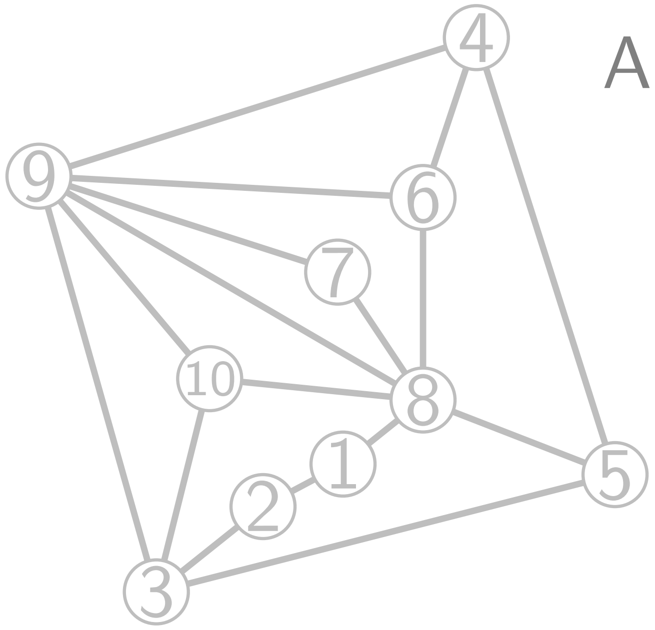
Visualisation of graphs

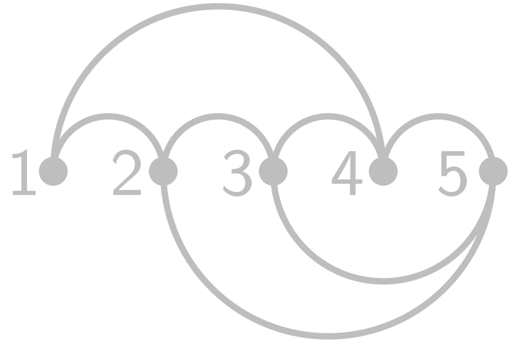
Introduction

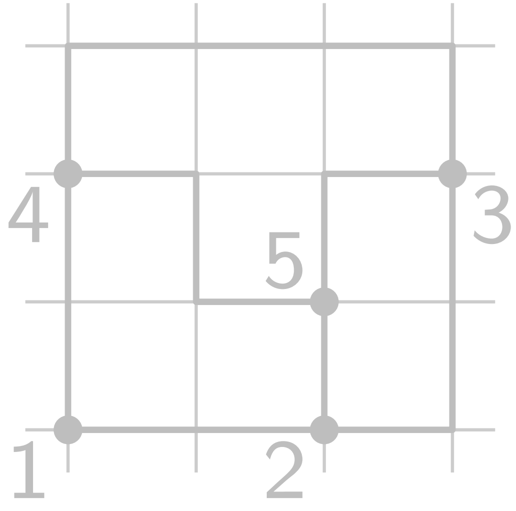
The graph visualisation problem

Antonios Symvonis · Chrysanthi Raftopoulou

Fall semester 2022







The slides of this presentation were created by researchers at Karlsruhe Institute of Technology (KIT), TU Wien, U Wuerzburg, U Konstanz, ...

Graphs and their representations

What is a graph?

- graph $G = (V, E)$
- vertices $V = \{v_1, v_2, \ldots, v_n\}$
- edge $E = \{e_1, e_2, \ldots, e_m\}$

Graphs and their representations

What is a graph?

- graph $G = (V, E)$
- vertices $V = \{v_1, v_2, \ldots, v_n\}$
- edge $E = \{e_1, e_2, \ldots, e_m\}$

Representation?

Graphs and their representations

What is a graph?

- graph $G = (V, E)$
- vertices $V = \{v_1, v_2, \ldots, v_n\}$
- edge $E = \{e_1, e_2, \ldots, e_m\}$

Representation?

- Set notation

$$
\begin{aligned}
V = \{&v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}\} \\
E = \{&\{v_1, v_2\}, \{v_1, v_8\}, \{v_2, v_3\}, \{v_3, v_5\}, \{v_3, v_9\}, \\
&\{v_3, v_{10}\}, \{v_4, v_5\}, \{v_4, v_6\}, \{v_4, v_9\}, \{v_5, v_8\}, \\
&\{v_6, v_8\}, \{v_6, v_9\}, \{v_7, v_8\}, \{v_7, v_9\}, \{v_8, v_{10}\}, \\
&\{v_9, v_{10}\}\}
\end{aligned}
$$

Graphs and their representations

What is a graph?

- graph $G = (V, E)$
- vertices $V = \{v_1, v_2, \ldots, v_n\}$
- edge $E = \{e_1, e_2, \ldots, e_m\}$

Representation?

