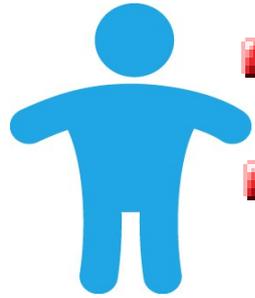


# Grafos – Aula 18

## Roteiro

- Aplicações do fluxo máximo
- Emparelhamento
- Caminhos distintos
- Corte mínimo
- Segmentação de imagens

# Formando Pares



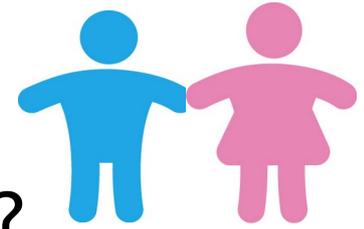
- $N$  rapazes
- Cada rapaz declara interesse em uma ou mais moça



- $N$  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

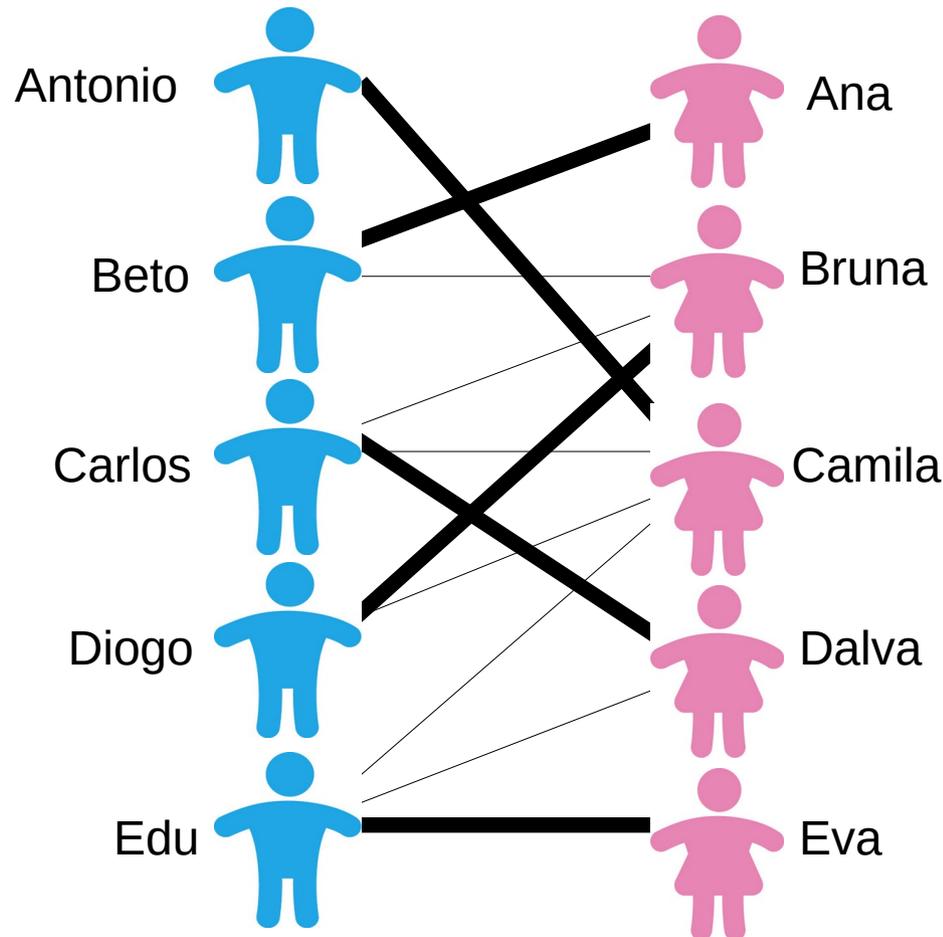


- Como abstrair o problema (usando grafos)?
- Objeto: pessoas (rapazes e moças)
- Relacionamento: interesse mútuo em sair

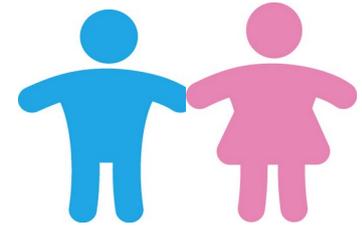
Exemplo:

Ana e Beto  
têm interesse  
mútuo!

