Data-driven Decision Making

Graph problems: partitioning, stable sets, cliques

João Pedro Pedroso

Kseniia Klimetnova

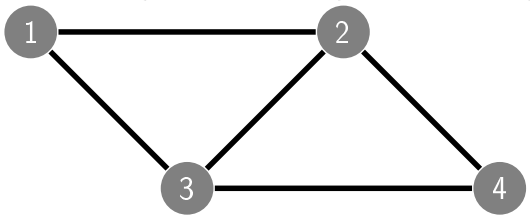
2025/2026

Last class

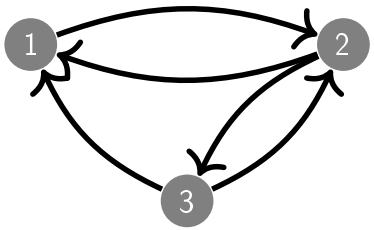
- Location problems:
  - k-Median
  - k-Center
  - k-Cover
- Graph problems
  - coloring
- Models:
  - how to formulate
  - how to improve the formulation

Graphs

- Undirected graphs: lines connecting two vertices have no implied direction
  - lines with no direction $\rightarrow$ edges
  - example: $G = (V, E)$
    $V = \{1, 2, 3, 4\}$, $E = \{\{1, 2\}, \{2, 3\}, \{2, 4\}, \{3, 1\}, \{3, 4\}\}$



- Directed graphs: direction of lines connecting two vertices is important
  - directed lines $\rightarrow$ arcs
  - example: $G = (V, A)$
    $V = \{1, 2, 3\}$, $A = \{(1, 2), (2, 1), (2, 3), (3, 2), (3, 1)\}$

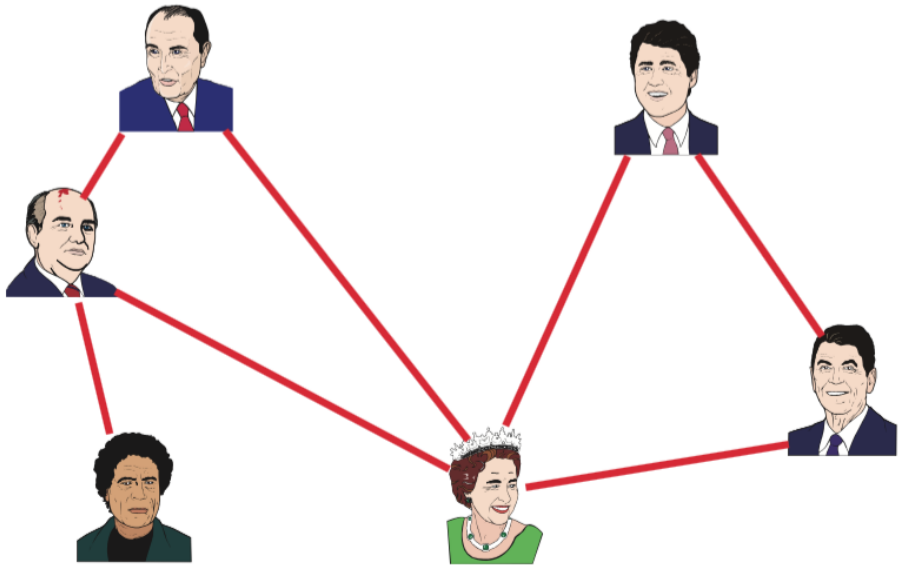


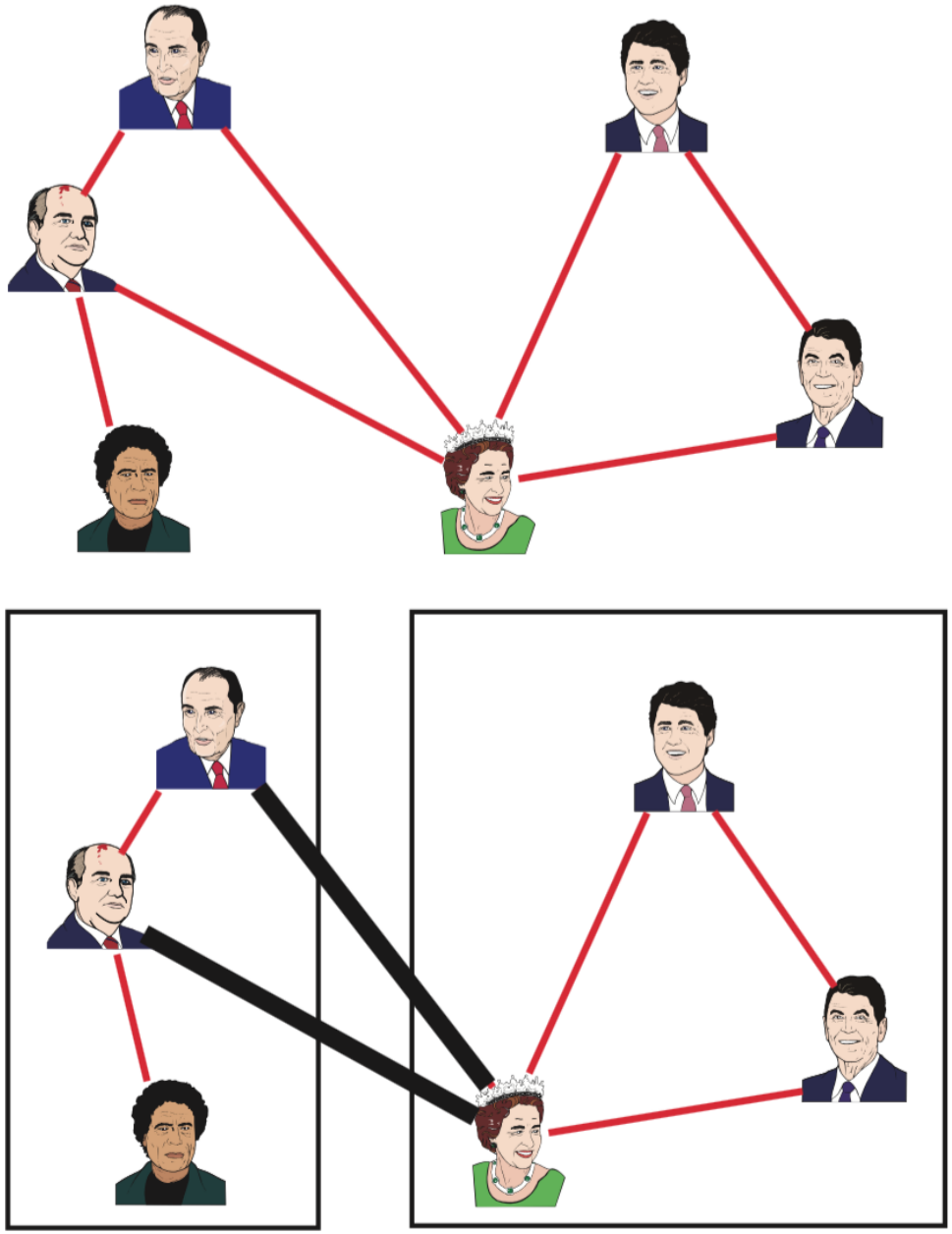
Graph partitioning problem

Graph partitioning problem

Six friends are deciding how to split for forming two teams of mini-soccer. Of course, in order to be fair, each team must have the same number of persons — three, in this case. However, having good friends in separate teams should be avoided as much as possible. So, how should the group of persons be divided into two teams?

