Métodos de Apoio à Decisão

Problemas de Otimização em Grafos

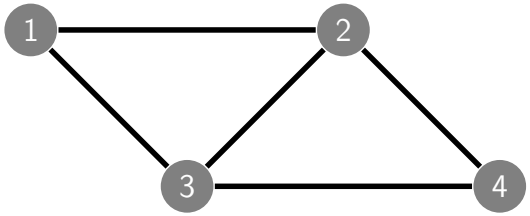
João Pedro Pedroso

2024/2025

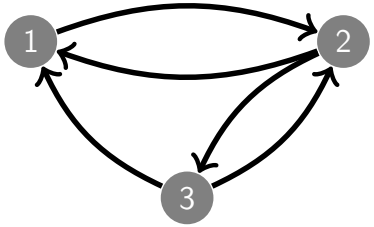
- Última aula:
  - Problema de coloração de grafos
  - Modelos:
    - como formular o problema
    - como melhorar a formulação
- Hoje:
  - Problema da partição de grafos
  - Problema do conjunto estável máximo
  - Problema da clique máxima

Graphs

- Grafos não dirigidos: as linhas que ligam dois vértices não têm uma direção implícita
  - ligações sem direção $\rightarrow$ arestas (*edges*)
  - exemplo: $G = (V, E)$
    $V = \{1, 2, 3, 4\}$, $E = \{\{1, 2\}, \{2, 3\}, \{2, 4\}, \{3, 1\}, \{3, 4\}\}$



- Grafos dirigidos: direção nas linhas que ligam dois vértices é importante
  - ligações dirigidas $\rightarrow$ arcos (*arcs*)
  - exemplo: $G = (V, A)$
    $V = \{1, 2, 3\}$, $A = \{(1, 2), (2, 1), (2, 3), (3, 2), (3, 1)\}$



