

Teoria dos Grafos

Aula 27

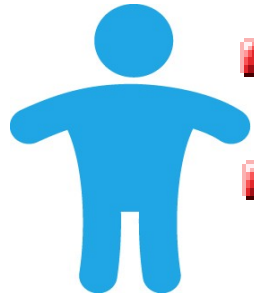
Aula passada

- Algoritmo de Ford-Fulkerson
- Análise do algoritmo
- Melhorando algoritmo inicial

Aula de hoje

- Aplicações do fluxo máximo
- Emparelhamento perfeito
- Caminhos distintos
- Corte mínimo

Formando Pares



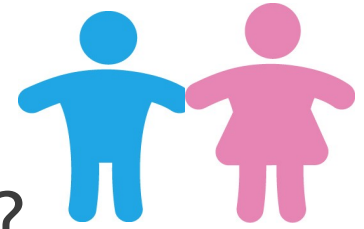
- N rapazes
- Cada rapaz declara interesse em uma ou mais moça



- N moças
- Cada moça declara interesse em um ou mais rapaz

- Casal pode “sair junto” (formar um par) se existe **interesse mútuo**
- **Problema 1:** Qual maior número de pares que podemos formar?
- **Problema 2:** Quais pares devemos formar?

Formando Pares

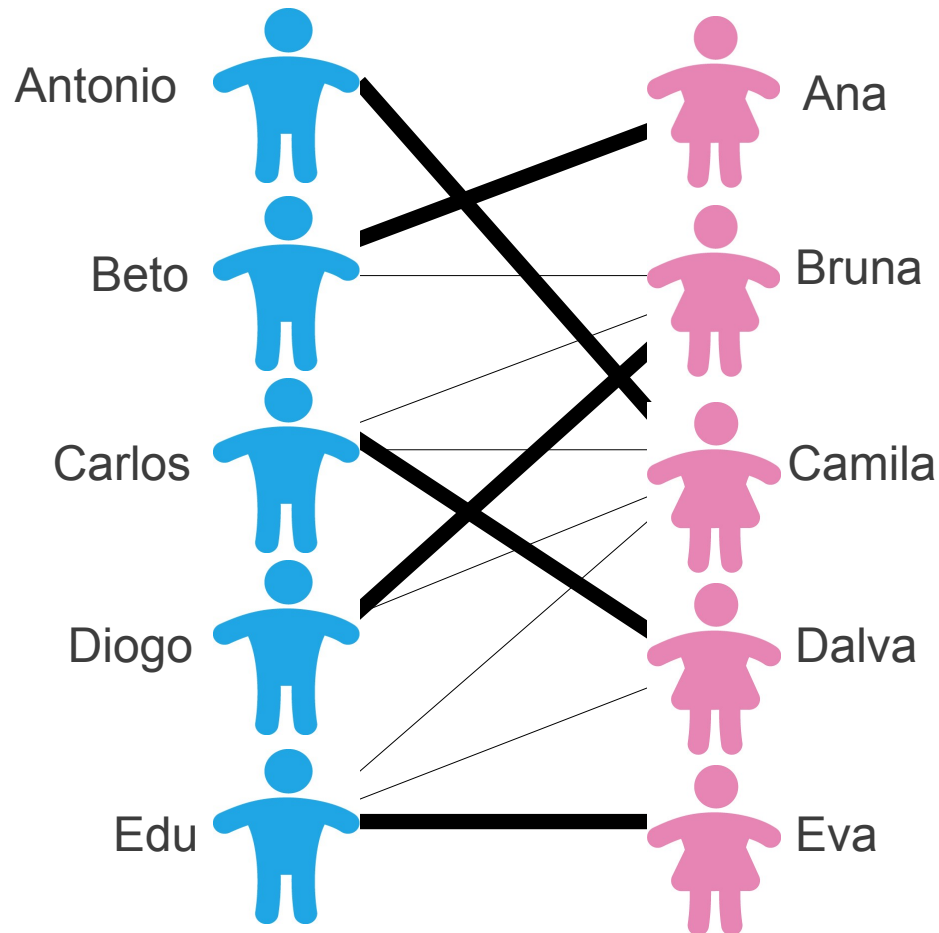


- Como abstrair o problema (usando grafos)?
- Objeto: pessoas (rapazes e moças)
- Relacionamento: interesse mútuo em sair