- Graph:
  - nodes $\rightarrow$ persons
  - edge between two persons $\rightarrow$ good friends





- Pairs of good friends belonging to different teams $\rightarrow$ thick line

Graph partitioning problem

Given an undirected graph $G = (V, E)$ with an even number of vertices $n = |V|$, divide V into two subsets L and R with the same number of vertices, satisfying the conditions below, so as to minimize the number of edges across L and R *(NP-hard)*

- $L \cap R = \emptyset \rightarrow$ *no intersection*
- $L \cup R = V \rightarrow$ *union is the whole set of vertices*
- $|L| = |R| = n/2$

- Applications:
  - VLSI (very-large-scale integration) design
  - network immunization: pick out a set of nodes to immunize so that network is infected as little as possible by an epidemic
  - ...
- Subsets with same cardinality
  - uniform partition or equipartition
  - (cardinality of set $|V|$: its number of elements)
- Edges across L and R:
  - $\{i, j\}$ such that either $i \in L$ and $j \in R$, or $i \in R$ and $j \in L$

Formulation as an integer optimization problem

- (Binary) Variables: for each vertex $i \in V$:
  - $x_i = 1$ if i is included in subset L
  - $x_i = 0$ if i is included in subset R
- Vertices equally divided $\rightarrow$ sum x_i must be equal to $n/2$
- When an edge $\{i, j\}$ is across L and R: either
  - $x_i(1 - x_j) = 1$, or
  - $(1 - x_i)x_j = 1$

$$\text{minimize} \quad \sum_{\{i,j\} \in E} (x_i(1 - x_j) + (1 - x_i)x_j)$$

$$\text{subject to} \quad \sum_{i \in V} x_i = n/2$$

$$x_i \in \{0, 1\} \qquad \forall i \in V$$

Notes

- previous model: quadratic expressions in objective
- many solvers do not support minimization problems whose objective function is not convex
- quadratic terms are not convex
- even if some solvers provide support for these cases, they usually are much more efficient for solving linear problems
- if we can find an equivalent linear formulation, is it advisable to use it

Reformulation

- (Binary) variables y_{ij} model the case where edges are incident to different subsets:
  - $y_{ij} = 1$ if the endpoints of edge $\{i, j\}$ are across L and R
  - $y_{ij} = 0$ otherwise
- $x_i, i \in V \rightarrow$ as in previous formulation

$$\text{minimize} \quad \sum_{\{i,j\} \in E} y_{ij}$$

$$\text{subject to} \quad \sum_{i \in V} x_i = n/2$$

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

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

$$x_i \in \{0, 1\} \qquad \forall i \in V$$

$$y_{ij} \in \{0, 1\} \qquad \forall \{i, j\} \in E$$

Remarks

- Objective: minimize sum of variables y_{ij}
  - their value will tend to be zero, but constraints force some of them to be one
- First constraint: equal division of the set of vertices
- Second constraint: if $i \in L$ and $j \notin L$ then $y_{ij} = 1$
  - *i.e.*, edge $\{i, j\}$ is across subsets L and R
- Third constraint: if $j \in L$ and $i \notin L$, then $y_{ij} = 1$.

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

AMPL model

```
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)
;
```

Larger graphs

Create a random graph

```
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
```

AMPL + Python usage

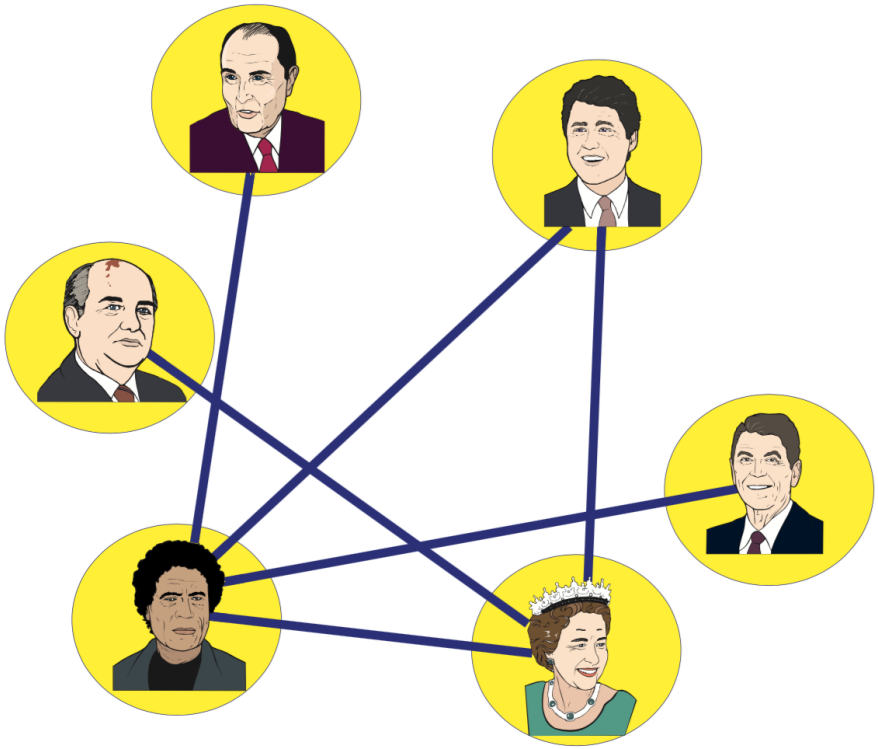
```
V, E = make_data(10, .5)

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)
```