**Podemos  
formar 5 pares?**



# Problema Genérico



- Emparelhamento em grafos bipartido
- Determinar maior *emparelhamento*
  - emparelhamento: formação de pares
  - perfeito: todos vértices estão emparelhados

**Algoritmo para problema genérico!**

- Vimos em aula (caminho aumentante, etc)

# Aplicação de Fluxo Máximo

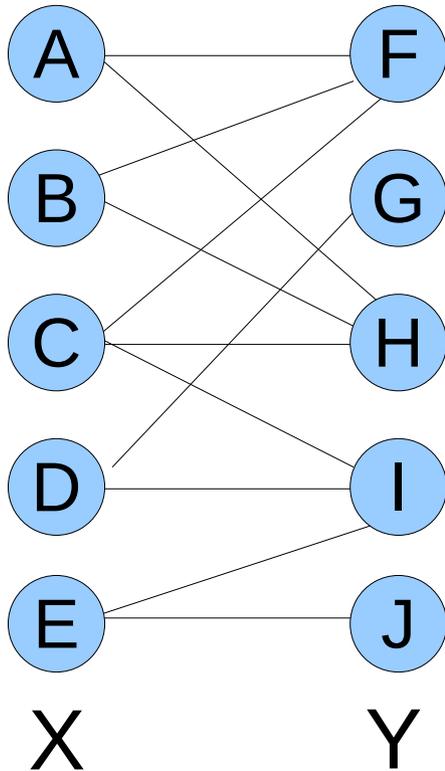
- **Ideia:** Converter problema original em problema de fluxo máximo