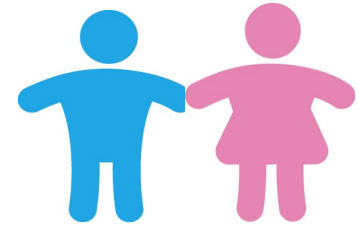
Exemplo:

Ana e Beto
têm interesse
mútuo!

Podemos
formar 5 pares?



Problema Genérico



- Emparelhamento em grafos bipartites
- Grafo bipartite (ou bipartidos)
 - dois tipos de vértices, arestas apenas entre tipos diferentes
- Determinar maior *emparelhamento*
 - emparelhamento: formação de pares
 - perfeito: todos vértices estão emparelhados

Algoritmo para problema genérico?

Aplicação de Fluxo Máximo

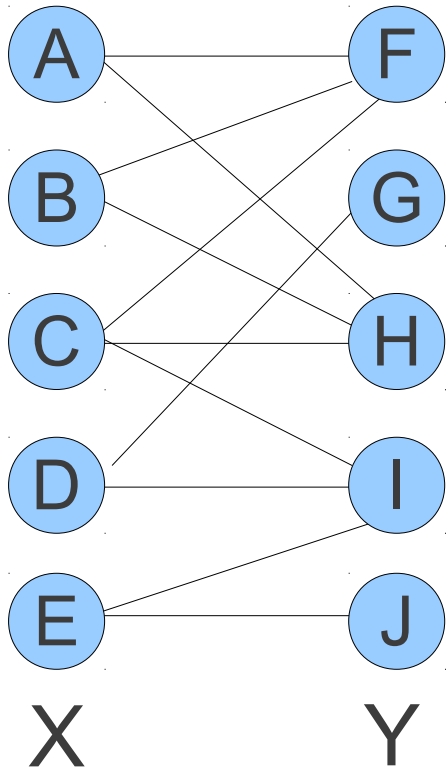
- **Ideia:** Converter problema original em problema de fluxo máximo

Como?

- Construir redes de fluxo direcionada
- Adicionar vértice s
 - conectar ao vértices do conjunto X
- Adicionar vértice t
 - conectar vértices do conjunto Y a t
- Direcionar arestas do grafo original $X \rightarrow Y$
- Capacidade 1 em todas as arestas

