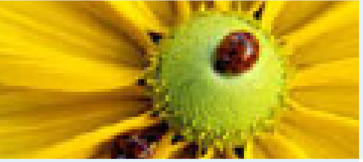
Fluxo Máximo/Corte Mínimo (Min-Cut/Max-Flow Algorithm)

Prof. Dr. Paulo A. V. de Miranda
Instituto de Matemática e Estatística (IME),
Universidade de São Paulo (USP)
pmiranda@vision.ime.usp.br

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?

Definição

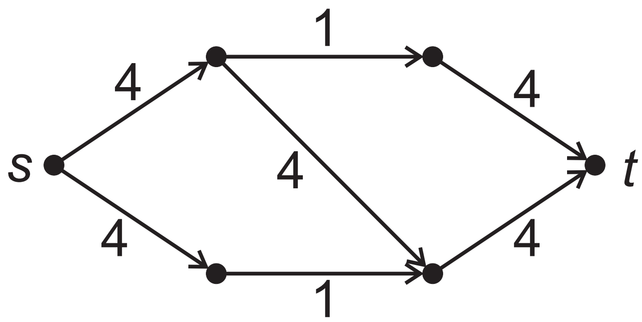
- Uma rede de fluxo $G = (V, E)$ é um grafo direcionado no qual cada arco $(u, v) \in E$ possui uma capacidade não negativa $c(u, v) \geq 0$.

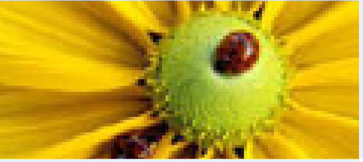
Definição

- Uma rede de fluxo $G = (V, E)$ é um grafo direcionado no qual cada arco $(u, v) \in E$ 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 arco $(u, v) \in E$ 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 dado 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:

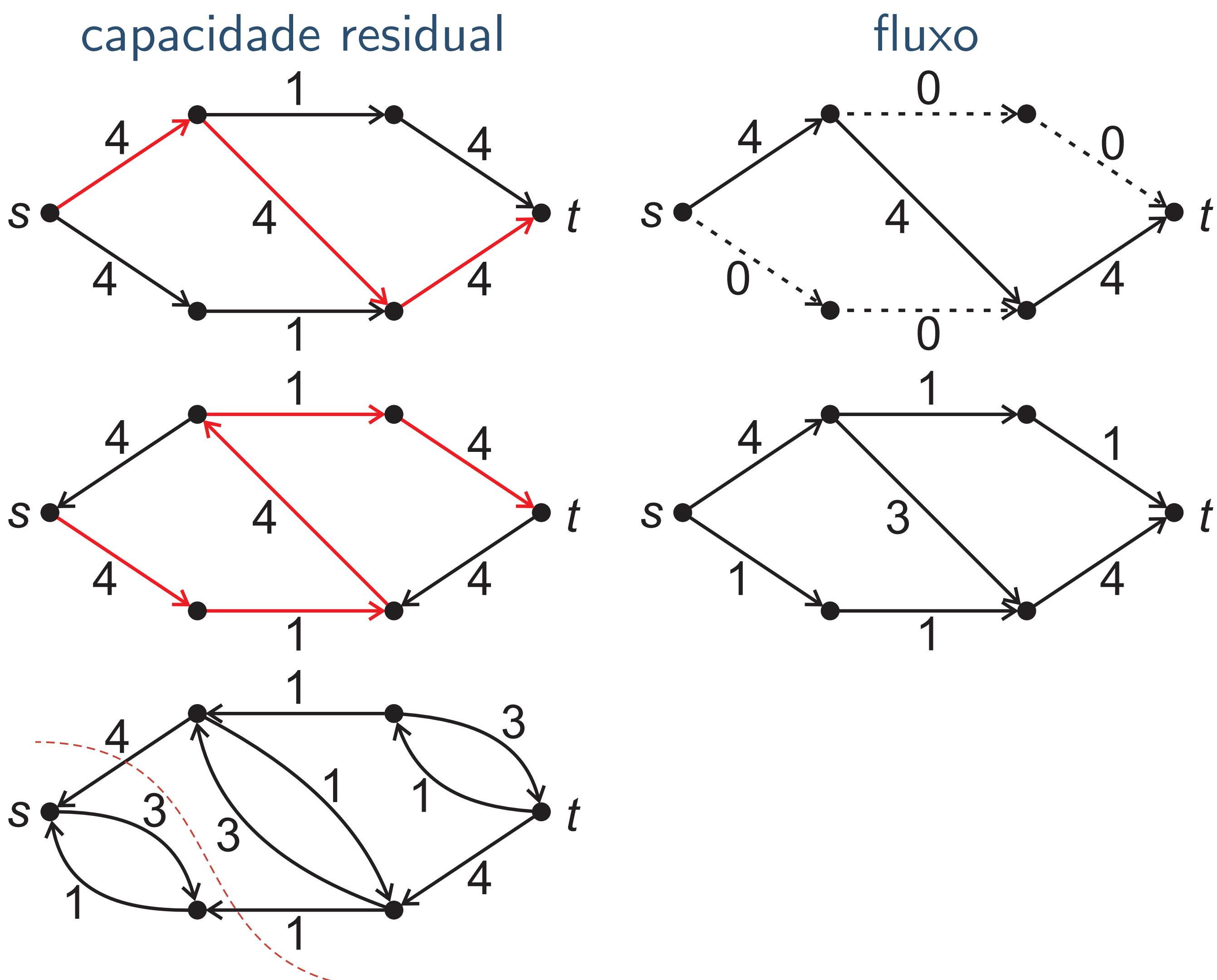
- Começamos com fluxo inicial zero (i.e., $f(u,v) = 0$ para todos $u, v \in V$).
- A cada iteração, aumentamos o fluxo total encontrando algum caminho (a partir da fonte s até o destino t) ao longo do qual podemos empurrar mais fluxo (*augmenting paths*).
- Repetimos este processo até que nenhum caminho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