## 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

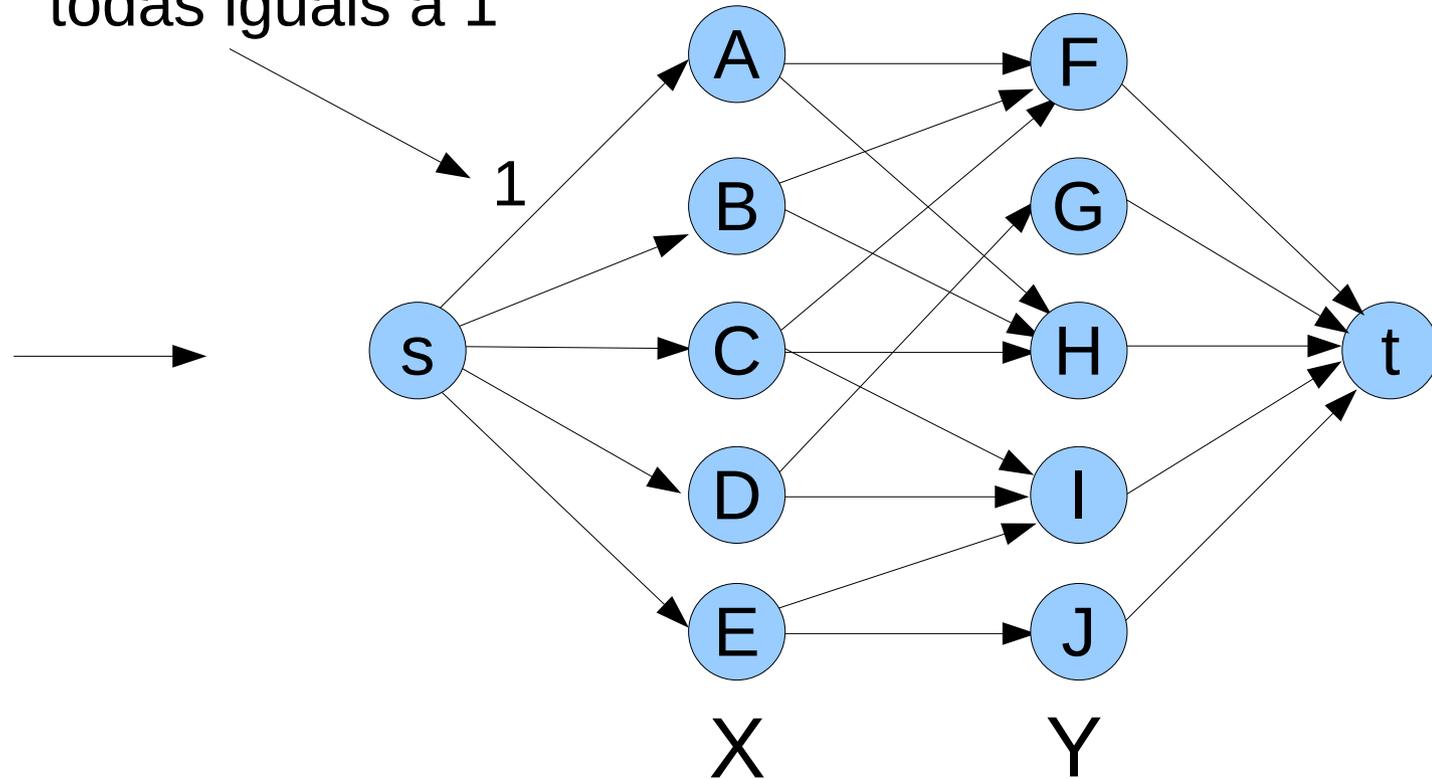
# Exemplo

## 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)

- 1) Emparelhamento com  $k$  pares em  $G$  tem fluxo  $k$  em  $G'$ 
  - cada par contribui unidade de fluxo
- 2) Fluxo com valor  $k$  em  $G'$  emparelha  $k$  pares em  $G$ 
  - assumir integralidade 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, capacidade de entrada
- Cada vértice em  $Y$  incidente em no máximo uma aresta em  $M'$ 
  - conservação de fluxo, capacidade de saída
- 3) Valor do fluxo máximo em  $G'$  é igual ao maior emparelhamento em  $G$ 
  - arestas com  $f(e) = 1$  correspondem às arestas do emparelhamento

# Complexidade

- Algoritmo original:  $O(mC)$
- Algoritmo modificado:  $O(m^2 \log C)$
- Limitante superior para  $C$ 
  - $C = n$  (capacidade de saída de  $s$ )
- Custo via algoritmo original:  $O(mn)$
- Algoritmo original tem custo menor
  - valor de  $C$  é da ordem de  $n$
- Mesmo custo que algoritmo específico para emparelhamento de aula passada