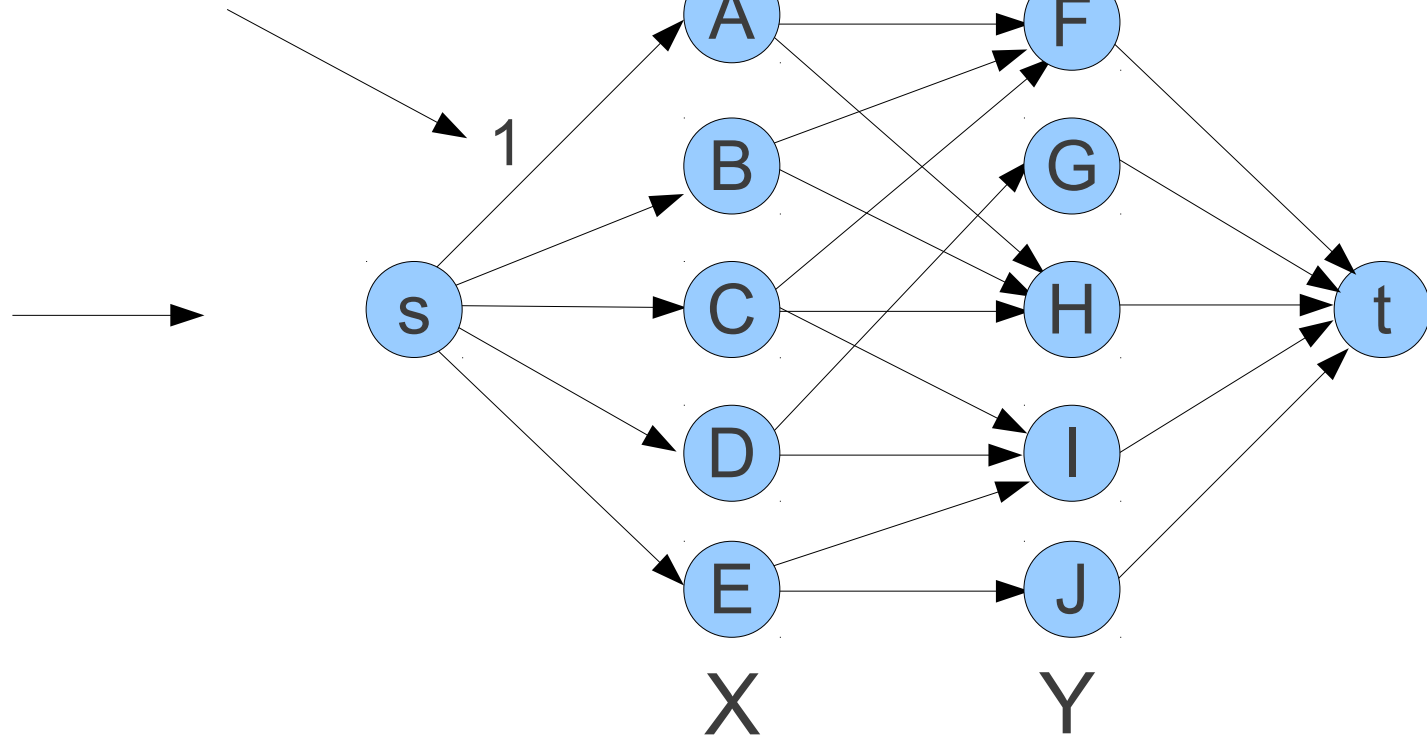
Exemplo

Original (G)



Capacidades
todas iguais a 1

Rede de fluxos (G')



- Hipótese: Fluxo máximo em G' é igual ao maior emparelhamento em G !

Provando Hipótese (1/2)

- Emparelhamento com k pares em G tem fluxo k em G'
 - cada par contribui unidade de fluxo
- Fluxo com valor k em G' emparelha k pares em G
 - assumir integridade de fluxo (fluxo inteiros):
 $f(e) = 0$ ou $f(e) = 1$
 - arestas com fluxo são arestas emparelhadas
- Seja M' conjunto de arestas com $f(e) = 1$
 - M' possui k arestas

Provando Hipótese (2/2)

- Cada vértice em X incidente em no máximo uma aresta em M'
 - Conservação de fluxo
- Cada vértice em Y incidente em no máximo uma aresta em M'
 - Conservação de fluxo
- Valor do fluxo máximo em G' é igual ao maior emparelhamento em G
- Vértices com $f(e) = 1$ correspondem aos vértices do emparelhamento

Complexidade

- Algoritmo original: $O(mC)$
- Algoritmo modificado: $O(m^2 \log C)$
- Quanto vale C ?
- $C = O(n)$
- Custo via algoritmo original: $O(mn)$
- Custo menor via algoritmo original!
- Porque?