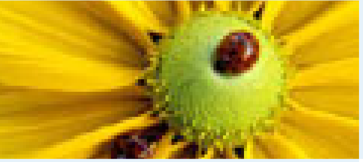
Redes Residuais

- Dados uma rede de fluxo G e um fluxo f, a rede residual consiste de arcos 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





Algorithm 1 – Ford-Fulkerson algorithm

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

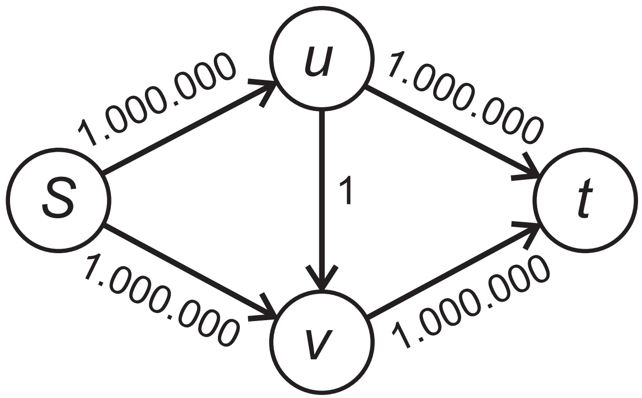
OUTPUT: The maximum flow f in G.

AUXILIARY: The residual network G_f.

