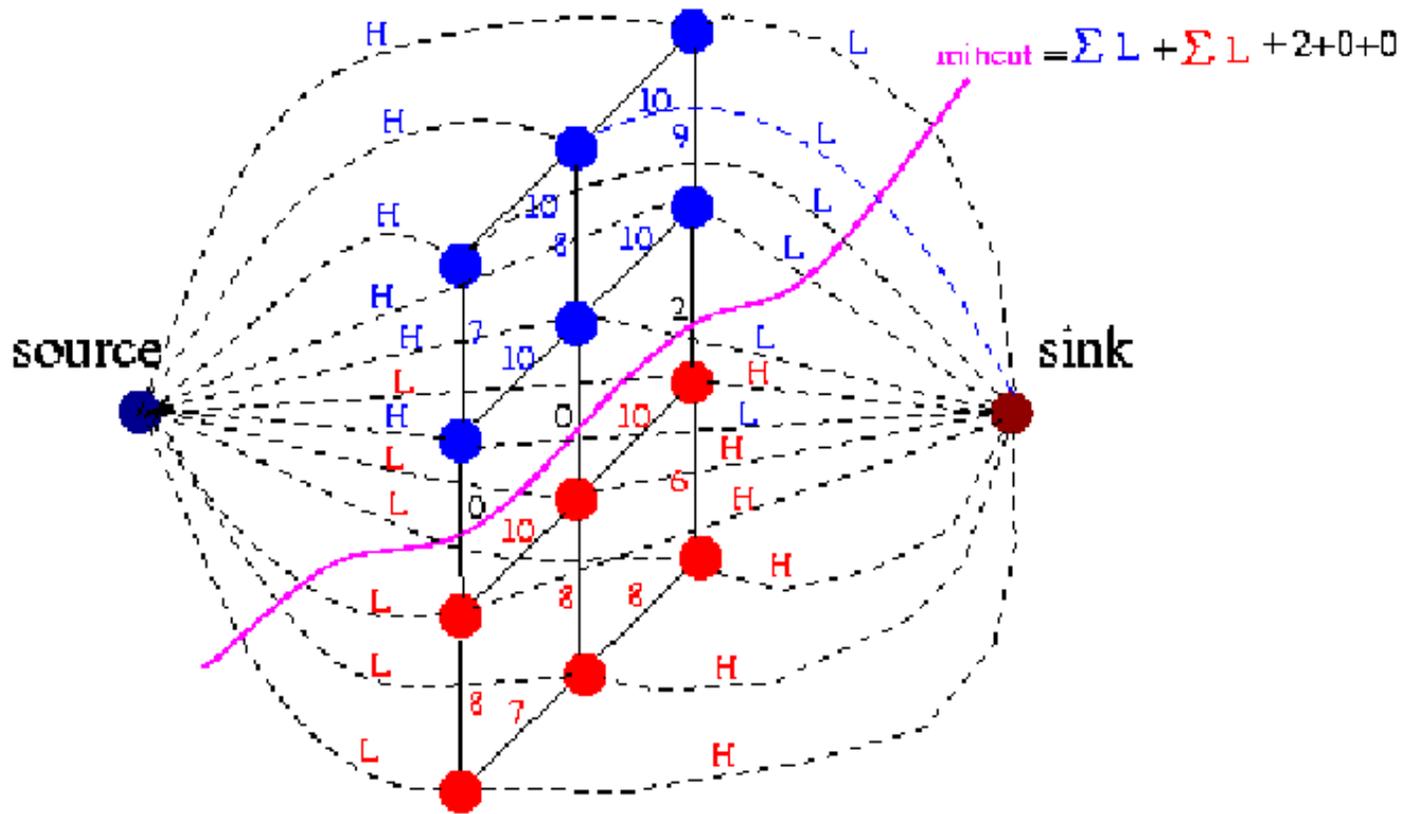
- Qual é a maior taxa de transmissão de material a partir da fonte até o destino sem violar as restrições de capacidade entre as várias partes da rede?