- Set notation

$$
\begin{aligned}
V &= \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}\} \\
E &= \{\{v_1, v_2\}, \{v_1, v_8\}, \{v_2, v_3\}, \{v_3, v_5\}, \{v_3, v_9\}, \\
&\quad \{v_3, v_{10}\}, \{v_4, v_5\}, \{v_4, v_6\}, \{v_4, v_9\}, \{v_5, v_8\}, \\
&\quad \{v_6, v_8\}, \{v_6, v_9\}, \{v_7, v_8\}, \{v_7, v_9\}, \{v_8, v_{10}\}, \\
&\quad \{v_9, v_{10}\}\}
\end{aligned}
$$

- Adjacency list

v_1: v_2, v_8
v_2: v_1, v_3
v_3: v_2, v_5, v_9, v_{10}
v_4: v_5, v_6, v_9
v_5: v_3, v_4, v_8
v_6: v_4, v_8, v_9
v_7: v_8, v_9
v_8: $v_1, v_5, v_6, v_7, v_9, v_{10}$
v_9: $v_3, v_4, v_6, v_7, v_8, v_{10}$
v_{10}: v_3, v_8, v_9

Graphs and their representations

What is a graph?

- graph $G = (V, E)$
- vertices $V = \{v_1, v_2, \ldots, v_n\}$
- edge $E = \{e_1, e_2, \ldots, e_m\}$

Representation?

- Set notation

$$
\begin{aligned}
V &= \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}\} \\
E &= \{\{v_1, v_2\}, \{v_1, v_8\}, \{v_2, v_3\}, \{v_3, v_5\}, \{v_3, v_9\}, \\
&\quad \{v_3, v_{10}\}, \{v_4, v_5\}, \{v_4, v_6\}, \{v_4, v_9\}, \{v_5, v_8\}, \\
&\quad \{v_6, v_8\}, \{v_6, v_9\}, \{v_7, v_8\}, \{v_7, v_9\}, \{v_8, v_{10}\}, \\
&\quad \{v_9, v_{10}\}\}
\end{aligned}
$$

- Adjacency list

v_1: v_2, v_8
v_2: v_1, v_3
v_3: v_2, v_5, v_9, v_{10}
v_4: v_5, v_6, v_9
v_5: v_3, v_4, v_8
v_6: v_4, v_8, v_9
v_7: v_8, v_9
v_8: $v_1, v_5, v_6, v_7, v_9, v_{10}$
v_9: $v_3, v_4, v_6, v_7, v_8, v_{10}$
v_{10}: v_3, v_8, v_9

- Adjacency matrix

$$
\begin{pmatrix}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\
0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0
\end{pmatrix}
$$

Graphs and their representations

What is a graph?

- graph $G = (V, E)$
- vertices $V = \{v_1, v_2, \dots, v_n\}$
- edge $E = \{e_1, e_2, \dots, e_m\}$

Representation?

- Set notation

$$V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}\}$$
$$E = \{\{v_1, v_2\}, \{v_1, v_8\}, \{v_2, v_3\}, \{v_3, v_5\}, \{v_3, v_9\}, \{v_3, v_{10}\}, \{v_4, v_5\}, \{v_4, v_6\}, \{v_4, v_9\}, \{v_5, v_8\}, \{v_6, v_8\}, \{v_6, v_9\}, \{v_7, v_8\}, \{v_7, v_9\}, \{v_8, v_{10}\}, \{v_9, v_{10}\}\}$$

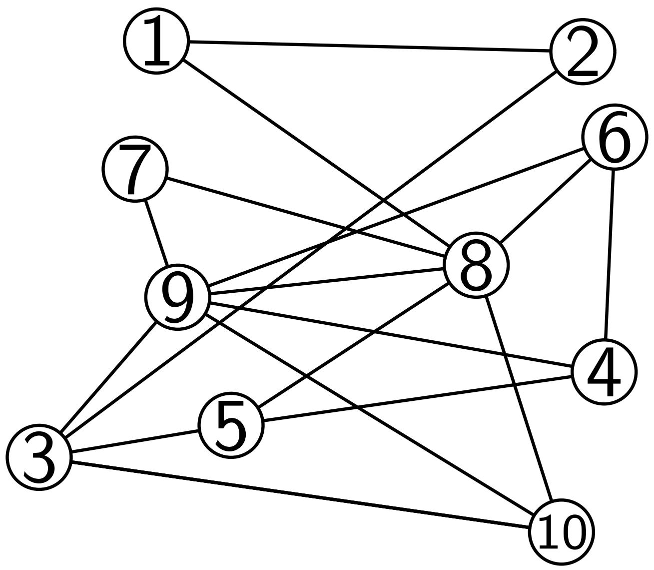
- Adjacency list

v_1: v_2, v_8
v_2: v_1, v_3
v_3: v_2, v_5, v_9, v_{10}
v_4: v_5, v_6, v_9
v_5: v_3, v_4, v_8
v_6: v_4, v_8, v_9
v_7: v_8, v_9
v_8: $v_1, v_5, v_6, v_7, v_9, v_{10}$
v_9: $v_3, v_4, v_6, v_7, v_8, v_{10}$
v_{10}: v_3, v_8, v_9

- Adjacency matrix

$$\begin{pmatrix}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\
0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0
\end{pmatrix}$$

- Drawing



Graphs and their representations

What is a graph?