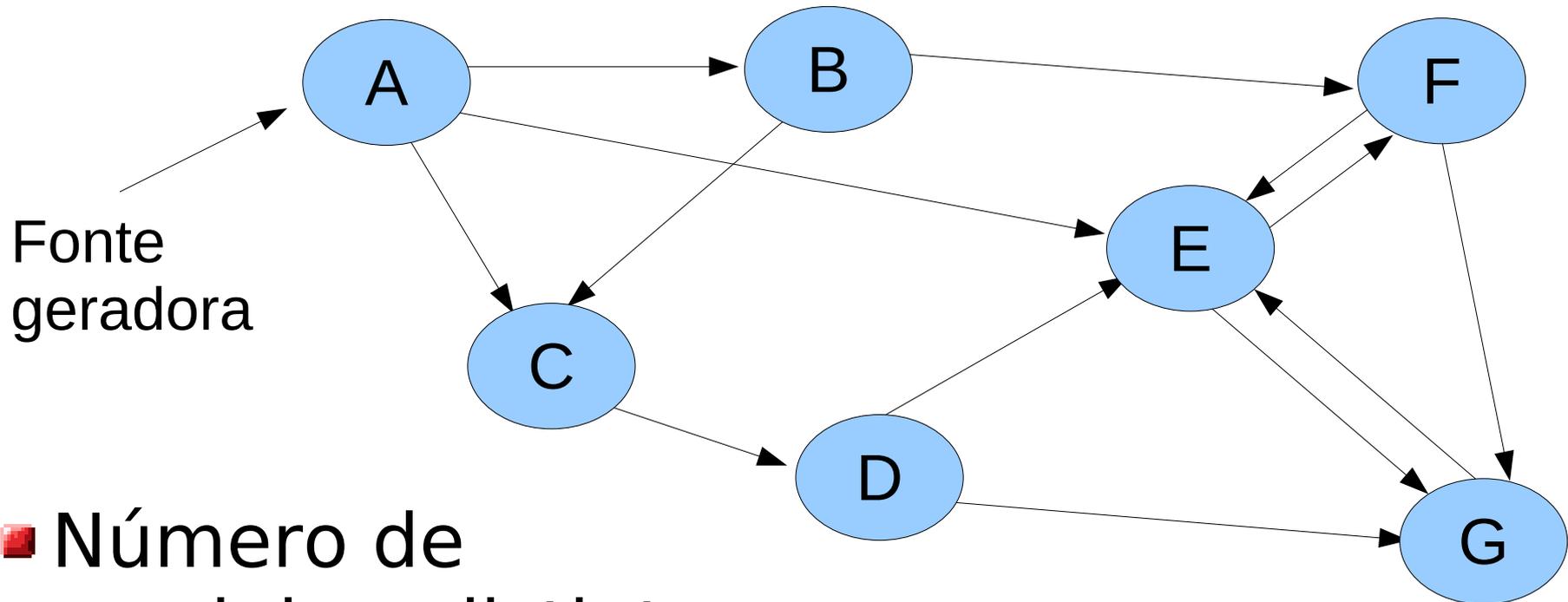
# Robustez da Malha Elétrica

- 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 também os caminhos
  - não somente o número de caminhos

**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 pelo 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:  $O(mC)$
- $C = \min(d_s, d_t)$ , onde  $d_i$  é grau do vértice  $i$ 
  - $C \leq g_{\max}$
- Custo via algoritmo original:  $O(mg_{\max})$
- Custo menor via algoritmo original
  - $C$  é da ordem do maior grau

# Grafos Não-Direcionados

- Considerar grafos não-direcionados
  - e dois vértices quaisquer  $s, t$
- Número de caminhos distintos entre eles?
  - em termos de arestas
- Quais os caminhos entre eles?
  - e não só o número de caminhos

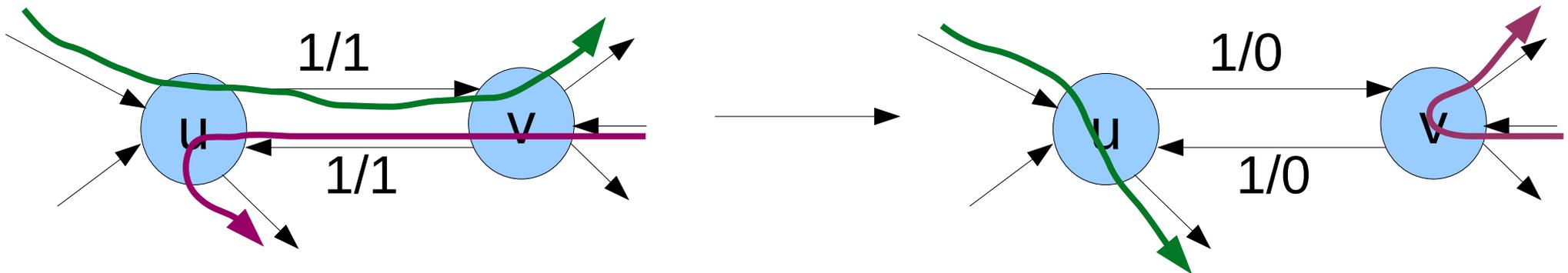
**Como estender modelo anterior?**

# Aplicação de Fluxo Máximo

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

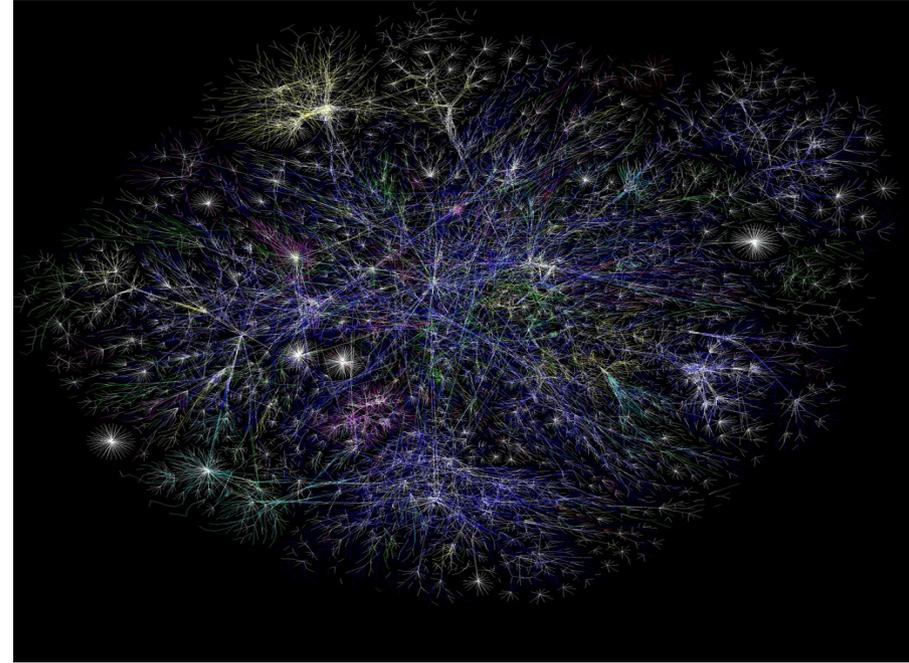
# Aplicação de Fluxo Máximo

- Problema não será um 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 válido