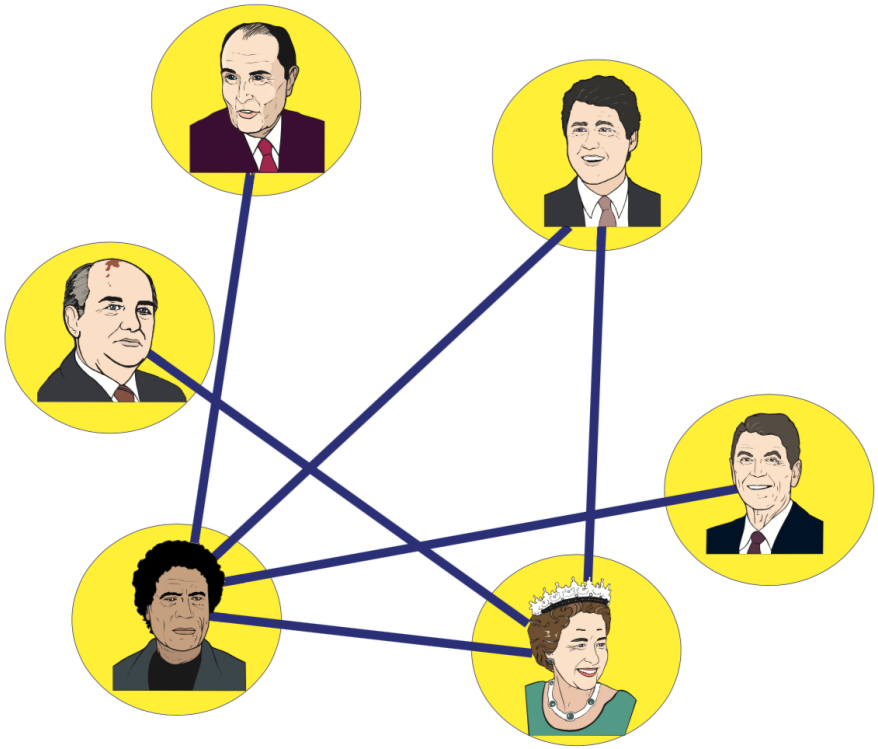
Maximum stable set and maximum clique problems

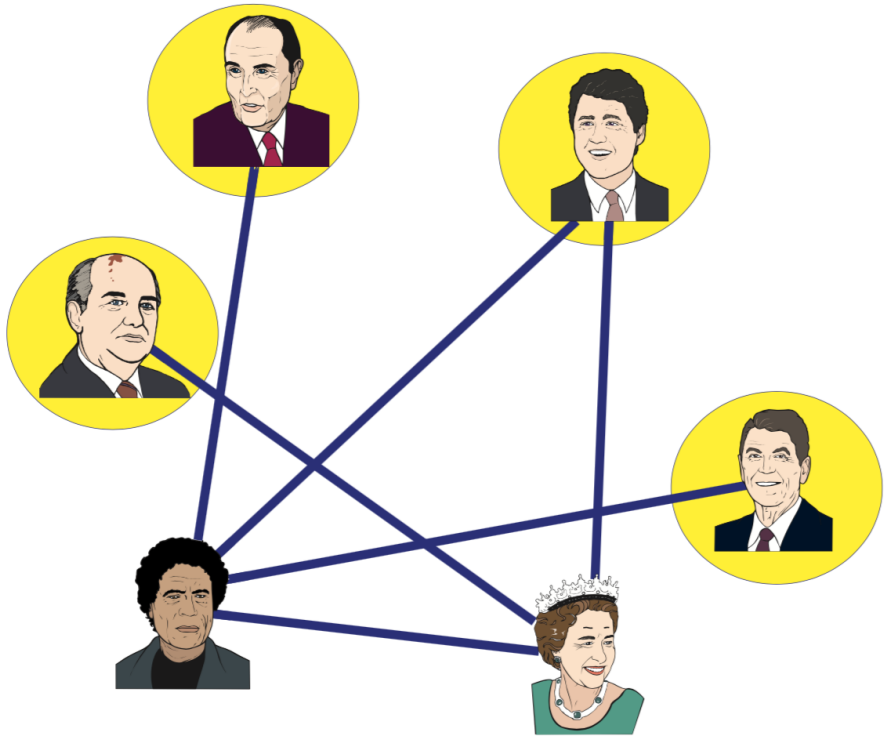
Maximum stable set problem

You are choosing, from a group of six friends, with whom to go for a picnic. However, persons linked with an edge in the figure are on very unfriendly terms with each other, so if both of them go to the picnic, it will be spoiled. To have as many friends as possible in the picnic, who should be invited?

- Graph:
  - nodes $\rightarrow$ persons
  - edge between two persons $\rightarrow$ bad terms







- nodes encircled $\rightarrow$ persons who can all be at the picnic without spoiling it

Maximum stable set problem

Given an undirected graph $G = (V, E)$, a subset $S \subseteq V$ is called a stable set when there isn't any edge among vertices of S. The problem is to find a stable set S such that its cardinality is maximum.