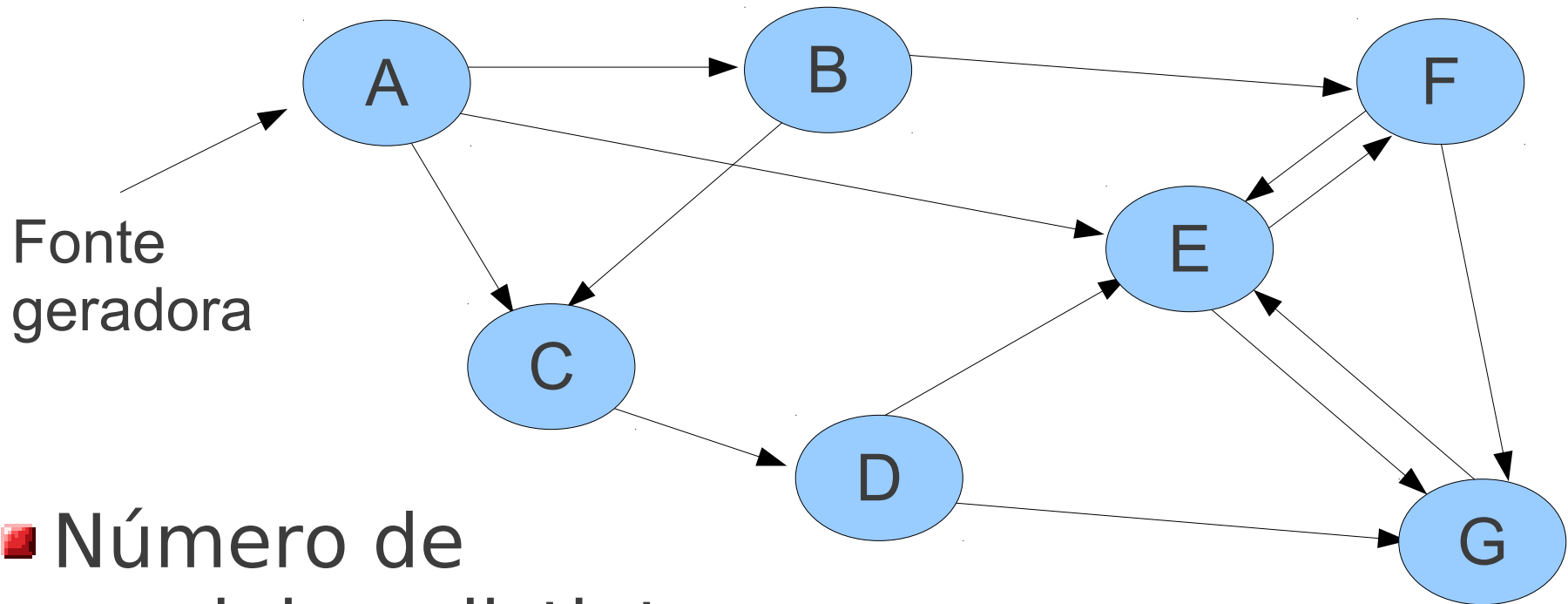
Robustez da Malha Elétrica

- Malha elétrica (distribuição de energia)
- Torres e linhas de transmissão
- **Robustez:** número de caminhos distintos (em linhas)
- **Problema:** Determinar robustez entre fonte geradora e ponto consumidor



Robustez da Malha Elétrica

- Objetos: torres de transmissão
- Arestas: linhas entre torres



- Número de caminhos distintos (em arestas)?

Problema Genérico

- Dado grafo direcionado G
 - e dois vértices s, t quaisquer
- Encontrar número de caminhos distintos
 - em termos de arestas
 - podemos repetir vértices
- Encontrar os caminhos em si
 - não só o número

Algoritmo para problema genérico?

Aplicação de Fluxo Máximo

- **Ideia:** Converter problema original em problema de fluxo máximo

Como?

- Construir rede de fluxos
 - s e t são origem/destino da rede
 - remover arestas entrada s , saída t
 - capacidade 1 em todas outras arestas
 - assumir fluxo inteiro

Hipótese e Prova

- Fluxo máximo s-t é igual número de caminhos distintos entre s e t
- Se existirem k caminhos distintos s-t, então fluxo máximo será ao menos k
 - cada caminho carrega fluxo 1, independente
- Se existir fluxo máximo k, então temos k caminhos distintos
 - Assumir fluxo inteiro
 - Intuição: cada aresta com $f(e) = 1$ pertence a exatamente 1 caminho (algum)

Complexidade

- Algoritmo original: $O(mC)$
- Quanto vale C ?
- $C = O(n)$
- Custo via algoritmo original: $O(mn)$
- Custo menor via algoritmo original
 - De novo!