Aplicação em segmentação

- Explora o Teorema do Fluxo Máximo e Corte Mínimo.
 - Nós fonte e destino são adicionados ao grafo da imagem e cada pixel deve ser conectado a esses nós terminais por arcos.
 - O peso dos arcos entre pixels deve ser maior dentro e fora do objeto do que na fronteira do objeto.
 - Os pesos de arco com a fonte devem ser maiores no interior do objeto do que fora dele e o contrário em relação ao destino.

Aplicação em segmentação

$$E = \sum_{\forall(u,v) \in \mathcal{A} | u \in S, v \in T} w(u,v) + \sum_{\forall u \in \mathcal{I} | u \in S} w(u,t) + \sum_{\forall u \in \mathcal{I} | u \in T} w(s,u)$$



min-cut/max-flow solution

Definição

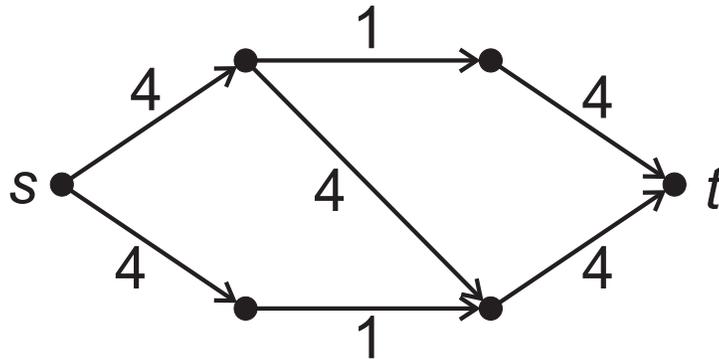
- Uma rede de fluxo $G = (V, E)$ é um grafo direcionado no qual cada aresta $(u, v) \in E$ possui uma capacidade não negativa $c(u, v) \geq 0$.

Definição

- Uma rede de fluxo $G = (V, E)$ é um grafo direcionado no qual cada aresta $(u, v) \in E$ possui uma capacidade não negativa $c(u, v) \geq 0$.
- O grafo possui dois vértices especiais: fonte (*source*) s e destino (*sink*) t .

Definição

- Uma rede de fluxo $G = (V, E)$ é um grafo direcionado no qual cada aresta $(u, v) \in E$ possui uma capacidade não negativa $c(u, v) \geq 0$.
- O grafo possui dois vértices especiais: fonte (*source*) s e destino (*sink*) t .
- Exemplo:



Fluxo no grafo

● O fluxo em G é uma função $f : V \times V \rightarrow R$ que satisfaz as seguintes três propriedades:

- **Restrição de capacidade:** Para todos $u, v \in V$, exigimos que $f(u, v) \leq c(u, v)$.
- **Anti-simetria:** Para todos $u, v \in V$, temos que $f(u, v) = -f(v, u)$.
- **Conservação de fluxo:** Para todo $u \in V - \{s, t\}$, temos que

$$\sum_{v \in V} f(u, v) = 0$$

Fluxo total no grafo

- O valor total de fluxo é definido como a soma do fluxo que sai da fonte s :

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

Fluxo total no grafo

- O valor total de fluxo é definido como a soma do fluxo que sai da fonte s :

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

- No problema de fluxo máximo, nos é dada uma rede de fluxo G com fonte s e destino t , e queremos encontrar um fluxo de valor máximo indo de s para t .

Visão geral da solução

Método de Ford-Fulkerson:

- Começamos com fluxo inicial zero (i.e., $f(u, v) = 0$ para todos $u, v \in V$).

Visão geral da solução

Método de Ford-Fulkerson:

- Começamos com fluxo inicial zero (i.e., $f(u, v) = 0$ para todos $u, v \in V$).
- A cada iteração, aumentamos o fluxo total encontrando algum caminho (a partir da fonte s até o destino t) ao longo do qual podemos empurrar mais fluxo (*augmenting paths*).

Visão geral da solução

Método de Ford-Fulkerson:

- Começamos com fluxo inicial zero (i.e., $f(u, v) = 0$ para todos $u, v \in V$).
- A cada iteração, aumentamos o fluxo total encontrando algum caminho (a partir da fonte s até o destino t) ao longo do qual podemos empurrar mais fluxo (*augmenting paths*).
- Repetimos este processo até que nenhum caminho de aumento pode ser encontrado.