- Cardinality of a set $|S|$: the number of elements it contains

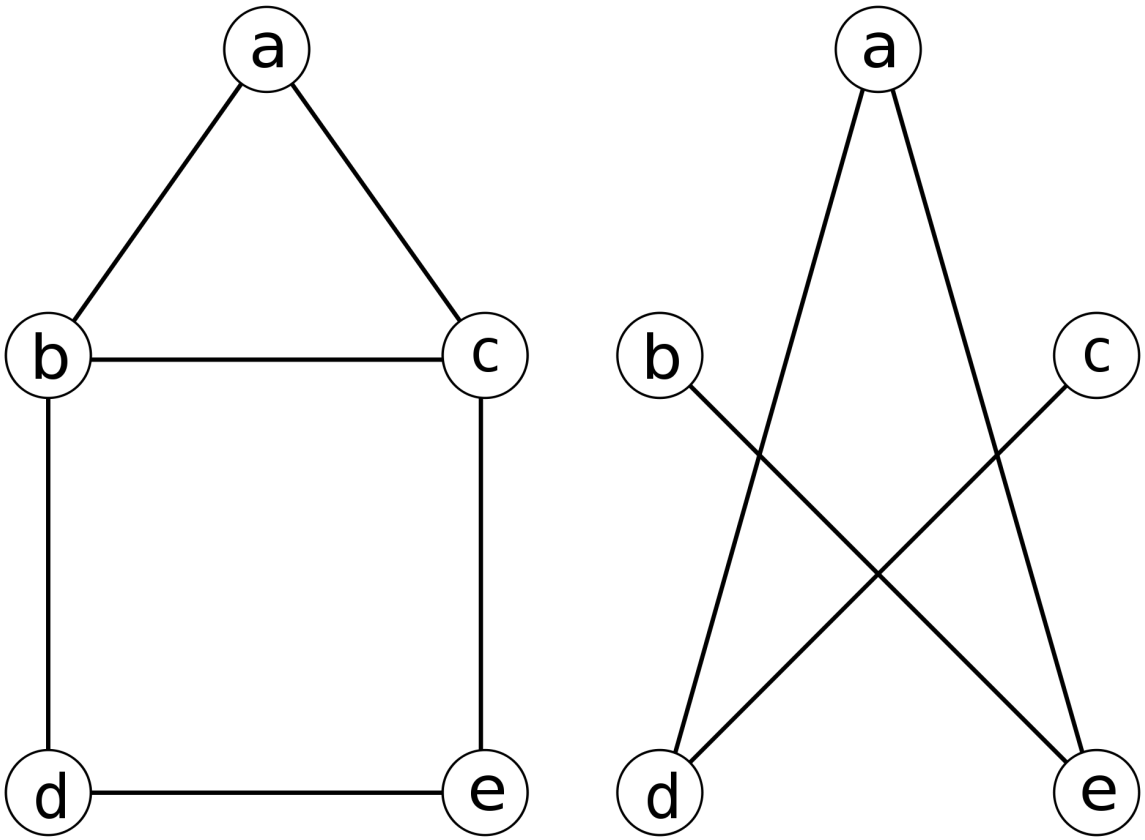
Maximum clique problem

Given an undirected graph $G = (V, E)$, a subset $C \subseteq V$ is called a clique when the subgraph induced by C is complete. The problem is to find a clique C which maximizes cardinality $|C|$.

- Complete graph: there is edge connecting all pairs of vertices
- Clique: subgraph induced by a subset of vertices containing all the edges of the original graph with both ends in that subset
- Applications: coding theory, reliability, genetics, archeology and VLSI design, ...

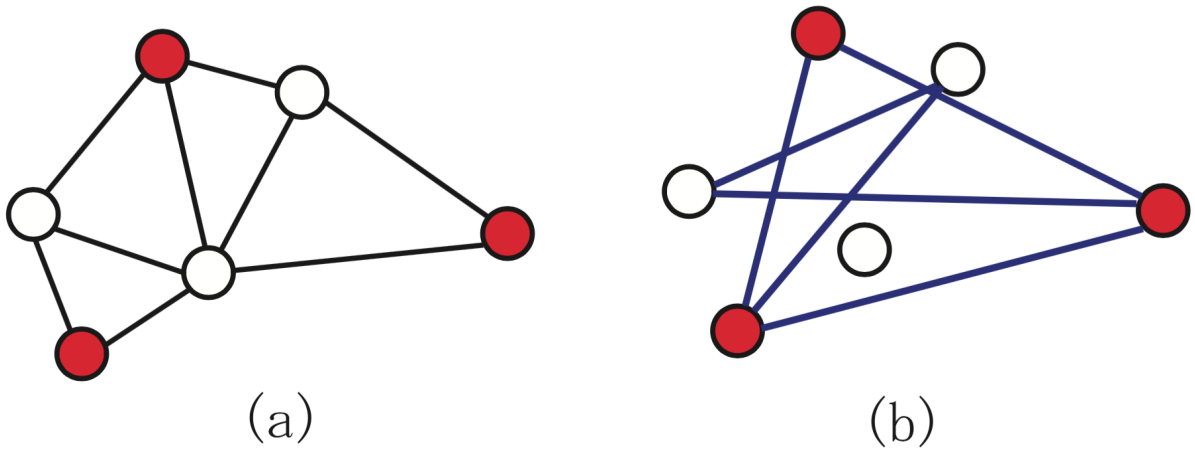
Complementary graph

- Given a graph, its complementary graph inverts the edges:
  - contains edges only between pairs of vertices for which there is no edge in the original graph



Maximum stable set and Maximum clique

- Maximum clique problem: equivalent to maximum stable set:
  - can be converted through a simple transformation, and
  - solution is the same



- Red vertices are both:
  - Maximum stable set on the original graph
  - Maximum clique on the complementary graph

Maximum stable set: formulation in integer optimization

- (Binary) Variables: for each vertex $i \in V$:
  - $x_i = 1$ if i is included in the stable set
  - $x_i = 0$ otherwise
- Model:

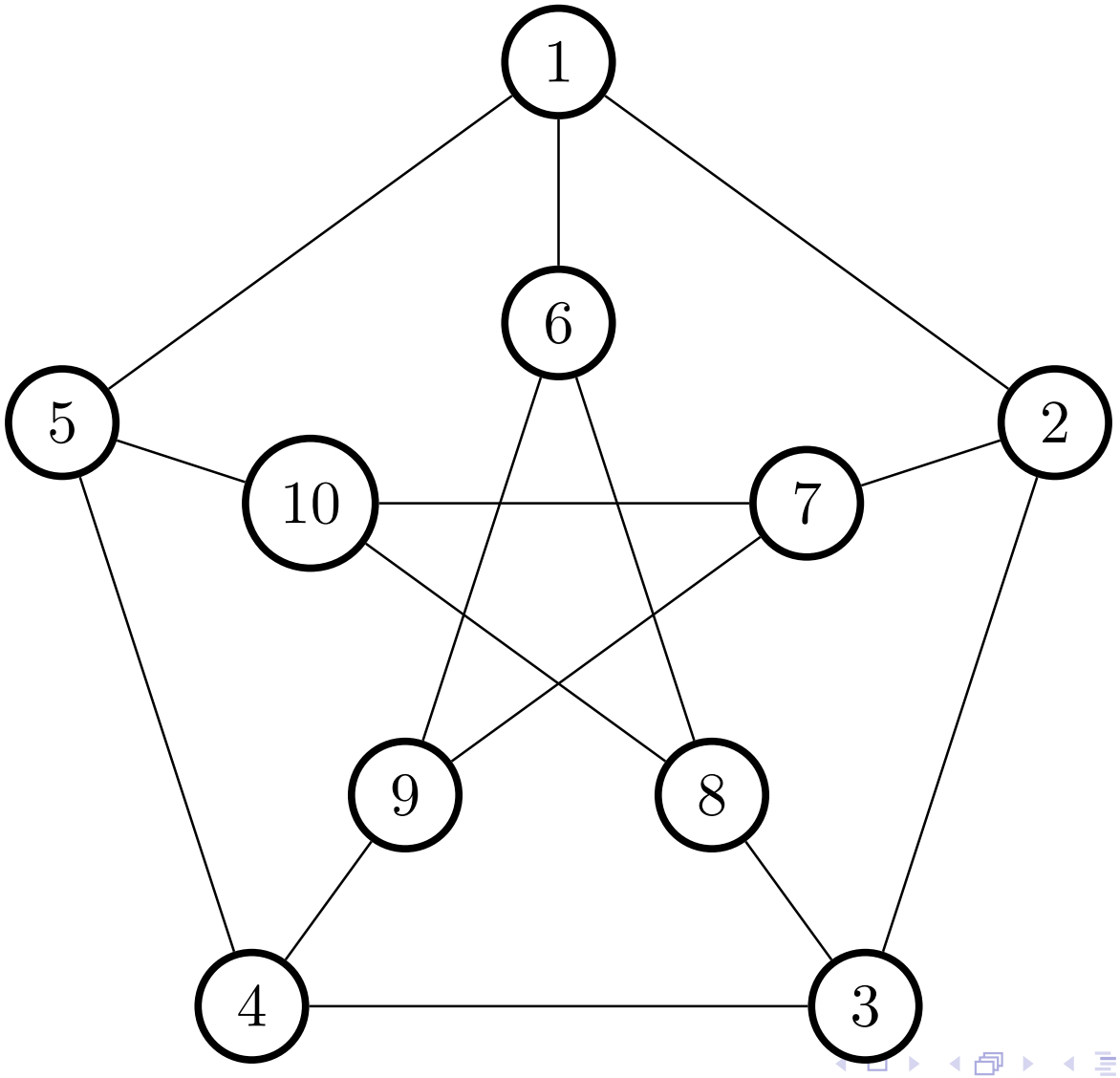
$$\begin{aligned} \text{maximize} \quad & \sum_{i \in V} x_i & \\ \text{subject to} \quad & x_i + x_j \leq 1 & \forall \{i,j\} \in E \\ & x_i \in \{0,1\} & \forall i \in V \end{aligned}$$