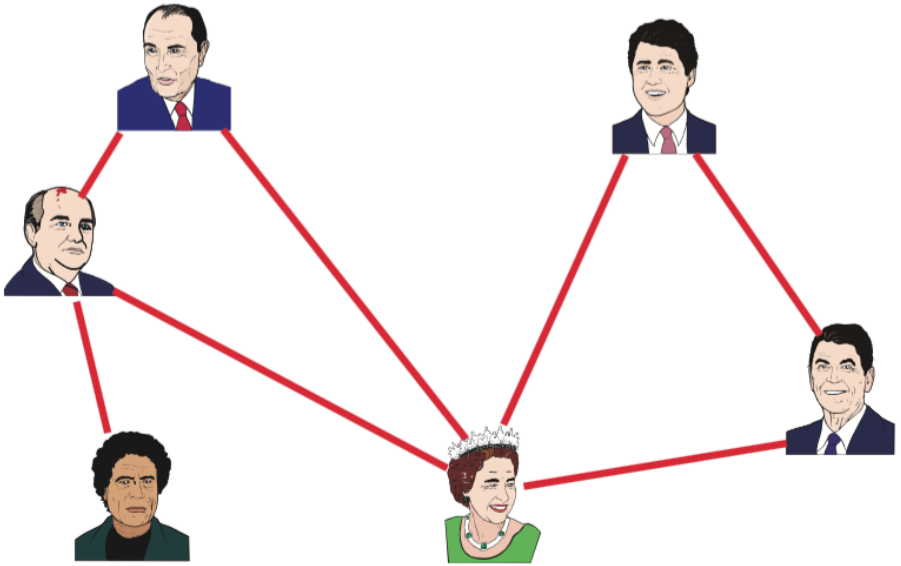
Problema da partição de grafos

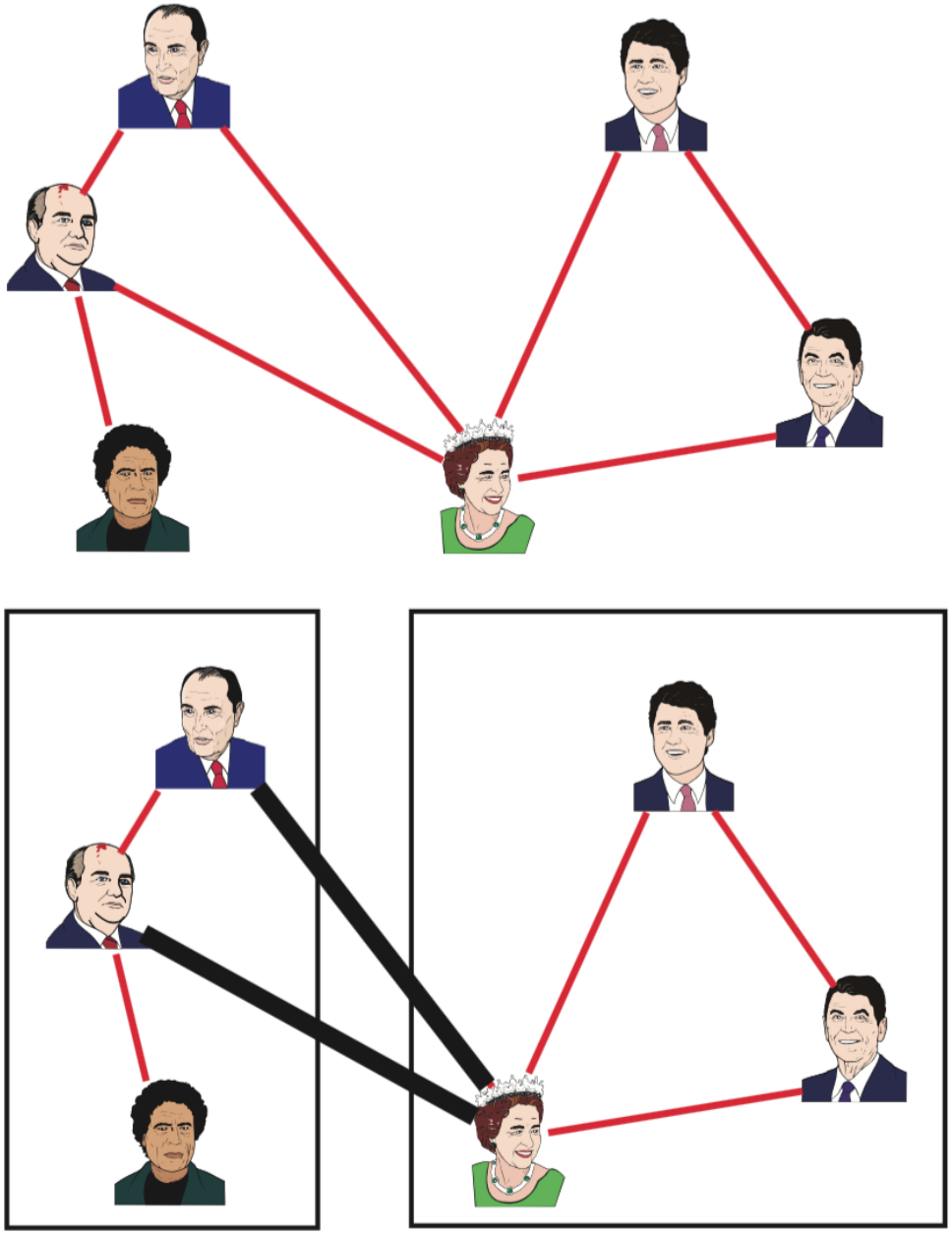
Graph partitioning problem

Problema da partição de grafos

Seis amigos estão a decidir como se dividir para formar duas equipas de mini-futebol. Claro, para ser justo, cada equipa deve ter o mesmo número de pessoas — três, neste caso. No entanto, ter bons amigos em equipas separadas deve ser evitado tanto quanto possível. Como se deve dividir o grupo de pessoas em duas equipes?