Grafos Não-Direcionados

- Considerar grafos não-direcionados
 - E dois vértices quaisquer s, t
- Número de caminhos distintos entre eles?
 - Em termos de arestas
- Caminho entre eles?
 - Não só o número

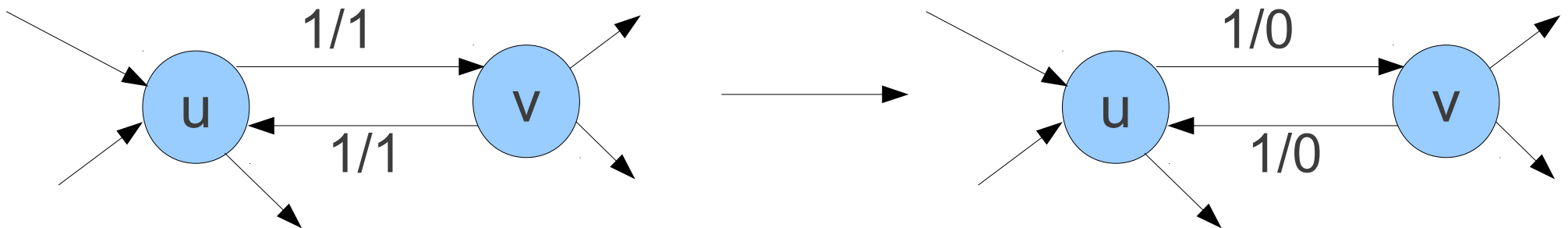
Como estender modelo anterior?

Aplicação de Fluxo Máximo

- Construir rede de fluxos direcionada
- Para cada aresta (u, v) original (não direcionada)
 - Adicionar aresta (u, v) direcionada
 - Adicionar aresta (v, u) direcionada
- Continuar como caso direcionado
 - remover arestas entrando s e saindo t , capacidade 1 em todas arestas, assumir fluxos inteiros
- **Problema:** apenas uma das arestas pode ser utilizada
 - aresta é única no grafo não-direcionado

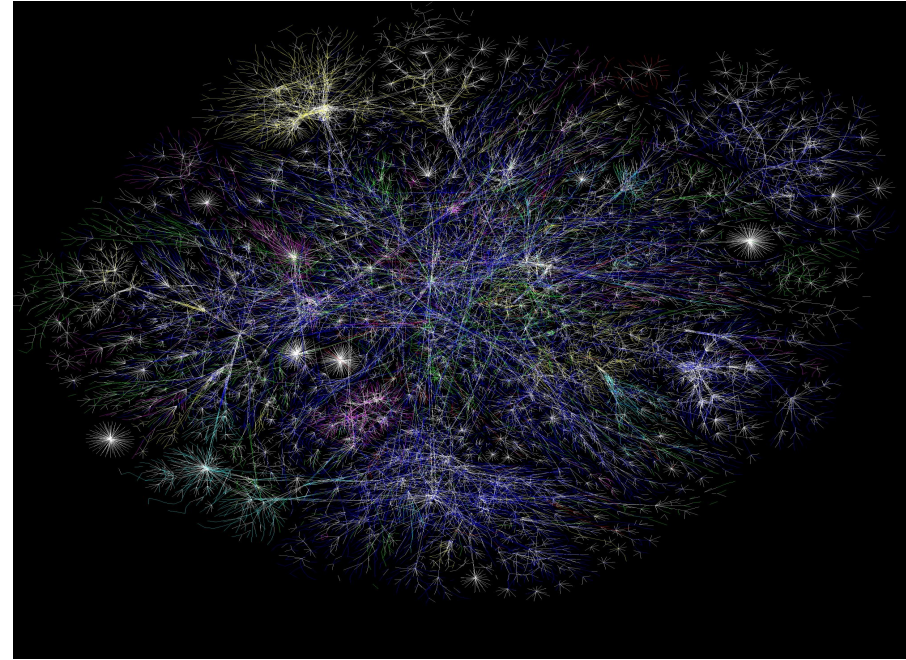
Aplicação de Fluxo Máximo

- Problema não será problema!
- Fluxo máximo existe utilizando aresta em apenas **uma** das direções
- Supor duas direções sendo utilizadas
- Novo fluxo onde nenhuma das duas é utilizada tem mesmo valor e é um fluxo