AMPL model

```
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:

Find the maximum clique for the Petersen graph:



Bonus: Routing problems: the (Asymmetric) TSP

Miller-Tucker-Zemlin (potential) formulation

- Consider an asymmetric traveling salesman problem
- We will now see a formulation with a number of constraints of polynomial order

  Given:
  - *directed graph* $G = (V, A)$, *where*
    - $V \rightarrow$ *set of vertices*
    - $A \rightarrow$ *a set of (directed) arcs*
  - *function* $c : A \rightarrow \mathbb{R}$ *associating a distance to each arc*

  Find a tour which passes exactly once in each city and minimizes the total distance

Miller-Tucker-Zemlin formulation: notation

- Variables:
  - (binary) $x_{ij} = 1$ if visiting vertex j next to i, 0 otherwise
  - (real) $u_i \rightarrow$ represents the visiting order of vertex i
- Interpretation: u_1 is the potential at each vertex
  - starting point: $u_1 = 0$
- When visiting j next to vertex i:
  - force the potential of j to be $u_j = u_i + 1$
  - do this for any vertex except 1
  - possible values for u_i are $1, 2, \ldots ..., n - 1$
    - $n \rightarrow$ number or vertices

MTZ formulation for the asymmetric traveling salesman problem

$$\text{minimize} \quad \sum_{i \neq j} c_{ij} x_{ij}$$

$$\text{subject to} \quad \sum_{j:j \neq i} x_{ij} = 1 \qquad i = 1, \ldots, n$$

$$\sum_{j:j \neq i} x_{ji} = 1 \qquad i = 1, \ldots, n$$

$$u_i + 1 - (n-1)(1 - x_{ij}) \leq u_j \qquad i = 1, \ldots, n, \quad j = 2, \ldots, n : i \neq j$$

$$0 \leq u_i \leq n - 1 \qquad i = 1, \ldots, n$$

$$x_{ij} \in \{0, 1\} \qquad \forall i \neq j$$

Remarks

- First and second constraints: assignment constraints $\rightarrow$ ensure that each vertex is incident to one outgoing arc and one incoming arc.
- Third constraint: Miller-Tucker-Zemlin or potential constraint $\rightarrow$ define the order in which each vertex i is visited on a tour.
  - $u_i \rightarrow$ potential at vertex i
  - forcing $u_j = u_i + 1$ only when $x_{ij} = 1$ in
    $$u_i + 1 - (n-1)(1 - x_{ij}) \leq u_j$$
    - coefficient $n-1$ of term $(1 - x_{ij}) \rightarrow$ *"Big M" constraint*
- Fourth constraint: upper and lower bounds of the potential.
- This formulation is much weaker than subtour elimination constraints

AMPL model

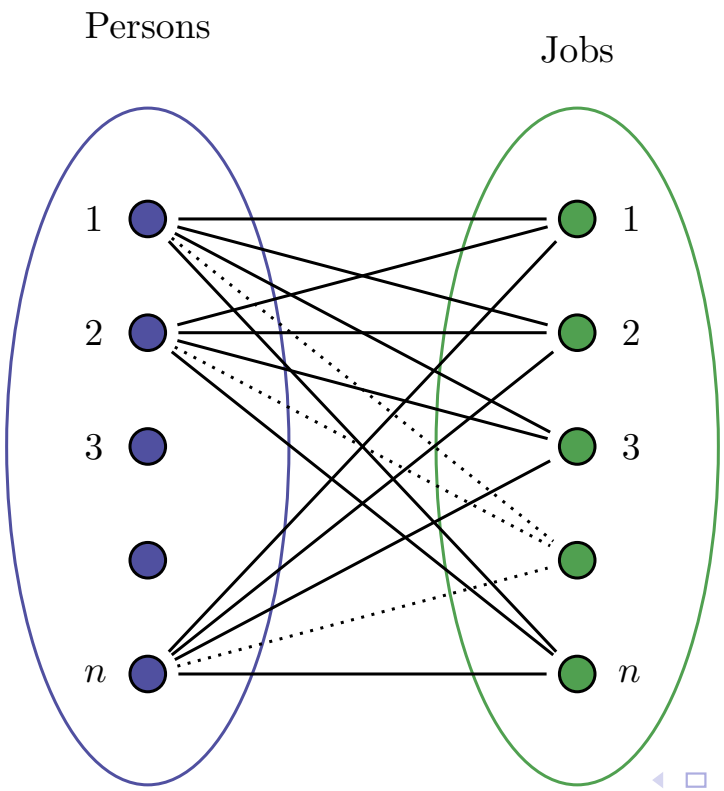
```
param n;
set V := {1..n};
param c {i in V, j in V};

var x {i in V, j in V: i != j} binary;
var u {V} >= 0, <= n-1;

minimize z: sum {i in V, j in V: i != j} c[i,j] * x[i,j];