- Grafo:
  - vértices $\rightarrow$ pessoas
  - aresta entre duas pessoas $\rightarrow$ bons amigos





- Pares de amigos pertencendo a equipas diferentes $\rightarrow$ linha carregada

Problema da partição de grafos

Dado um grafo não dirigido $G = (V, E)$ com um número par de vértices $n = |V|$, divida V em dois subconjuntos L e R com o mesmo número de vértices, satisfazendo a condições abaixo, de modo a minimizar o número de arestas entre L e R (NP-hard)

- $L \cap R = \emptyset \rightarrow$ *interseção nula*
- $L \cup R = V \rightarrow$ *união é o conjunto completo de vértices*
- $|L| = |R| = n/2$

- Aplicações:
  - VLSI (very-large-scale integration) design
  - imunização de redes: escolher um conjunto de nós a imunizar para que a rede seja infetada o mínimo possível por uma epidemia
    - ...
- Subconjuntos com a mesma cardinalidade
  - uniform partition or equipartition
  - (cardinalidade de conjunto, $|V|$: número de elementos que contém)
- Arestas através de L e R *(edges across L and R)*:
  - $\{i, j\}$ tais que ou $i \in L$ e $j \in R$, ou $i \in R$ e $j \in L$

Formulação como um problema de otimização inteira

- Variáveis (binárias): para cada vértice $i \in V$:
  - $x_i = 1$ se i estiver incluído no subconjunto L
  - $x_i = 0$ se i estiver incluído no subconjunto R
- Vértices igualmente divididos $\rightarrow$ soma de x_i deve ser igual a $n/2$
- Quando uma aresta $\{i, j\}$ está *através de* L e R: ou
  - $x_i(1 - x_j) = 1$, ou
  - $(1 - x_i)x_j = 1$

$$
\begin{aligned}
\text{minimize} \quad & \sum_{\{i,j\} \in E} (x_i(1 - x_j) + (1 - x_i)x_j) \\
\text{subject to} \quad & \sum_{i \in V} x_i = n/2 \\
& x_i \in \{0, 1\} \qquad \forall i \in V
\end{aligned}
$$

Notas

- modelo anterior: expressões quadráticas no objetivo
- termos quadráticos são não convexos
- muitos *solvers* não suportam problemas de minimização cuja função objetivo não é convexa
- mesmo que alguns forneçam suporte para esses casos, geralmente são muito mais eficientes em problemas lineares
- se pudermos encontrar uma formulação linear equivalente, é aconselhável usá-la

Reformulação

- Variáveis (binárias) y_{ij} modelam o caso em que as arestas são incidentes em diferentes subconjuntos:
  - $y_{ij} = 1$ se as extremidades da aresta $\{i, j\}$ estiverem através de L e R
  - $y_{ij} = 0$ caso contrário
- $x_i, i \in V \rightarrow$ como na formulação anterior

$$
\begin{aligned}
\text{minimize} \quad & \sum_{\{i,j\} \in E} y_{ij} \\
\text{subject to} \quad & \sum_{i \in V} x_i = n/2 \\
& x_i - x_j \leq y_{ij} && \forall \{i, j\} \in E \\
& x_j - x_i \leq y_{ij} && \forall \{i, j\} \in E \\
& x_i \in \{0, 1\} && \forall i \in V \\
& y_{ij} \in \{0, 1\} && \forall \{i, j\} \in E
\end{aligned}
$$

Notas

- Objetivo: minimizar a soma das variáveis y_{ij}
  - valor tenderá a ser zero, mas as restrições forçam algumas a serem um
- Primeira restrição: equipartição do conjunto de vértices
- Segunda restrição: se $i \in L$ e $j \notin L$ então $y_{ij} = 1$
  - *i.e.*, aresta $\{i, j\}$ está através dos subconjuntos L e R
- Terceira restrição: se $j \in L$ e $i \notin L$, então $y_{ij} = 1$.

$$x_i - x_j \leq y_{ij} \qquad \forall \{i, j\} \in E$$

$$x_j - x_i \leq y_{ij} \qquad \forall \{i, j\} \in E$$

Modelo AMPL

```
set V;
set E within {V,V};
param n := card(V);

var x {V} binary;
var y {E} binary;

minimize z: sum {(i,j) in E} y[i,j];

subject to
Equipart: sum {i in V} x[i] = n/2;
AcrossLR {(i,j) in E}: x[i] - x[j] <= y[i,j];
AcrossRL {(i,j) in E}: x[j] - x[i] <= y[i,j];
```