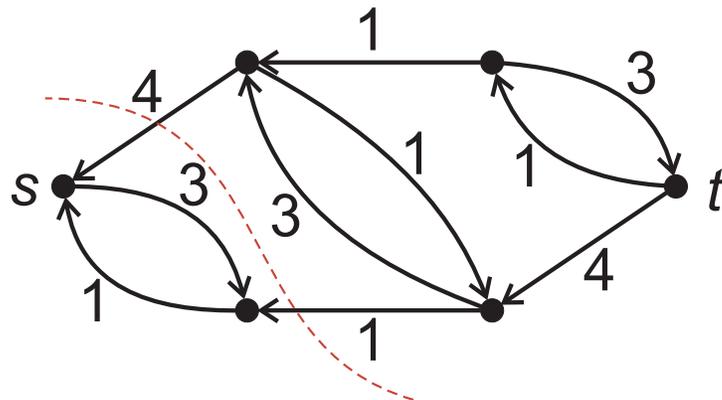
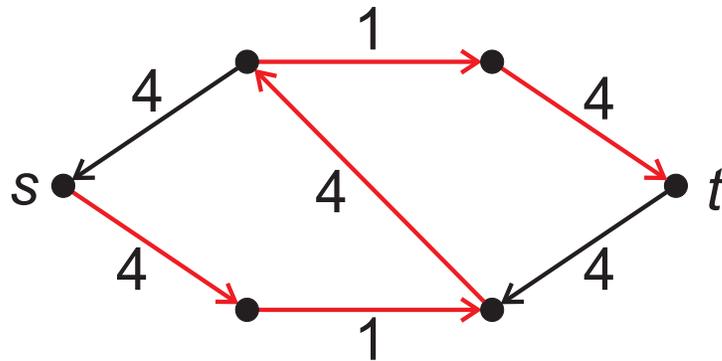
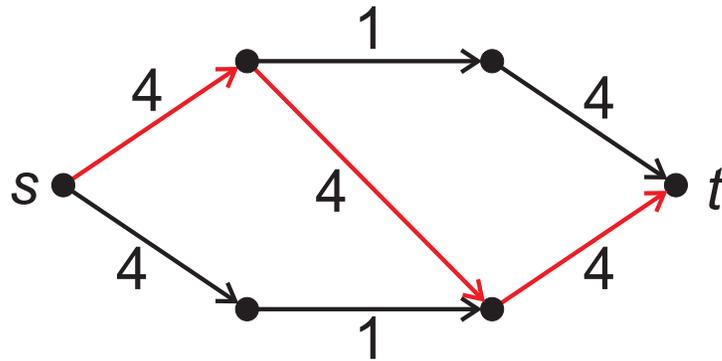
Redes Residuais

- Dados uma rede de fluxo G e um fluxo f , a rede residual consiste de arestas que podem admitir mais fluxo adicional.
- Para todos $u, v \in V$, a quantidade de fluxo adicional que podemos empurrar de u para v antes de exceder a capacidade $c(u, v)$ é a capacidade residual de (u, v) , dada por:

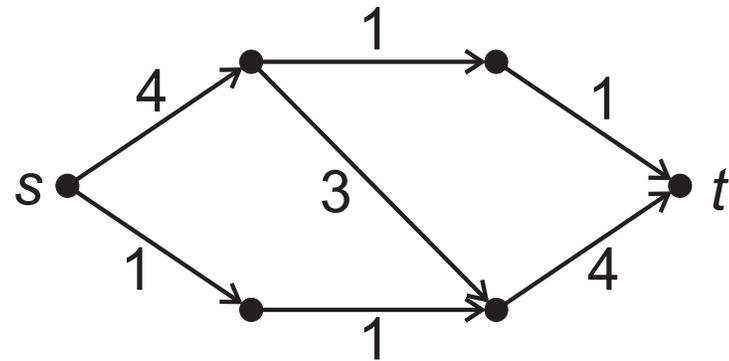
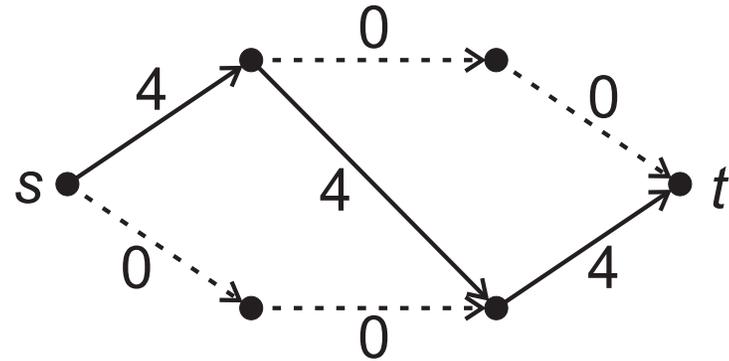
$$c_f(u, v) = c(u, v) - f(u, v)$$

Exemplo: *augmenting paths*

capacidade residual



fluxo



Alg. básico de Min-Cut/Max-Flow

Algorithm 1 — FORD-FULKERSON ALGORITHM

INPUT: A flow network $G = (V, E)$ with nodes s and t .

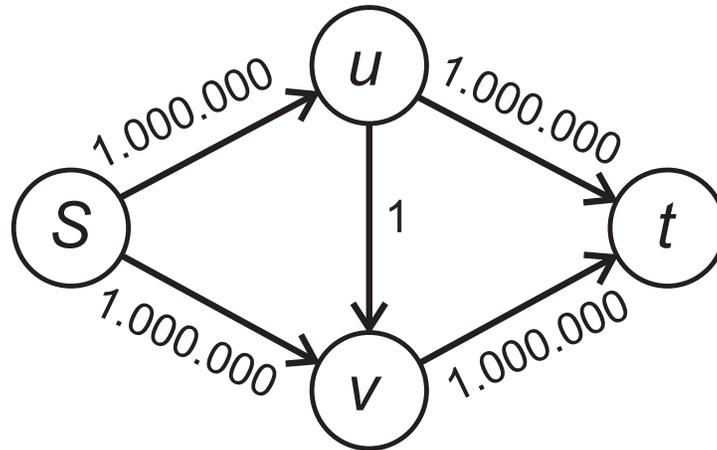
OUTPUT: The maximum flow f in G .

AUXILIARY: The residual network G_f .

1. **For each edge** $(u, v) \in E[G]$, **do**
2. $f[u, v] \leftarrow 0$ **and** $f[v, u] \leftarrow 0$.
3. **While there exists a path** π **from** s **to** t **in** G_f **do**
4. $c_f(\pi) \leftarrow \min\{c_f(u, v) : (u, v) \text{ is in } \pi\}$
5. **For each edge** (u, v) **in** π , **do**
6. $f[u, v] \leftarrow f[u, v] + c_f(\pi)$ **and** $f[v, u] \leftarrow -f[u, v]$.
7. $c_f[u, v] = c[u, v] - f[u, v]$ **and** $c_f[v, u] = c[v, u] - f[v, u]$.

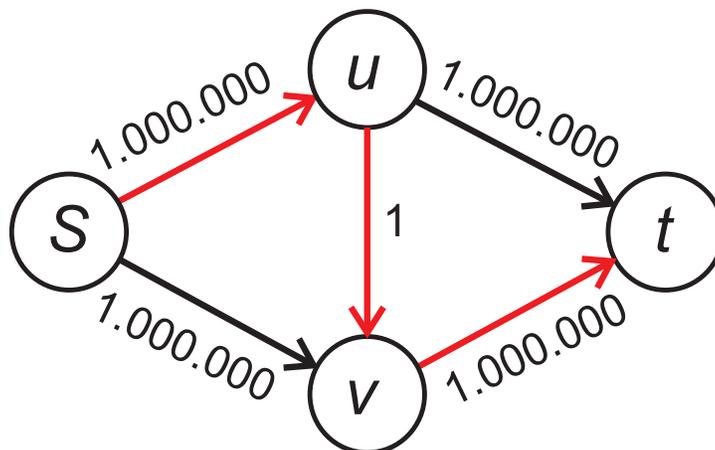
Análise de Complexidade

O tempo de execução depende de como os caminhos são determinados.