- graph $G = (V, E)$
- vertices $V = \{v_1, v_2, \ldots, v_n\}$
- edge $E = \{e_1, e_2, \ldots, e_m\}$

Representation?

- Set notation

$$V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}\}$$
$$E = \{\{v_1, v_2\}, \{v_1, v_8\}, \{v_2, v_3\}, \{v_3, v_5\}, \{v_3, v_9\}, \{v_3, v_{10}\}, \{v_4, v_5\}, \{v_4, v_6\}, \{v_4, v_9\}, \{v_5, v_8\}, \{v_6, v_8\}, \{v_6, v_9\}, \{v_7, v_8\}, \{v_7, v_9\}, \{v_8, v_{10}\}, \{v_9, v_{10}\}\}$$

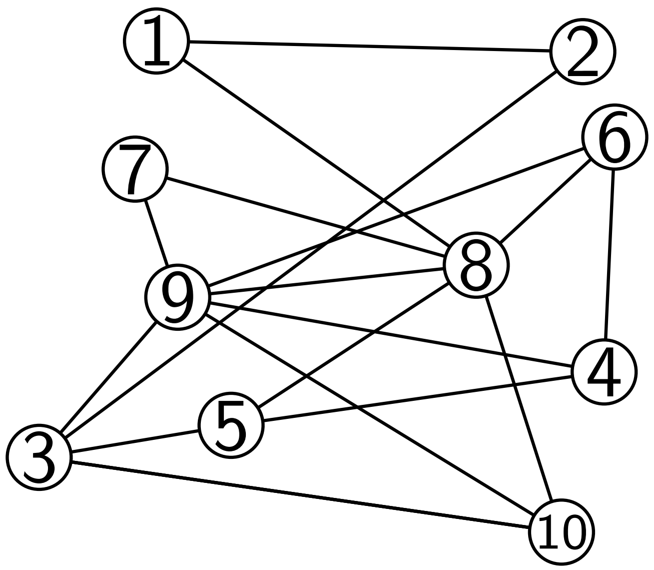
- Adjacency list

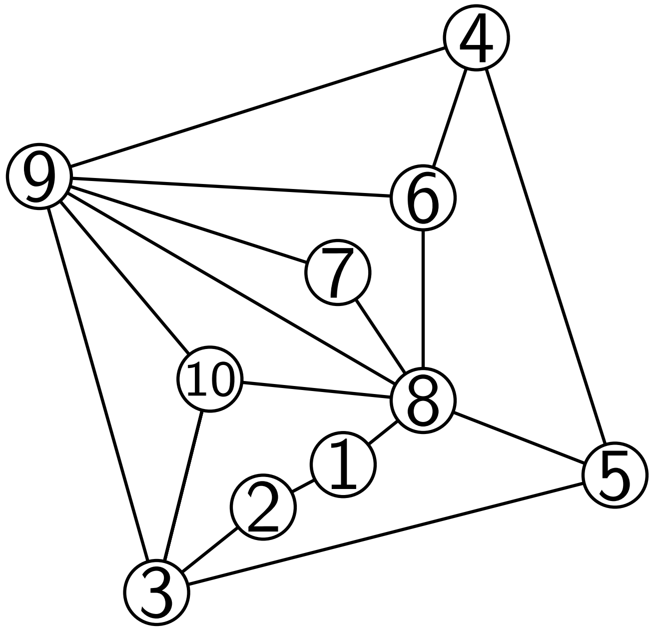
$v_1 :\ v_2, v_8$
$v_2 :\ v_1, v_3$
$v_3 :\ v_2, v_5, v_9, v_{10}$
$v_4 :\ v_5, v_6, v_9$
$v_5 :\ v_3, v_4, v_8$
$v_6 :\ v_4, v_8, v_9$
$v_7 :\ v_8, v_9$
$v_8 :\ v_1, v_5, v_6, v_7, v_9, v_{10}$
$v_9 :\ v_3, v_4, v_6, v_7, v_8, v_{10}$
$v_{10} :\ v_3, v_8, v_9$

- Adjacency matrix

$$\begin{pmatrix}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\
0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0
\end{pmatrix}$$

- Drawing





Why draw graphs?

Why draw graphs?

Graphs are a mathematical representation of real physical and abstract networks.

Why draw graphs?

Graphs are a mathematical representation of real physical and abstract networks.

Abstract networks

- Social networks
- Communication networks
- Phylogenetic networks
- Metabolic networks
- Class/Object Relation Digraphs (UML)
- ...