"Toy instance"

```
data;
set V := 1 2 3 4 5 6;
set E :=
    (1,2)
    (2,3)
    (2,6)
    (3,6)
    (4,5)
    (5,6)
    (4,6)
;
```

Grafos maiores

Para criar um grafo aleatório

```
import random
def make_data(n,prob):
    """make_data: prepare data for a random graph
    Parameters:
        - n: number of vertices
        - prob: probability of existence of an edge, for each pair of vertices
    Returns a tuple with a list of vertices and a list edges.
       """
    V = list(range(1,n+1))
    E = [(i,j) for i in V for j in V if i < j and random.random() < prob]
    return V,E
```

Usando AMPL + Python

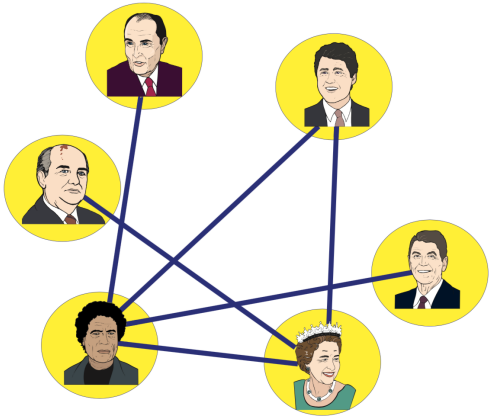
```
V, E = make_data(10, .5)

from amplpy import AMPL, Environment, DataFrame
ampl = AMPL()
ampl.option['solver'] = 'gurobi'
ampl.read("gpp.mod")
ampl.set['V'] = V
ampl.set['E'] = E

ampl.solve()
z = ampl.obj['z']
print("Optimum:", z.value())

x = ampl.var['x']
L = [i for i in V if x[i].value() >.5]
R = [i for i in V if x[i].value() <.5]
print("L =", L)
print("R =", R)

y = ampl.var['y']
across = [(i,j) for (i,j) in E if y[i,j].value() > .5]
print("Across edges:", across)
```

Problemas do conjunto estável máximo e da clique máxima

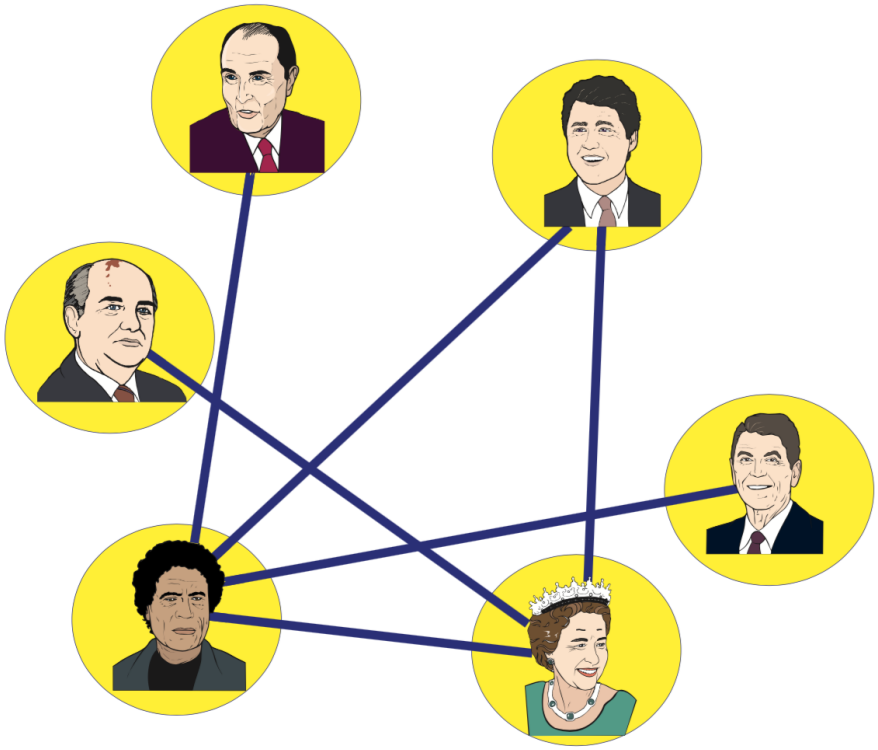
- Maximum stable set problem
- Maximum clique problem