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

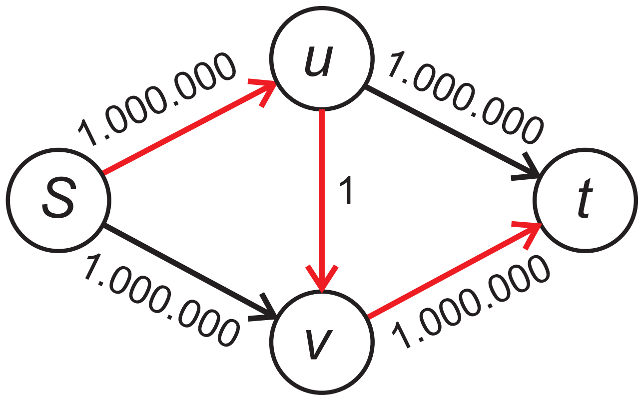
Análise de Complexidade

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



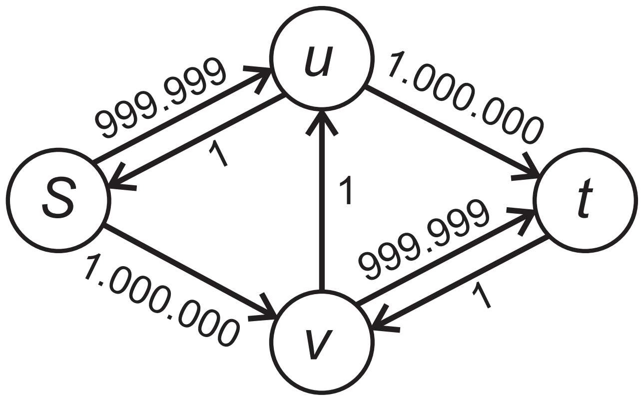
Análise de Complexidade

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



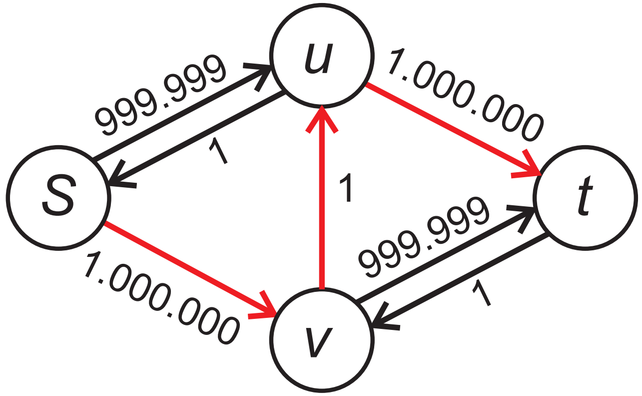
Análise de Complexidade

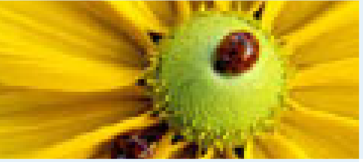
O tempo de execução depende de como os caminhos são determinados.



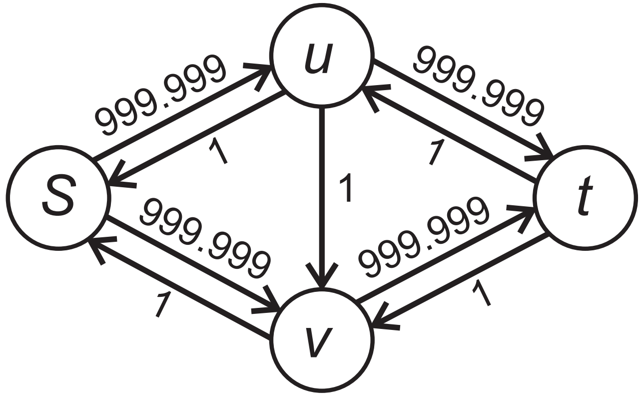
após 1 iteração.

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



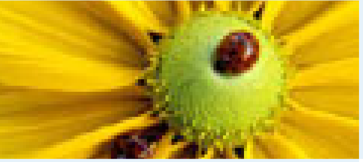


O tempo de execução depende de como os caminhos são determinados.



após 2 iterações.