Why draw graphs?

Graphs are a mathematical representation of real physical and abstract networks.

Abstract networks

- Social networks
- Communication networks
- Phylogenetic networks
- Metabolic networks
- Class/Object Relation Digraphs (UML)
- . . .

Physical networks

- Metro systems
- Road networks
- Power grids
- Telecommunication networks
- Integrated circuits
- . . .

Why draw graphs?

Graphs are a mathematical representation of
real physical and abstract networks.

- **People think visually** – complex graphs are hard to grasp without good visualisations!

Why draw graphs?

Graphs are a mathematical representation of real physical and abstract networks.

- **People think visually** – complex graphs are hard to grasp without good visualisations!
- Visualisations help with the **communication** and **exploration** of networks.

Why draw graphs?

Graphs are a mathematical representation of real physical and abstract networks.

- **People think visually** – complex graphs are hard to grasp without good visualisations!
- Visualisations help with the **communication** and **exploration** of networks.
- Some graphs are too big to draw them by hand.

Why draw graphs?

Graphs are a mathematical representation of
real physical and abstract networks.

- **People think visually** – complex graphs are hard to grasp without good visualisations!
- Visualisations help with the **communication** and **exploration** of networks.
- Some graphs are too big to draw them by hand.

We need algorithms that draw graphs automatically to make networks more accessible to humans.

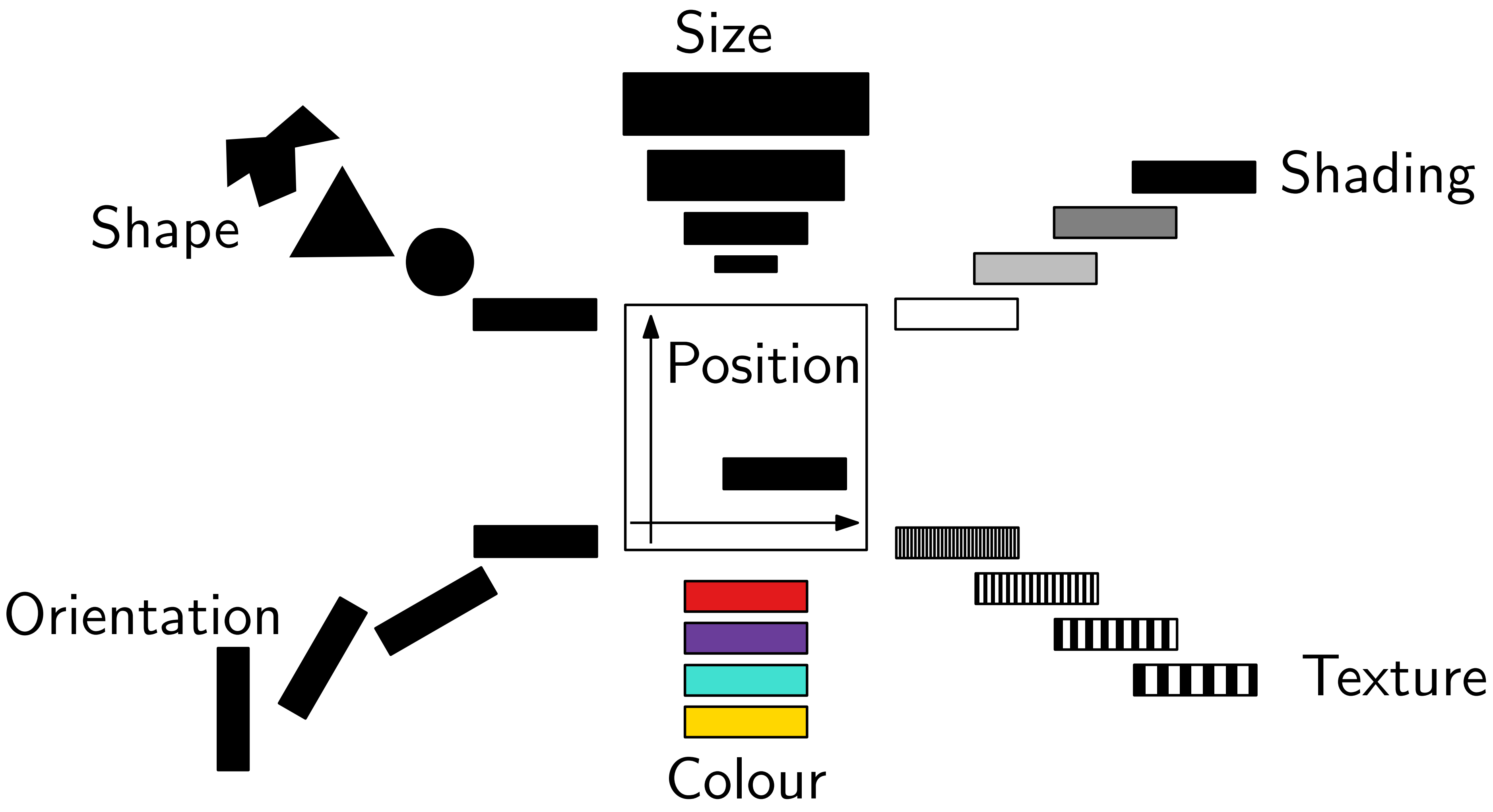
What are we interested in?