subject to
OutDegree {i in V}:
    sum {j in V: j != i} x[i,j] = 1;
InDegree {i in V}:
    sum {j in V: j != i} x[j,i] = 1;
Potential {i in 1..n, j in 2..n : i != j}:
    u[i] + 1 - (n-1) * (1-x[i,j]) <= u[j];
```

Preliminaries for next week:

Assignment problem

Assignment problem

- There are n people available to carry out n jobs
- Each person is assigned to carry out exactly one job
- Some persons are better suited to particular jobs than others
  - cost c_{ij} if person i is assigned to job j
- Problem: find a minimum cost / maximum reward assignment



Assignment problem

Formulation:

- $x_{ij} = 1$ if person i does job j, 0 otherwise
- all persons are assigned to a job
- all jobs are assigned to a person

$$\text{minimize} \quad \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}$$

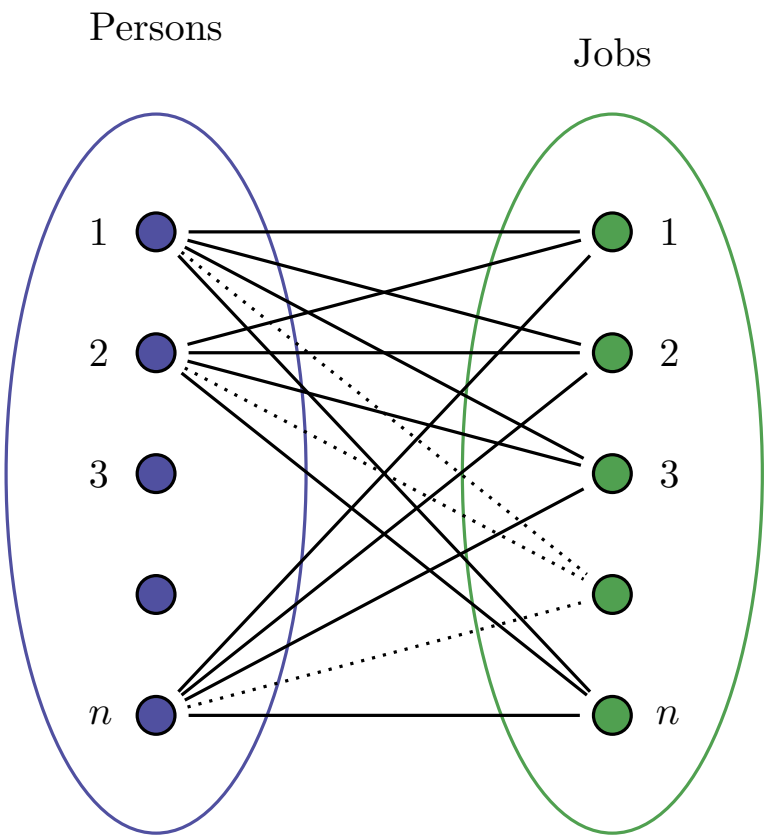
$$\text{subject to} \quad \sum_{j=1}^{n} x_{ij} = 1 \qquad i = 1, \ldots, n$$

$$\sum_{i=1}^{n} x_{ij} = 1 \qquad j = 1, \ldots, n$$

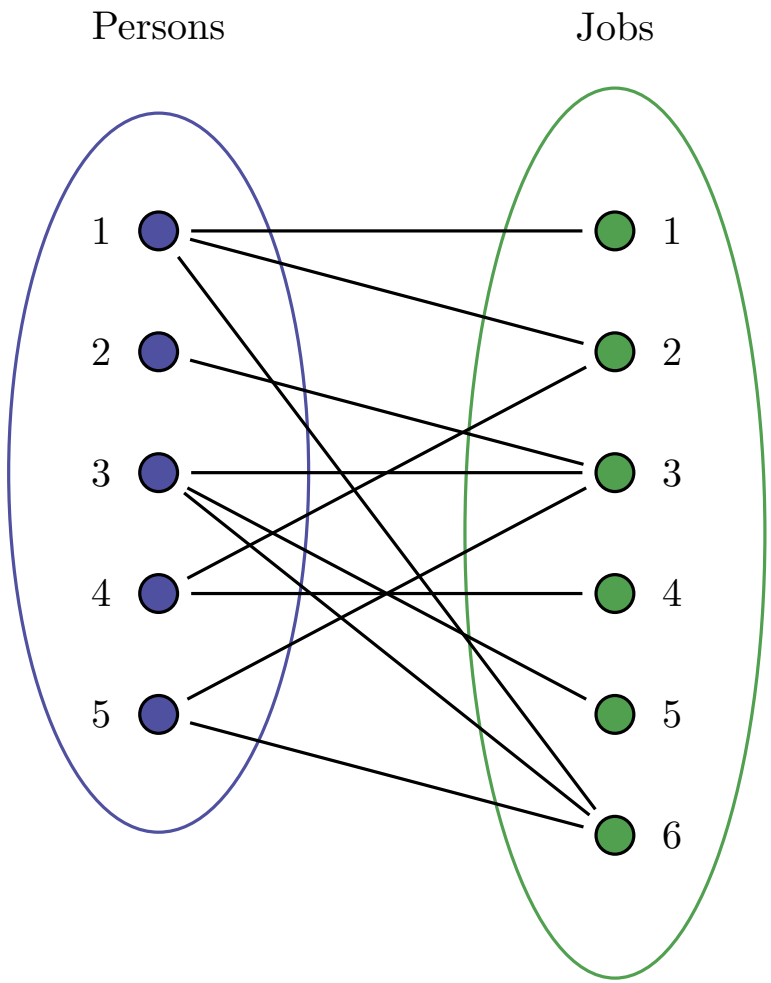
$$x_{ij} \in \{0, 1\} \qquad i = 1, \ldots, n, j = 1, \ldots, n$$

Bipartite graphs

Complete

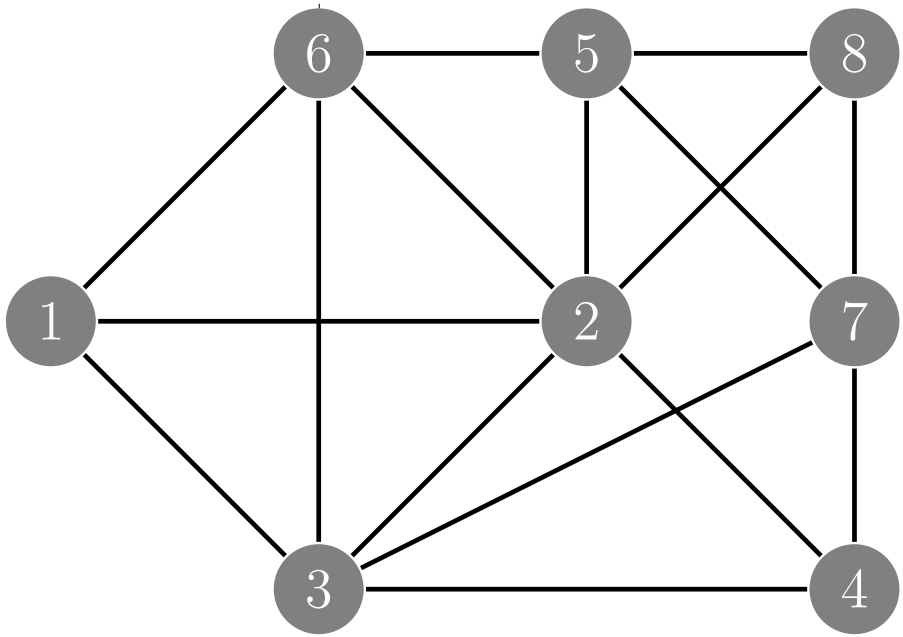


Arbitrary



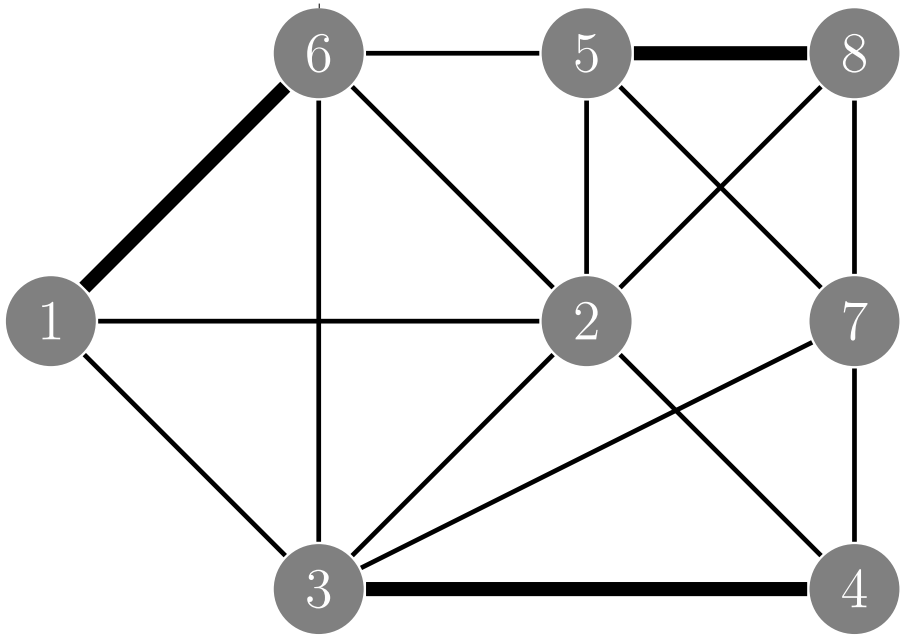
Matching

- Given a graph $G = (V, E)$ a matching is a set of disjoint edges



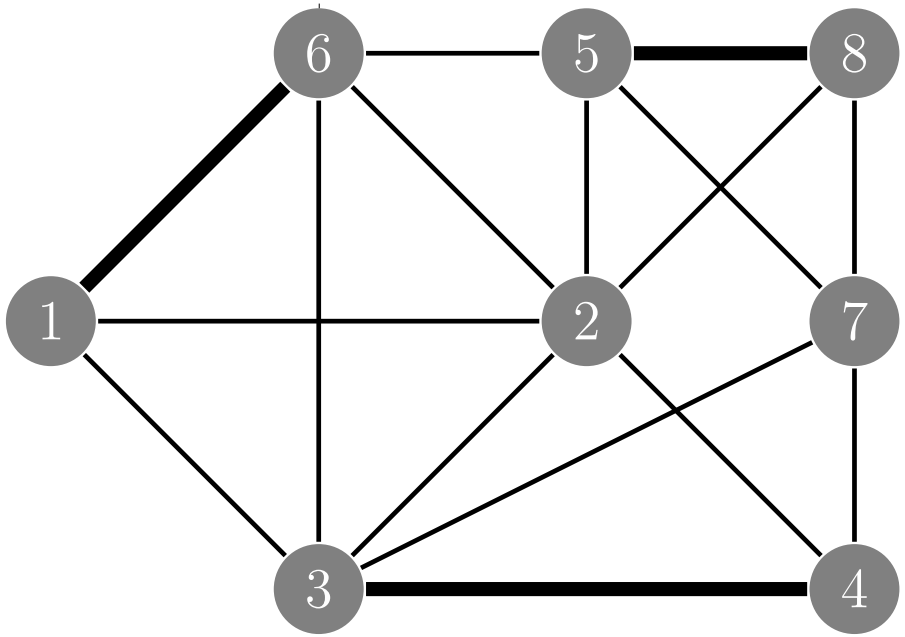
Matching

- Given a graph $G = (V, E)$ a matching is a set of disjoint edges



Matching