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

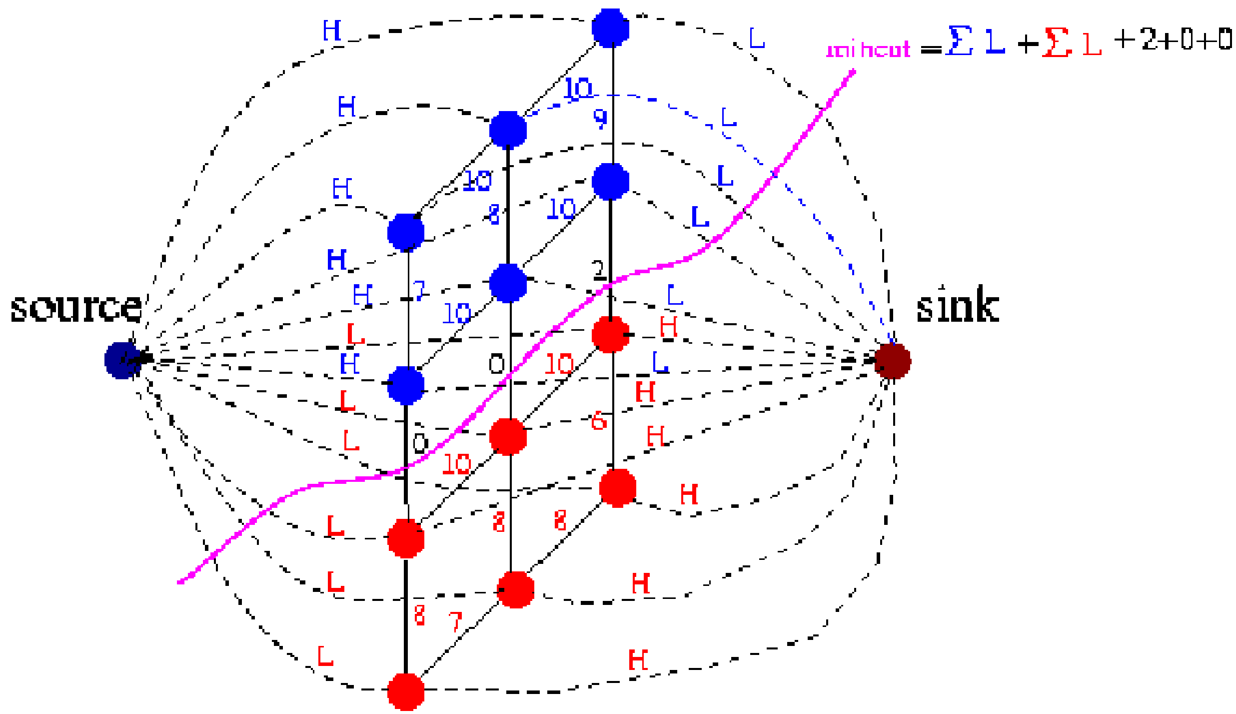


Aplicação em segmentação

- 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

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



min-cut/max-flow solution

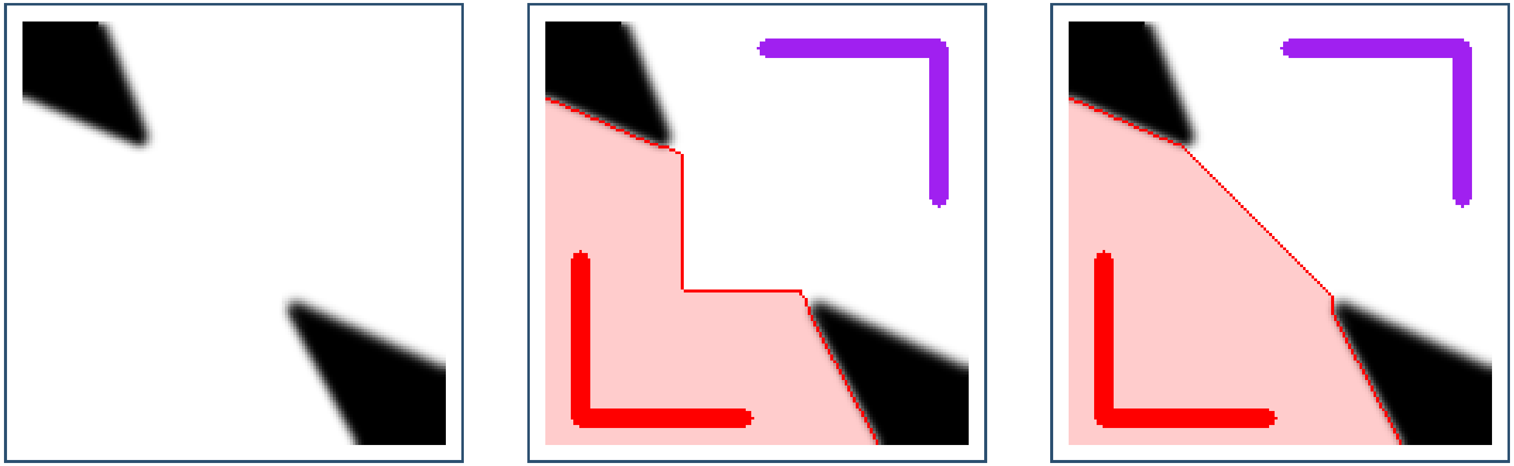
Problemas do Min-Cut/Max-Flow

No contexto de segmentação de imagens, existem duas preocupações na literatura sobre o uso do algoritmo de GC original (min-cut/max-flow):

- *metrication error* ("blockiness"), e
- viés de encolhimento ("shrinking bias").

Problemas do Min-Cut/Max-Flow

O problema métrico ("blockiness") surge quando calculamos o fluxo máximo em um grafo de imagem com vizinhança-4. As figuras abaixo mostram o problema.