What are we interested in?

- Jacques Bertin defined visualising variables (1967)

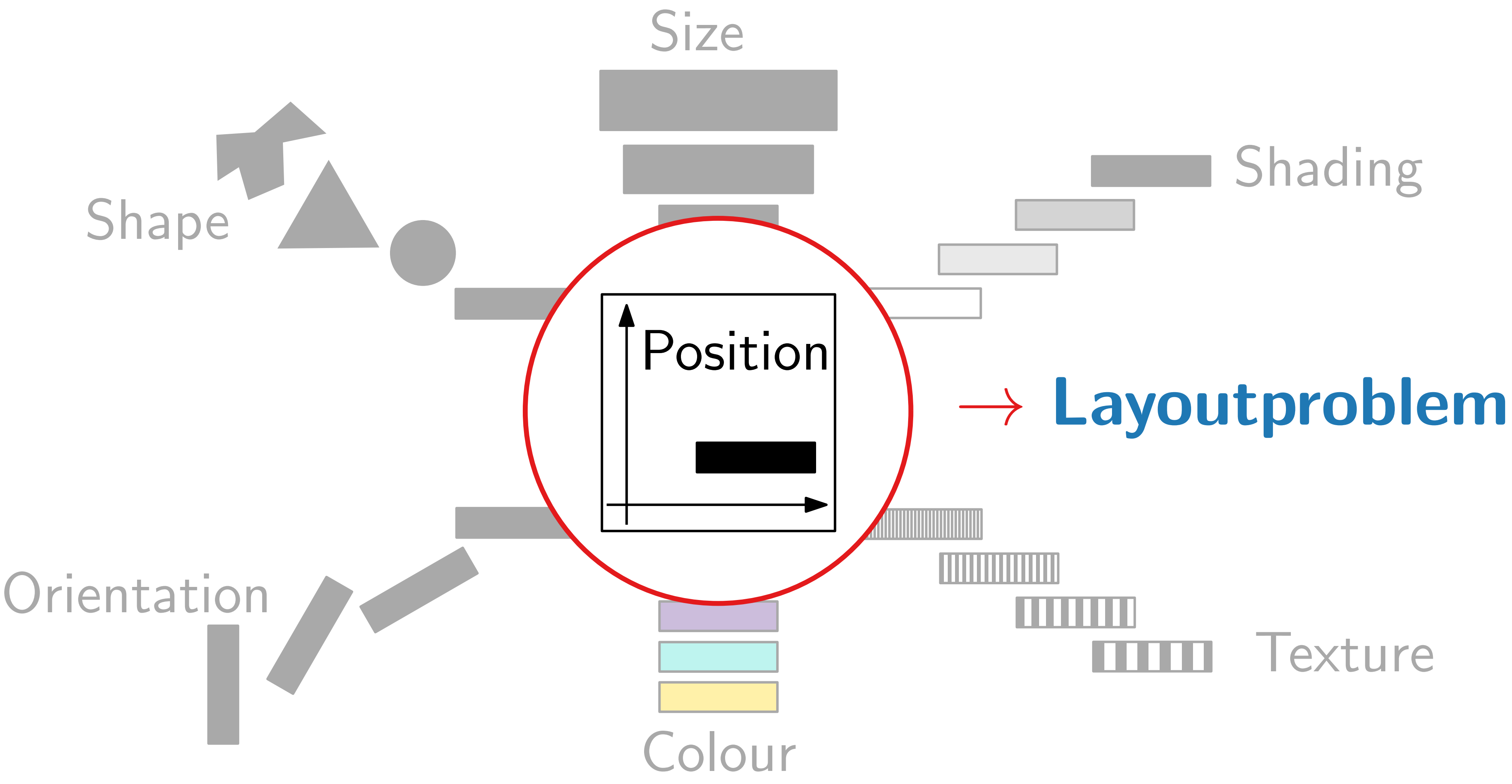
What are we interested in?