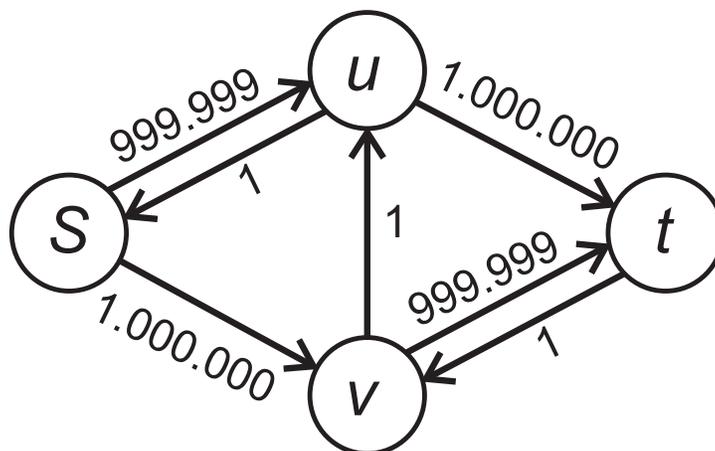
Análise de Complexidade

O tempo de execução depende de como os caminhos são determinados.



Análise de Complexidade

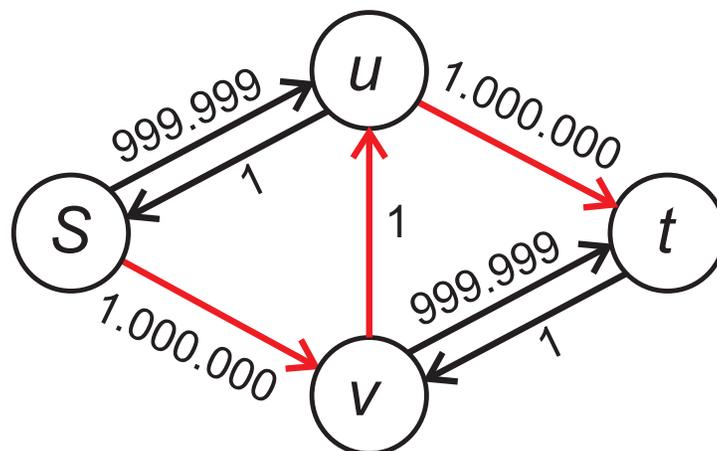
O tempo de execução depende de como os caminhos são determinados.



após 1 iteração.

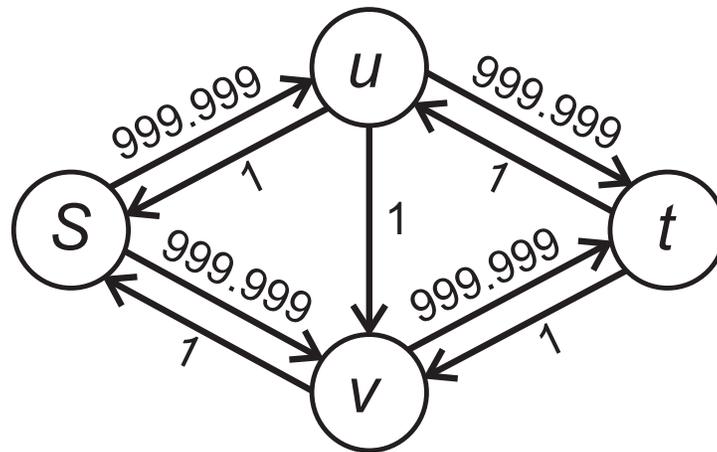
Análise de Complexidade

O tempo de execução depende de como os caminhos são determinados.



Análise de Complexidade

O tempo de execução depende de como os caminhos são determinados.



após 2 iterações.

Se as capacidades forem valores inteiros então o algoritmo executa em $O(|E| \cdot |f^*|)$, onde $|f^*|$ é o valor do fluxo máximo.