(a) (b) (c)

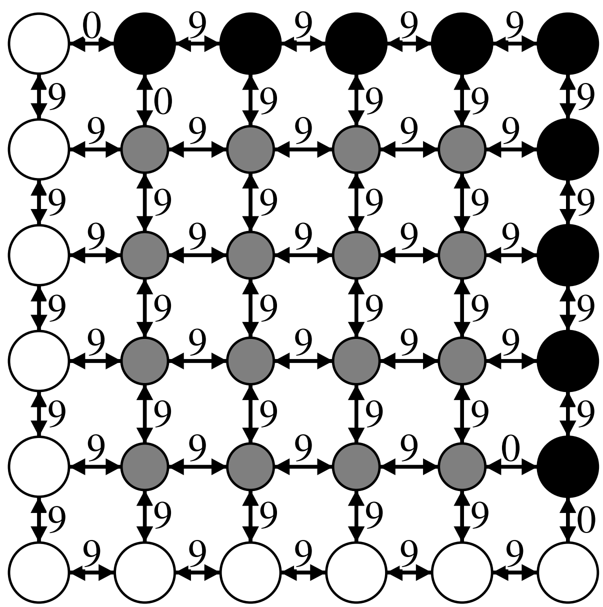
Claramente, o GC em vizinhança-4 está produzindo um corte irregular **(Fig. b)**, em vez da borda suave esperada. Um melhor resultado pode ser obtido usando GC com vizinhança-8 **(Fig. c)**.

Problemas do Min-Cut/Max-Flow

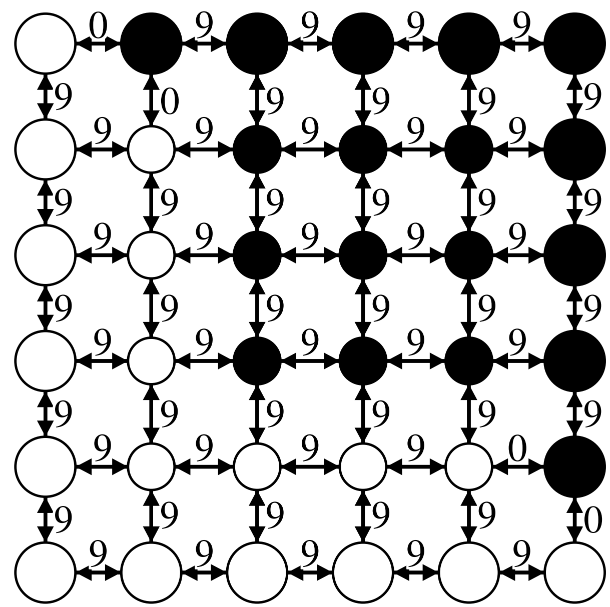
- Isso acontece porque, em um grafo de vizinhança-4, uma borda em diagonal corta o mesmo número de arcos que uma borda em formato de “L” (formando um canto de 90 graus).
- Isto contradiz a nossa expectativa, visto que a diagonal é o caminho mais curto entre os pontos considerados, o que levaria (intuitivamente) para um corte de menor valor.