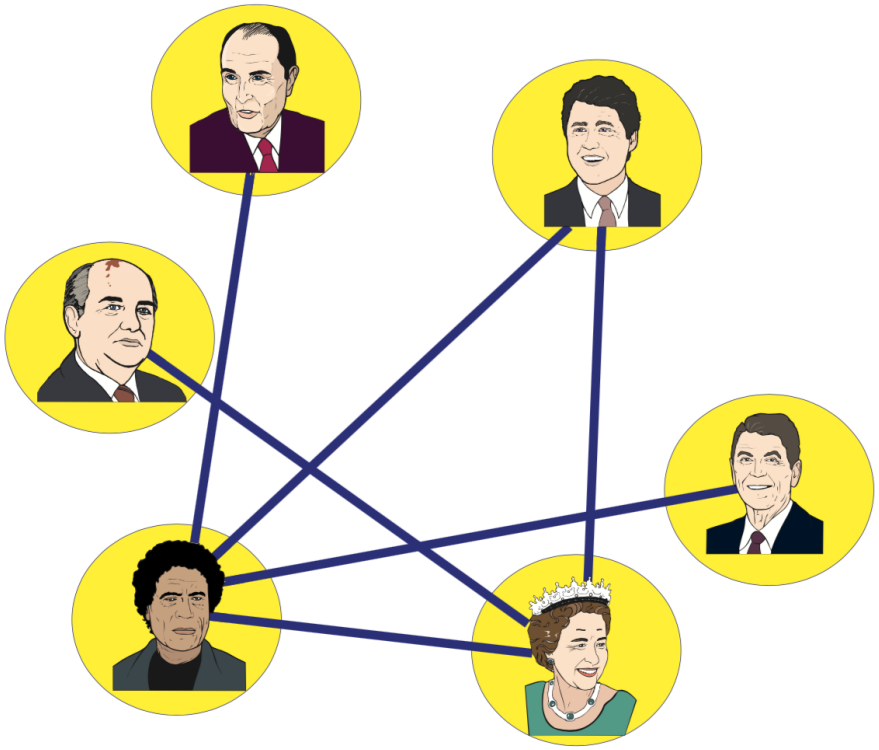
Problema do conjunto estável máximo

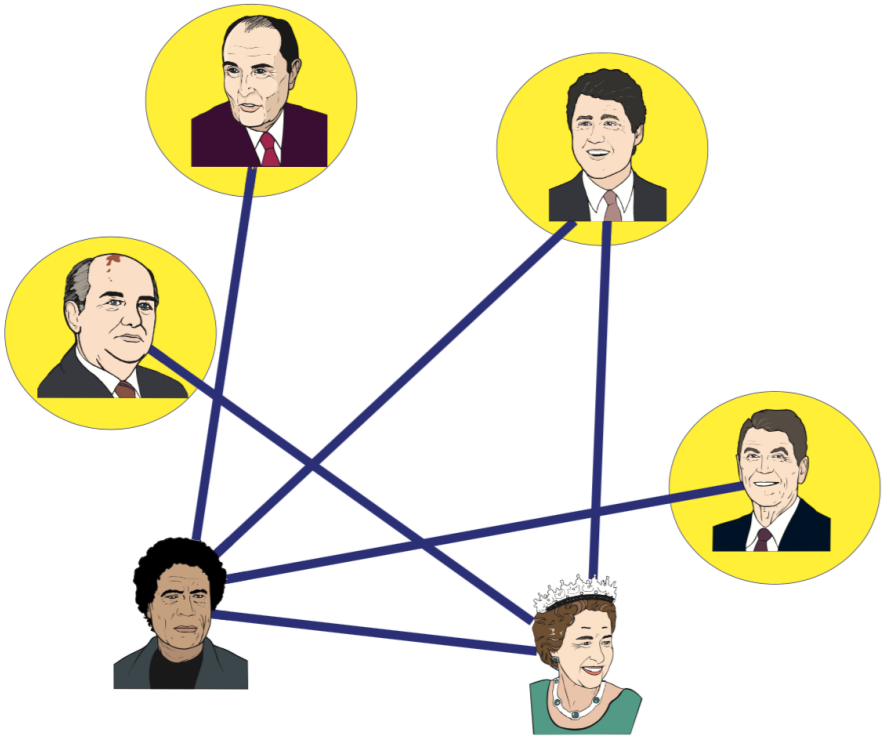
Está a escolher, entre um grupo de seis amigos, com quem fazer um piquenique. No entanto, as pessoas ligadas por uma aresta na figura mantêm relações muito hostis umas com as outras; portanto, se ambas forem ao piquenique, a festa será estragada. Para ter o maior número possível de amigos no piquenique, quem deve ser convidado?