- Given a graph $G = (V, E)$ a matching is a set of disjoint edges



- Maximum cardinality matching:
  - select the maximum number of edges that are a matching
- Maximum weight matching:
  - select edges that are a matching with maximum total weight
- Perfect matching:
  - for a graph with $2k$ nodes: matching with k edges

Maximum weight matching

- Simple, polynomial algorithms for the bipartite case
- Not so simple, still polynomial for the general case

Maximum cardinality matching: all weights c_{ij} are 1

Maximum weight matching

Given:

- graph $G = (V, E)$
- weight c_{ij} for all $ij \in E$

Formulation:

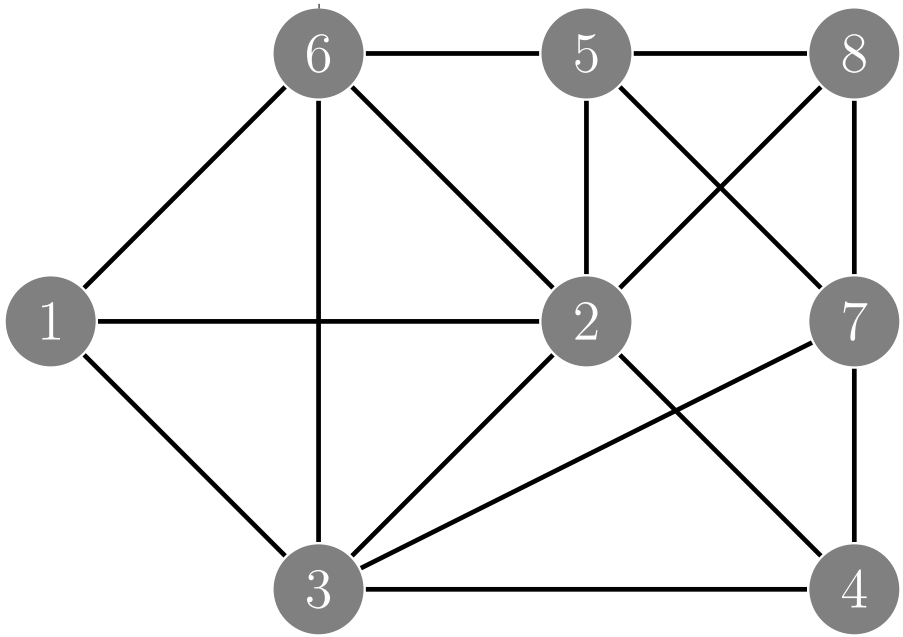
- $x_{ij} = 1$ if edge $\{i, j\}$ is in the matching, 0 otherwise $\quad \forall \{i, j\} \in E$

$$\begin{aligned} \text{maximize} \quad & \sum_{ij \in E} c_{ij} x_{ij} \\ \text{subject to} \quad & \sum_{j : ij \in E} x_{ij} \leq 1 && \forall i \in V \\ & x_{ij} \in \{0, 1\} && \forall ij \in E \end{aligned}$$

Note: I'm abusing notation, and representing an edge $\{i, j\}$ simply by ij

Example: maximum cardinality matching

Consider the graph



Find the maximum cardinality matching

Model

Recall:

$$\begin{aligned} \text{maximize} \quad & \sum_{ij \in E} c_{ij} x_{ij} & \\ \text{subject to} \quad & \sum_{j:ij \in E} x_{ij} \leq 1 & \forall i \in V \\ & x_{ij} \in \{0, 1\} & \forall ij \in E \end{aligned}$$

Model

Recall:

$$\text{maximize} \quad \sum_{ij \in E} c_{ij} x_{ij}$$