Problemas do Min-Cut/Max-Flow

A fim de compreender melhor esse fenômeno, a figura abaixo ilustra o mesmo problema numericamente.



(a)

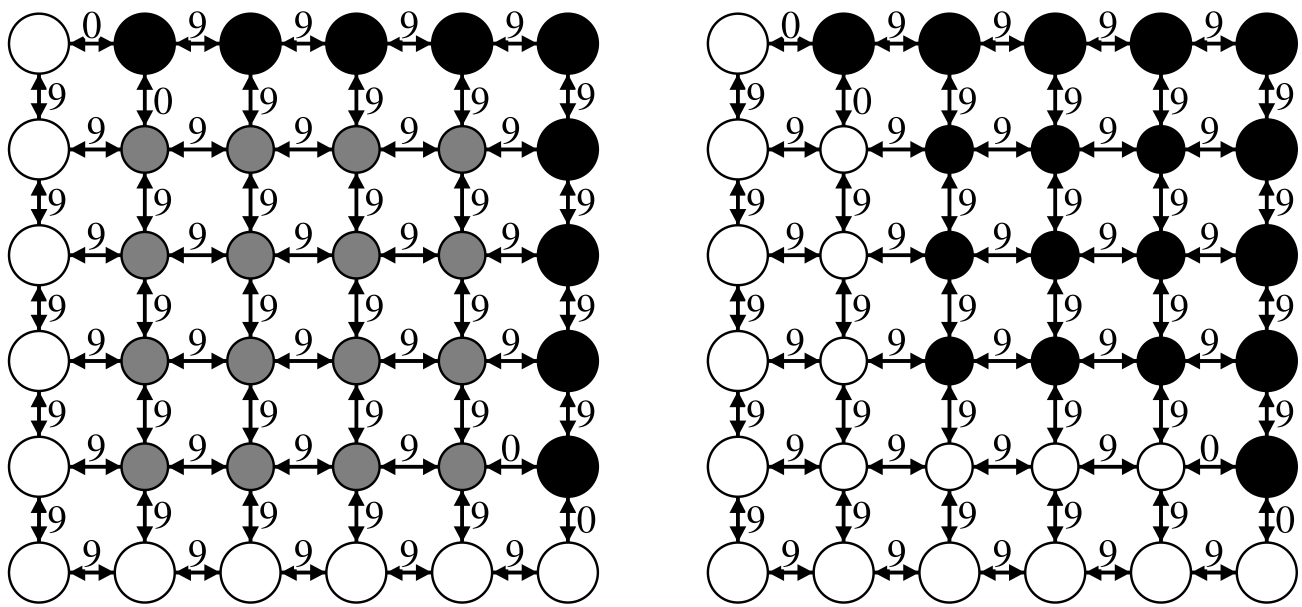


(b)

(a) Um grafo de vizinhança-4 onde os números indicam as capacidades dos arcos, com sementes de fundo em preto e sementes de objeto em branco.

Problemas do Min-Cut/Max-Flow

A fim de compreender melhor esse fenômeno, a figura abaixo ilustra o mesmo problema numericamente.