- Grafo:
  - vértices $\rightarrow$ pessoas
  - aresta entre duas pessoas $\rightarrow$ más relações







- vértices com círculos (direita) $\rightarrow$ grupo (máximo) de pessoas que podem ser convidadas para o piquenique sem estragar a festa

Problema do conjunto estável máximo

Dado um grafo não dirigido $G = (V, E)$, diz-se que $S \subseteq V$ é um conjunto estável quando não há nenhuma aresta entre pares de vértices de S. O problema é encontrar um conjunto estável S com cardinalidade máxima.

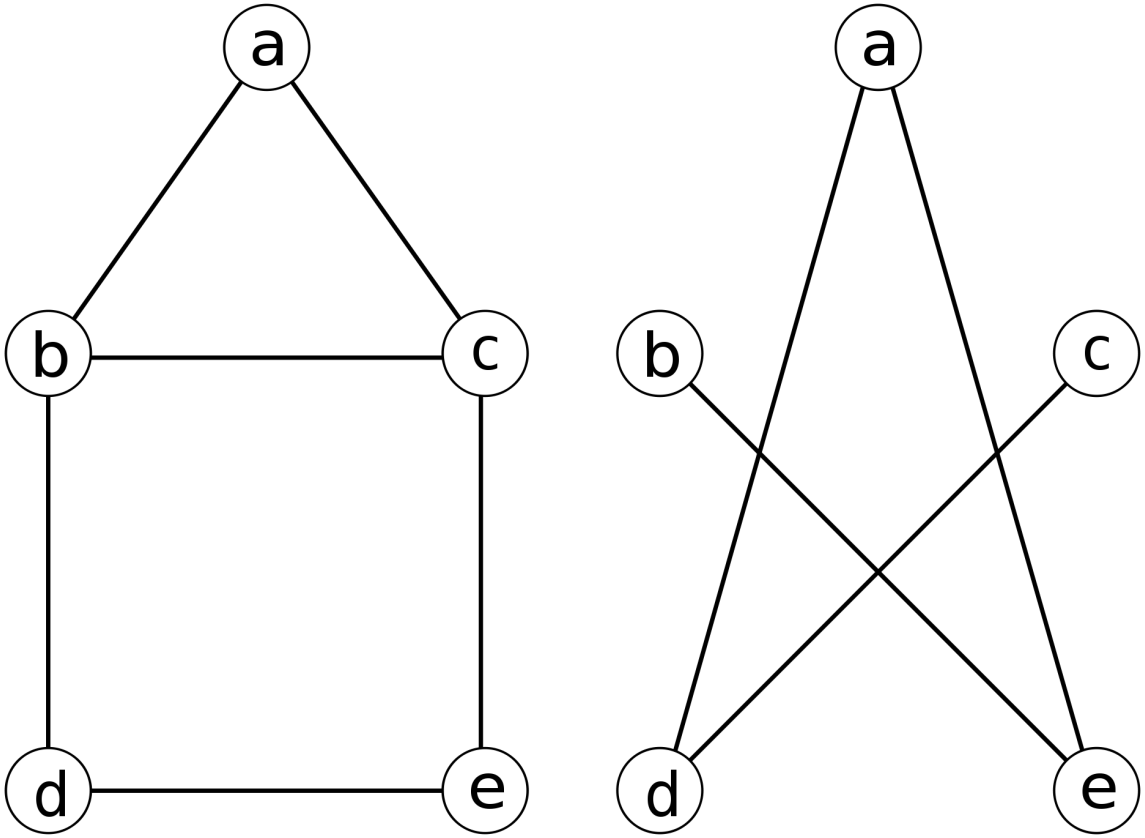
Problema da clique máxima

Dado um grafo não dirigido $G = (V, E)$, um subconjunto $C \subseteq V$ é chamado uma clique quando o subgrafo induzido por C é completo. O problema é encontrar uma clique C que maximize a cardinalidade $|C|$.

- Grafo completo: há uma aresta entre qualquer par de vértices
- Clique: subgrafo induzido por um subconjunto de vértices contendo todas as arestas (do grafo original) com ambas as extremidades nesse subconjunto