$$\text{subject to} \quad \sum_{j:ij \in E} x_{ij} \leq 1 \qquad \forall i \in V$$

$$x_{ij} \in \{0, 1\} \qquad \forall ij \in E$$

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

var x {E} binary;

maximize z: sum {(i,j) in E} x[i,j];

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

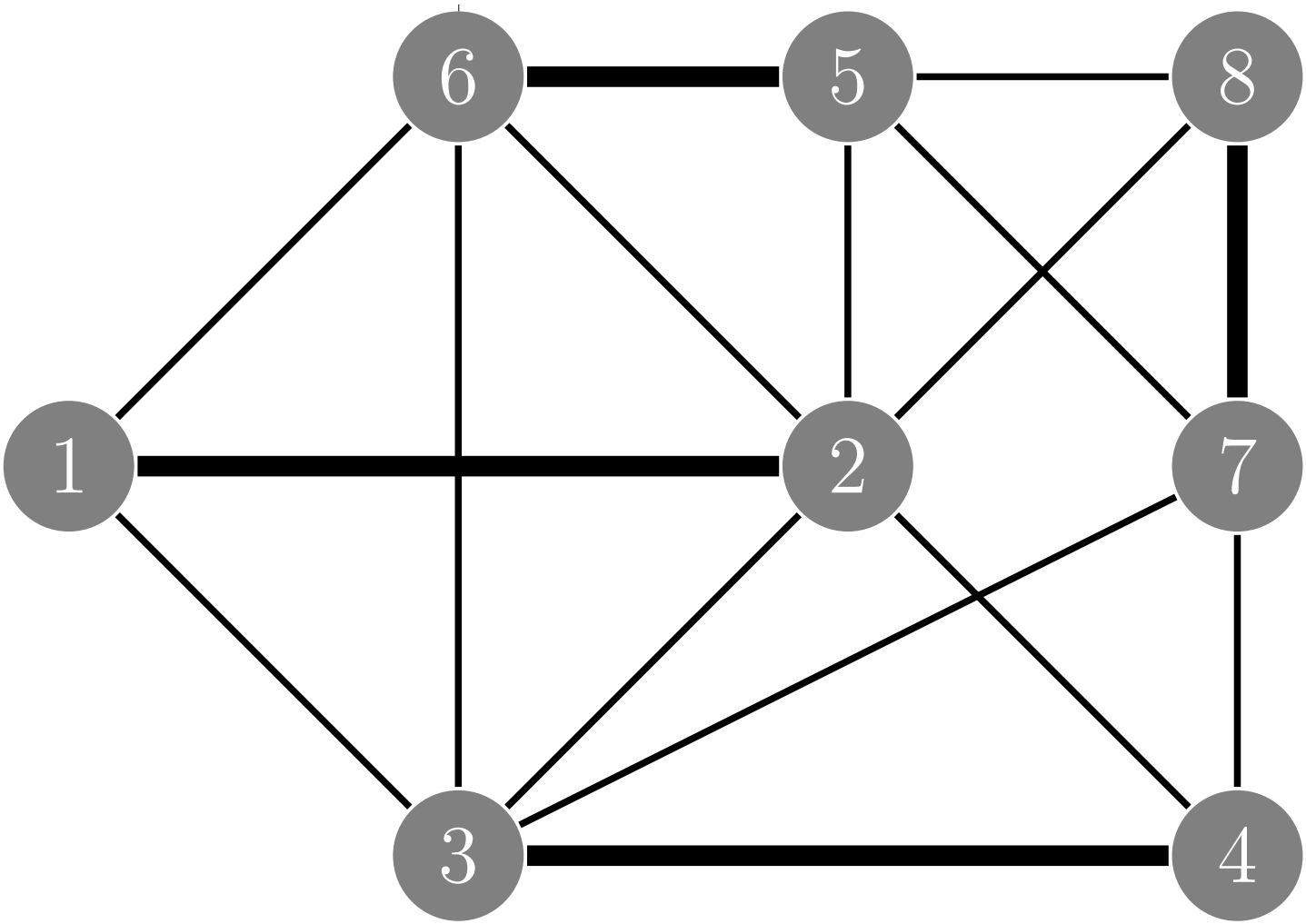
Data

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

var x {E} binary;

maximize z: sum {(i,j) in E} x[i,j];

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

```
set V := 1 2 3 4 5 6 7 8;
set E :=
      1 2
      1 3
      1 6
      2 3
      2 4
      2 5
      2 6
      2 8
      3 4
      3 6
      3 7
      4 7
      5 6
      5 7
      5 8
      7 8
;
```

Solution

```
z = 4

x :=
1 2    1
1 3    0
1 6    0
2 3    0
2 4    0
2 5    0
2 6    0
2 8    0
3 4    1
3 6    0
3 7    0
4 7    0
5 6    1
5 7    0
5 8    0
7 8    1
;
```



Related notions: covering and packing

Given:

- graph $G = (V, E)$

we can define:

- covering by nodes: a set $R \subseteq V$ such that every edge $e \in E$ has at least one endpoint in R
- packing by edges: a set $F \subseteq E$ such that every node $i \in V$ is the endpoint of at most one $e \in F$
  - i.e., *a matching*
  - but we may want to pack with more general structures
    $\rightarrow$ kidney exchange problems

This lesson

- Graph problems
  - partitioning
  - stable set
- Next lesson:
  - matching problems