Conectividade na Internet

- Internet: rede de redes
- Conectividade entre AS (sistemas autônomos)
- **Robustez:**
conectividade global



- **Problema:** Determinar robustez da Internet
 - Número mínimo de arestas que precisam falhar para desconectar o grafo

Problema Genérico

- Problema do corte mínimo em grafos não-direcionados
- **Corte:** conjunto de arestas que se removidas “desconectam” o grafo
 - geram mais de um componente conexo
- Tamanho do corte: número de arestas que o compõem
- Corte mínimo: corte com o menor número de arestas possível

Algoritmo para problema genérico?

Aplicação de Fluxo Máximo

- **Ideia:** Converter problema original em problema de fluxo máximo

Como?

- Escolher dois vértices s e t quaisquer
- Construir rede de fluxos s - t
 - Como no caso anterior, caminhos distintos
- Obter corte mínimo entre s - t
 - corte mínimo é igual ao fluxo máximo, que é igual ao número de caminhos distintos
- Corte mínimo global?

Aplicação de Fluxo Máximo

- Problema: corte mínimo s - t não necessariamente é corte mínimo global?

Solução?

- Fixar s , variar t para cada vértice do grafo
- Determinar corte mínimo para cada t
 - via redes de fluxo
- Corte mínimo global é o menor deles
- **Obs:** corte mínimo global necessariamente separa s de t para algum par s e t

Complexidade

- Fixar s , variar t
 - Total de $n-1$ rodadas
- Algoritmo original: $O(mC)$
- Quanto vale C ?
- $C = O(n)$
- Custo de cada rodada: $O(mn)$
- Custo total: $O(mn^2)$
- Mas existem algoritmos mais eficientes específicos para este problema

Segmentação de Imagens

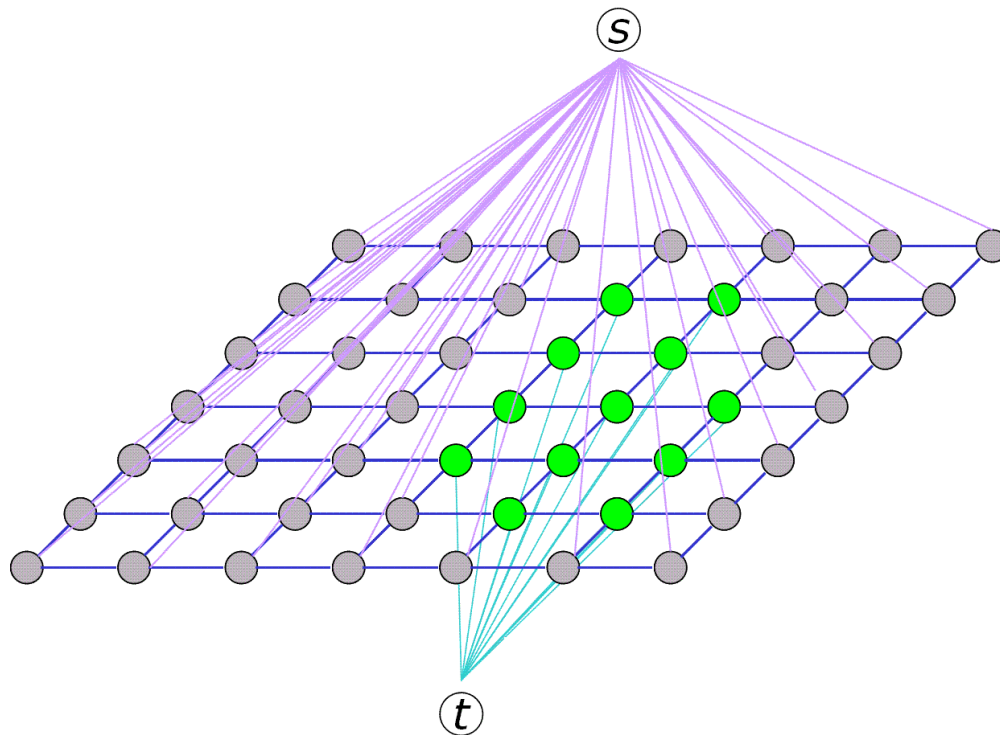
- Remoção de fundo
 - Separar o que é objeto de interesse do fundo
- **Problema:** quais pixels pertencem ao objeto, quais pertencem ao fundo (background)
- Como modelar o problema de imagens como um grafo?
- Como o algoritmo de fluxo máximo/corte mínimo pode ajudar?