# Conectividade na Internet

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



- **Problema:** Determinar robustez da Internet
  - Número mínimo de enlaces que precisam falhar para desconectar a rede

# Problema Genérico

- Problema do corte mínimo em grafos não-direcionados
- **Corte:** conjunto de arestas que se removidas “desconectam” o grafo
  - induzem mais de uma componente conexa
- Tamanho do corte: número de arestas que define o corte
- Corte mínimo: corte com o menor número de arestas possível (não há pesos)

**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, para todos valores de  $t$
- Algoritmo original:  $O(mC)$
- Valor de  $C \leq g_{\max}$ 
  - fixar  $s$  em grau mínimo:  $C = g_{\min}$
- Custo de cada rodada:  $O(mg_{\min})$
- Custo total:  $O(mng_{\min})$
- Existem algoritmos específicos mais eficientes para este problema

# Segmentação de Imagens

- Remoção de fundo: separar o que é objeto de interesse do restante da imagem



- Problema fundamental em visão computacional

- **Entrada:** imagem de  $w \times h$  pixels, cada pixel possui uma cor
- **Problema:** quais pixels pertencem ao objeto, quais pertencem ao fundo?

# Imagem vira Grafo

- Resolver o problema de imagens utilizando grafos
- Vértices: cada pixel é um vértice
- Arestas: pixels adjacentes na imagem
  - grau máximo 4

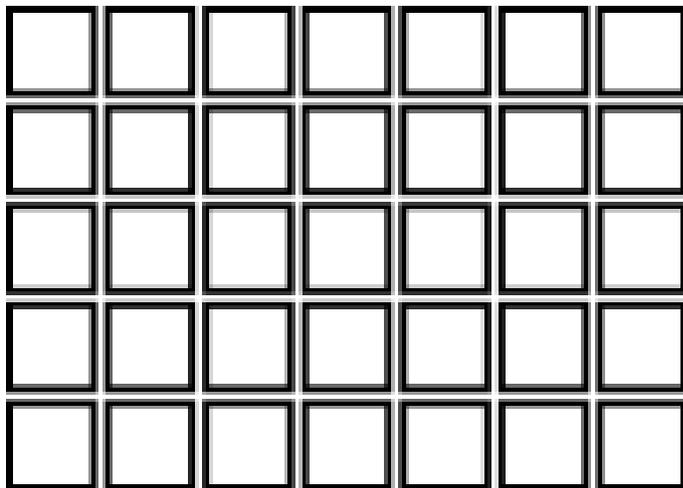
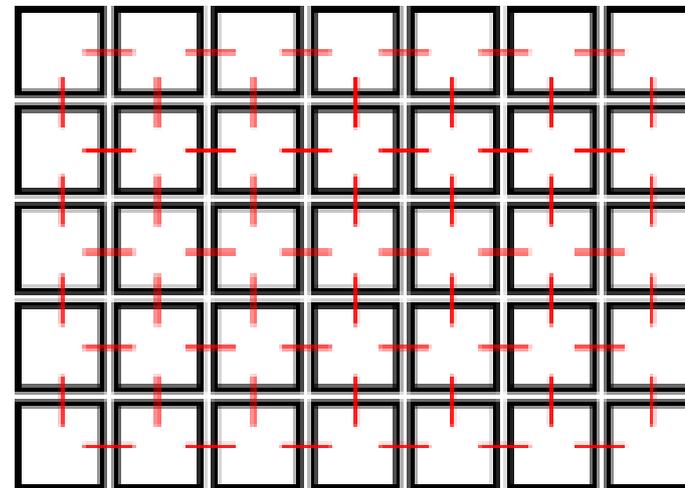


Imagem (pixels)



Grafo (vértices)

# Graph Cuts em Visão Computacional

- Problemas de visão transformados em problemas de fluxo máximo em redes
  - incluindo remoção de fundo
- Algoritmo de Boykov & Kolmogorov resolve este problema de forma eficiente
  - muito usado nas principais bibliotecas de visão
- Exemplo