- Jacques Bertin defined visualising variables (1967)



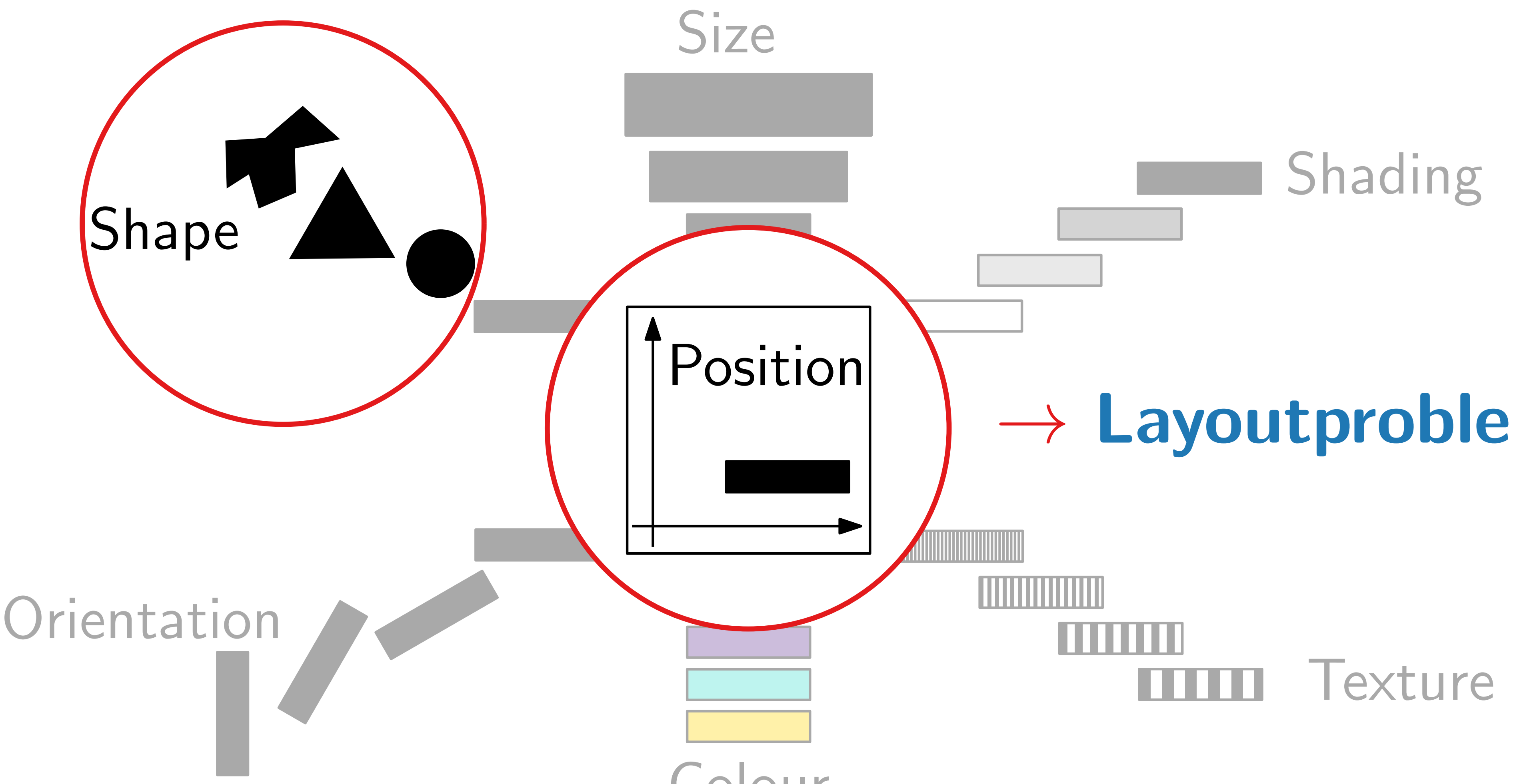
What are we interested in?

- Jacques Bertin defined visualising variables (1967)



What are we interested in?

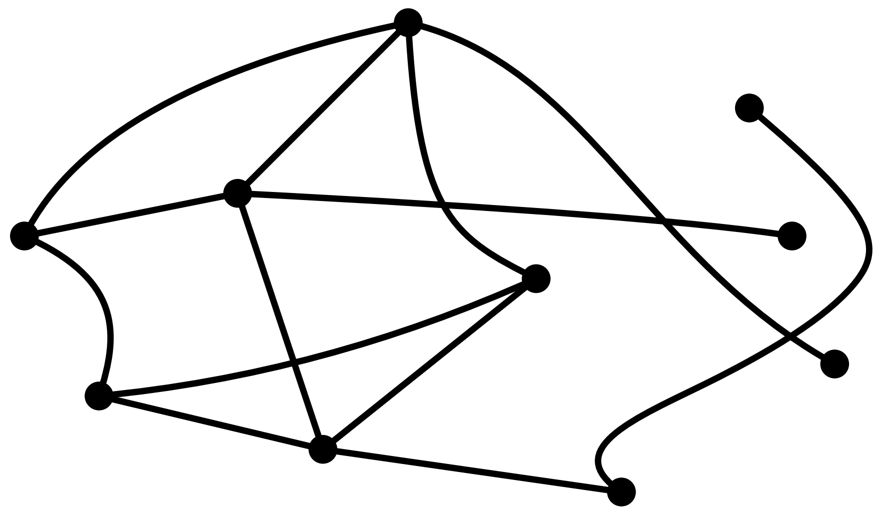
- Jacques Bertin defined visualising variables (1967)