Segmentação de Imagens

- Remoção de fundo
 - Separar o que é objeto de interesse do fundo
- **Problema:** quais pixels pertencem ao objeto, quais pertencem ao fundo (background)
- Como modelar o problema de imagens como um grafo?
- Como o algoritmo de fluxo máximo/corte mínimo pode ajudar?

Segmentação de Imagens

- Modelando o problema como grafo
 - Que pesos colocar nas arestas?



Segmentação de Imagens

- Ajuda do usuário
 - Indicar alguns pixels como foreground ou background
 - Montar histogramas

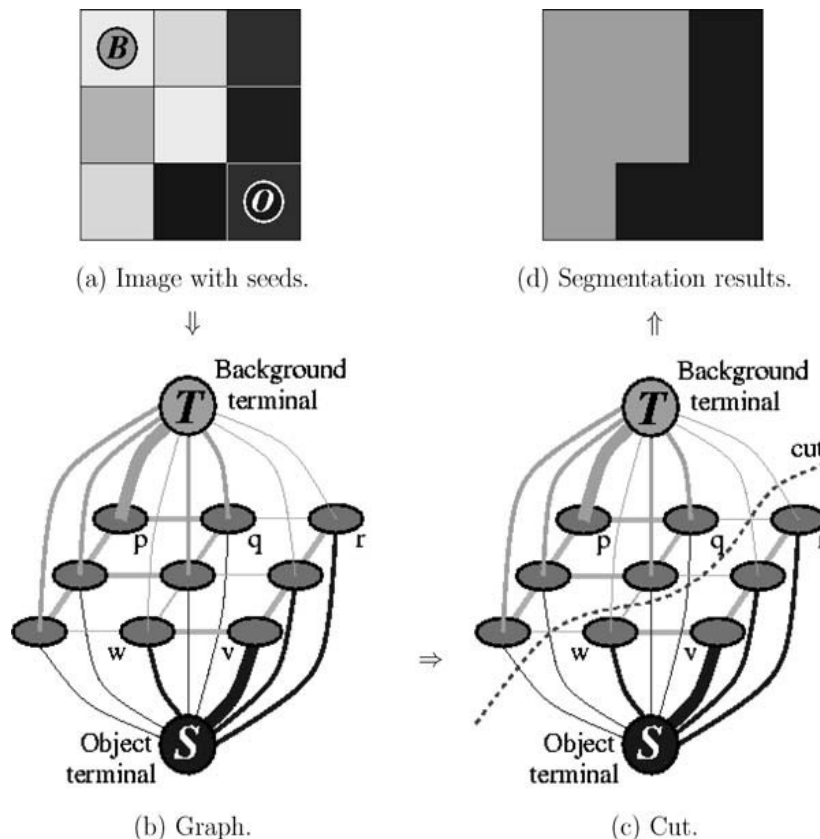


Segmentação de Imagens

- Peso das arestas:
 - Termo de fronteira (arestas entre pixels): diferença entre pixels vizinhos
 - Termo regional (arestas saindo de s ou chegando a t): histogramas, $\text{Prob}(\text{bg})$ e $\text{Prob}(\text{fg})$

Segmentação de Imagens

- Quanto maior o peso (mais similar) mais difícil de “cortar” a aresta



Graph Cuts (Boykov)