- Aplicações: teoria da codificação, confiabilidade, genética, arqueologia e design VLSI, ...

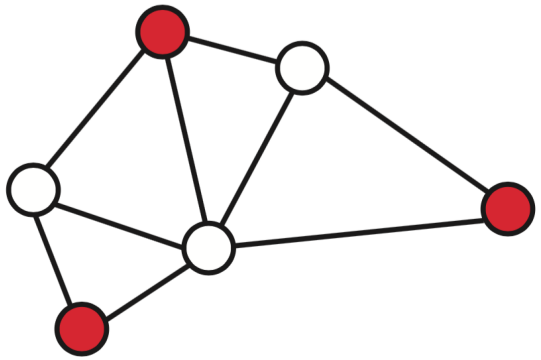
Grafo complementar

- Dado um grafo, o seu grafo complementar inverte as arestas:
  - tem arestas apenas entre pares de vértices para os quais não há aresta no grafo original

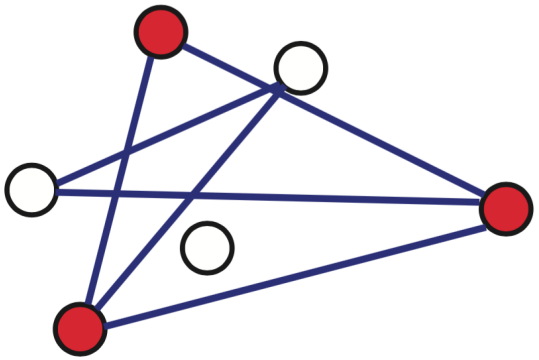


Conjunto estável máximo e clique máxima

- Problema da clique máxima: é equivalente ao conjunto estável máximo:
  - pode ser convertido através de uma transformação simples, e
  - a solução é a mesma



(a)



(b)

- Vértices vermelhos são:
  - conjunto estável máximo no grafo original
  - clique máxima no grafo complementar

Conjunto estável máximo: formulação em otimização inteira

- Variáveis (binárias): para cada vértice $i \in V$:
  - $x_i = 1$ se i estiver incluído no conjunto estável
  - $x_i = 0$ caso contrário
- Modelo:

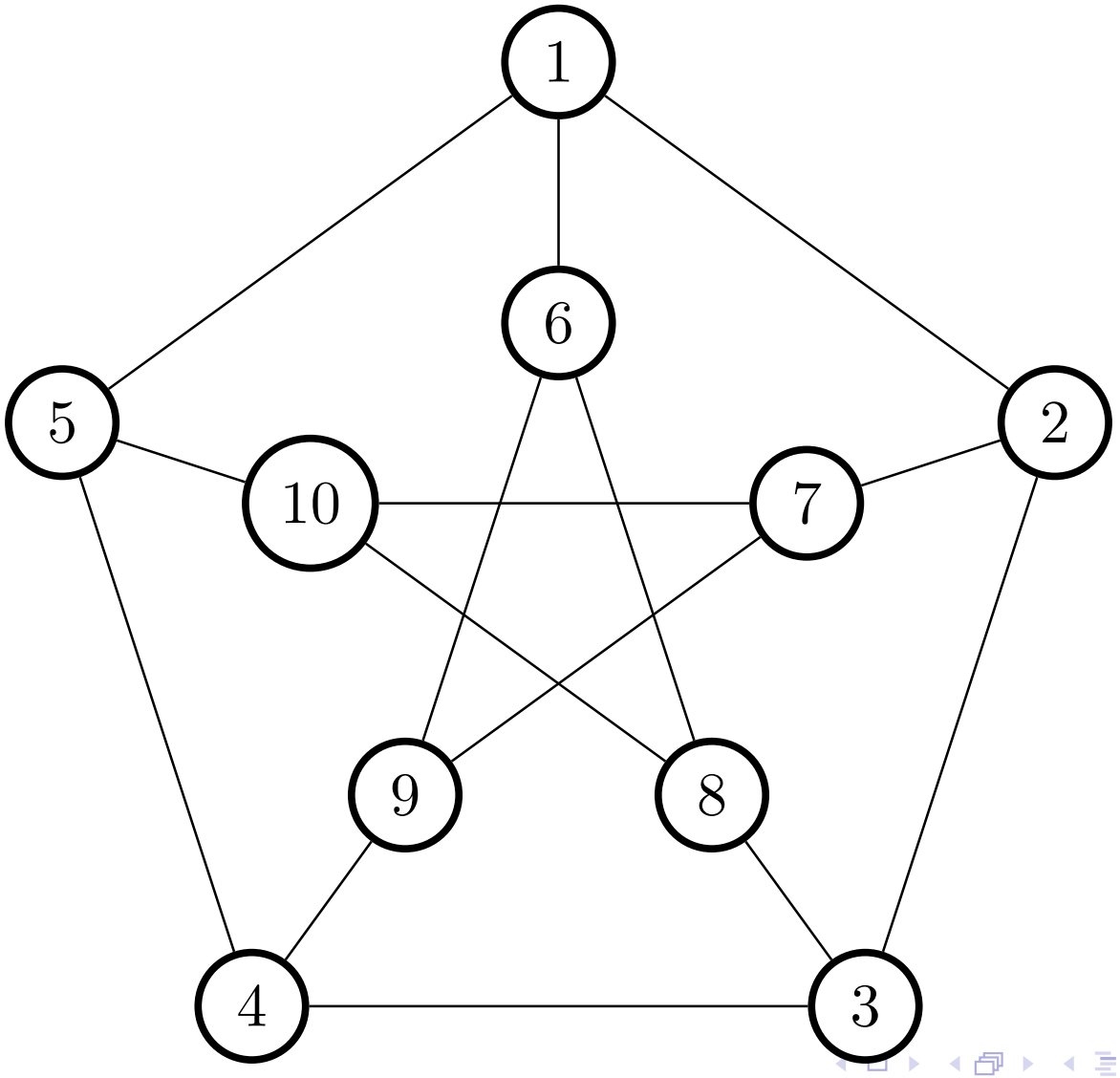
$$
\begin{aligned}
\text{maximize} \quad & \sum_{i \in V} x_i \\
\text{subject to} \quad & x_i + x_j \leq 1 && \forall \{i,j\} \in E \\
& x_i \in \{0,1\} && \forall i \in V
\end{aligned}
$$

Model AMPL

```
set V;
set E within {V,V};

var x {V} binary;

maximize z: sum {i in V} x[i];

subject to
Edge {(i,j) in E}: x[i] + x[j] <= 1;
```

Challenge:

Encontre a clique máxima no grafo de Petersen:



Aula de hoje

- Problema da partição de grafos
- Problema do conjunto estável máximo
- Problema da clique máxima

Próximas aulas

- Problemas de deteminação de rotas