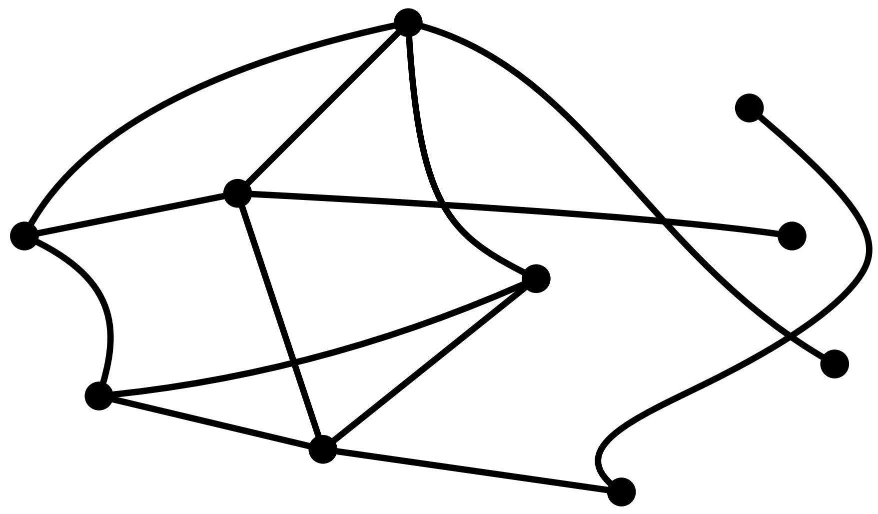
→ **Layoutproblem**

The layout problem

- *Here* restricted to the **standard representation**, so-called node-link diagrams.



The layout problem



- *Here* restricted to the **standard representation**, so-called node-link diagrams.

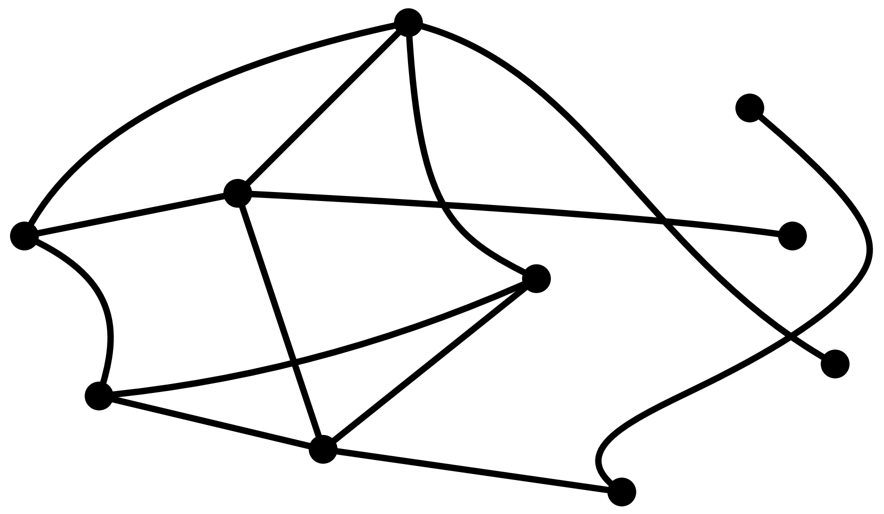
Graph visualisation problem

in: Graph $G = (V, E)$

out:

The layout problem

- *Here* restricted to the **standard representation**, so-called node-link diagrams.