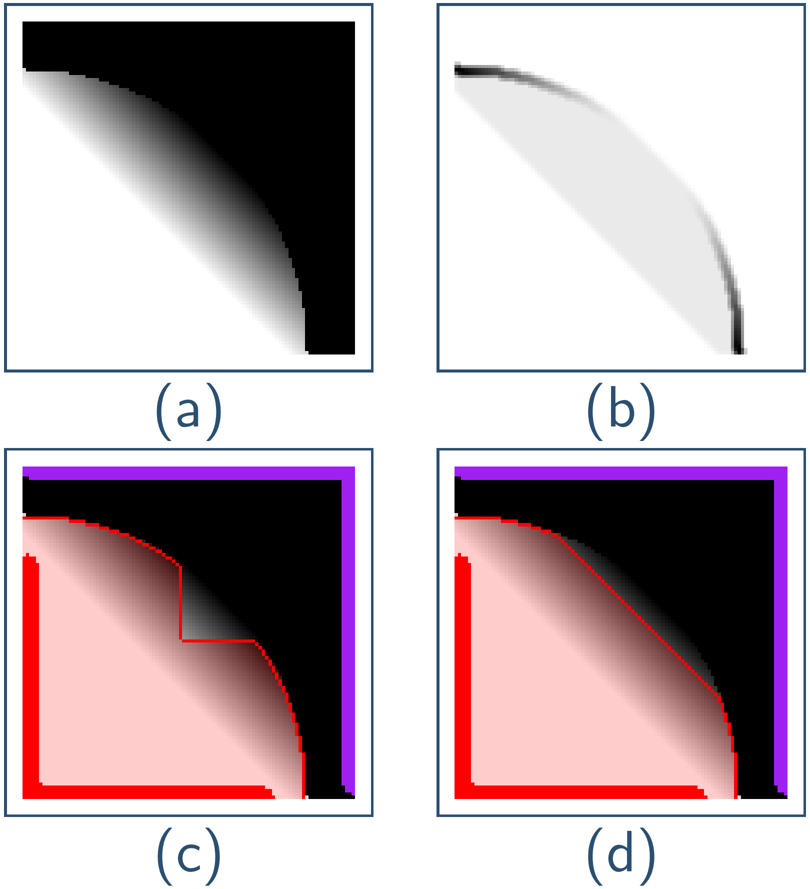


(a) (b)

(b) A segmentação por GC apresenta um efeito de "blockiness", formando um canto (corte irregular) ao invés da fronteira suave esperada.

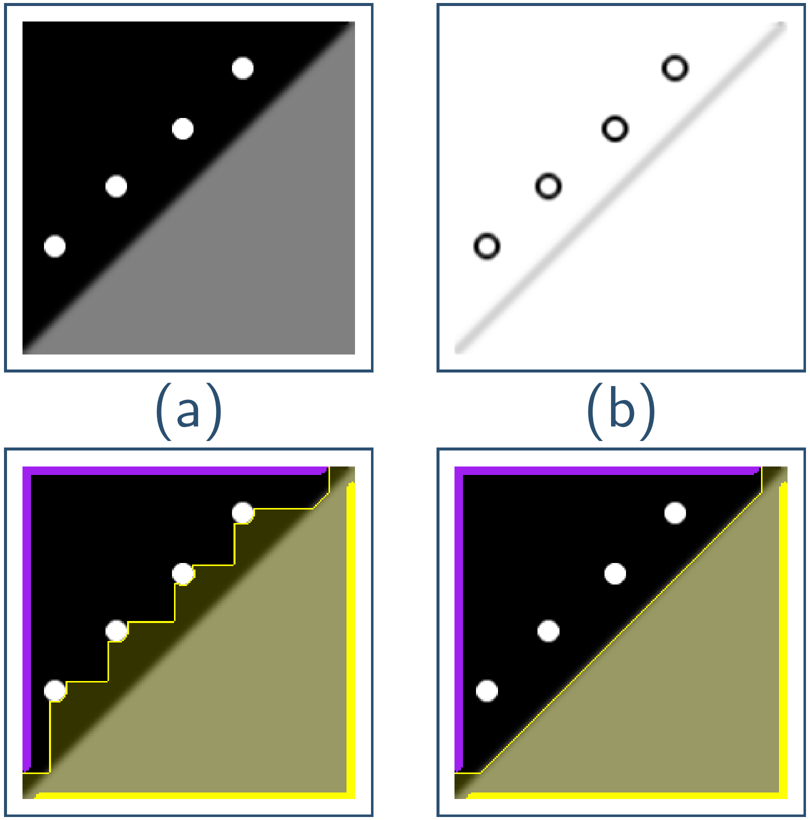
Problemas do Min-Cut/Max-Flow

Outros exemplos:



(a) Imagem original, (b) imagem das capacidades dos arcos, (c) GC com vizinhança-4, (d) GC com vizinhança-8.

Outros exemplos:



(a) (b) (c) (d)

(a) Imagem original, (b) imagem das capacidades dos arcos, (c) GC com vizinhança-4, (d) GC com vizinhança-8.