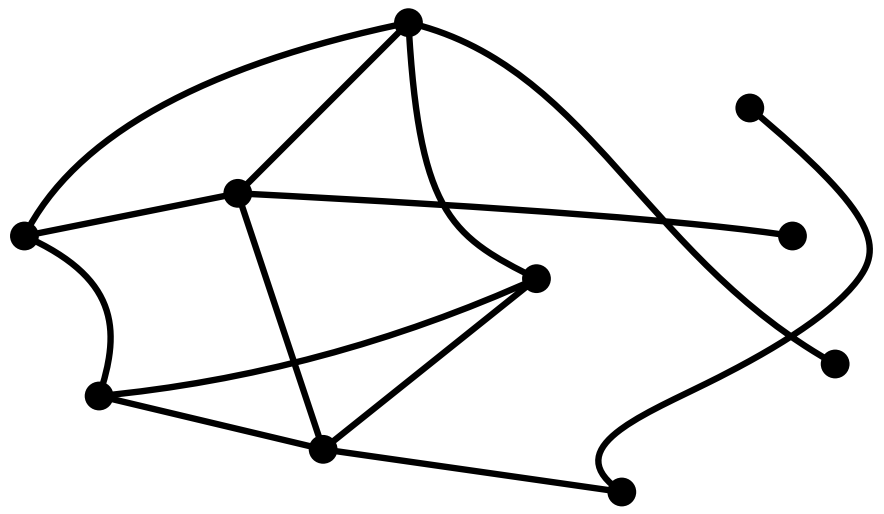
Graph visualisation problem

in: Graph $G = (V, E)$

out: **nice** drawing Γ of G

- $\Gamma \colon V \to \mathbb{R}^2$, vertex $v \mapsto$ point $\Gamma(v)$
- $\Gamma \colon E \to$ curves in $\mathbb{R}^2$, edge $\{u, v\} \mapsto$ simple, open curve $\Gamma(\{u, v\})$ with endpoints $\Gamma(u)$ und $\Gamma(v)$

The layout problem?



- *Here* restricted to the **standard representation**, so-called node-link diagrams.

Graph visualisation problem

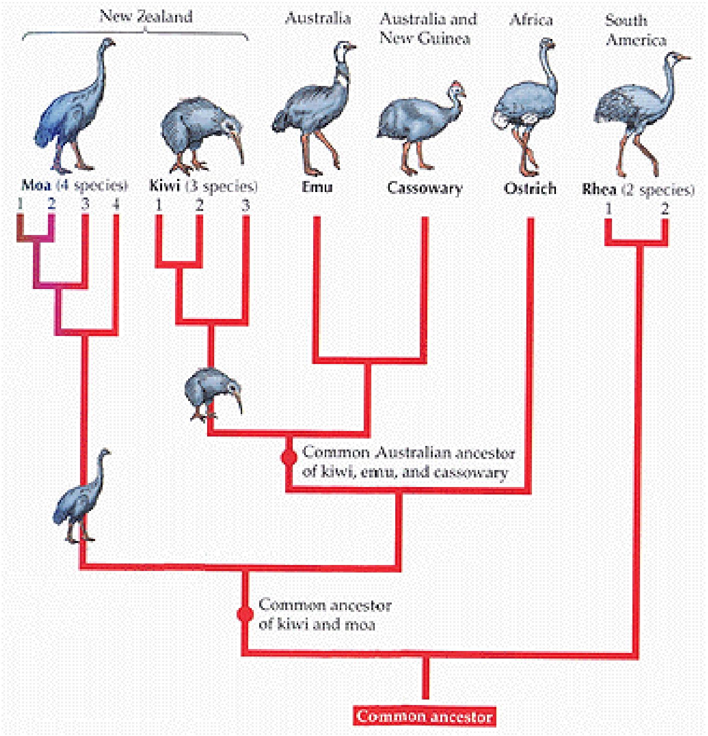
in: Graph $G = (V, E)$

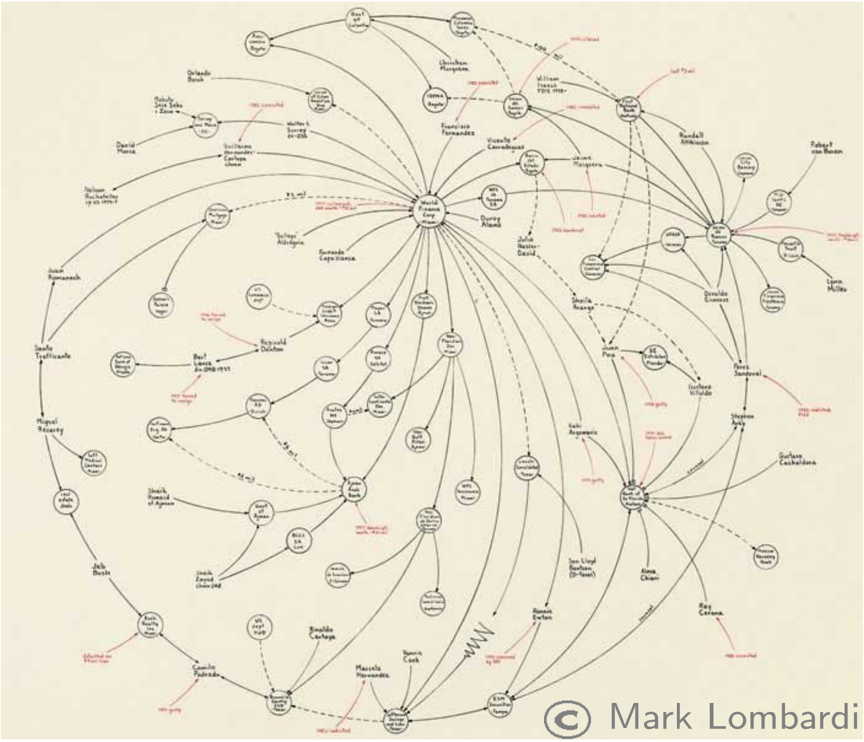
out: **nice** drawing Γ of G

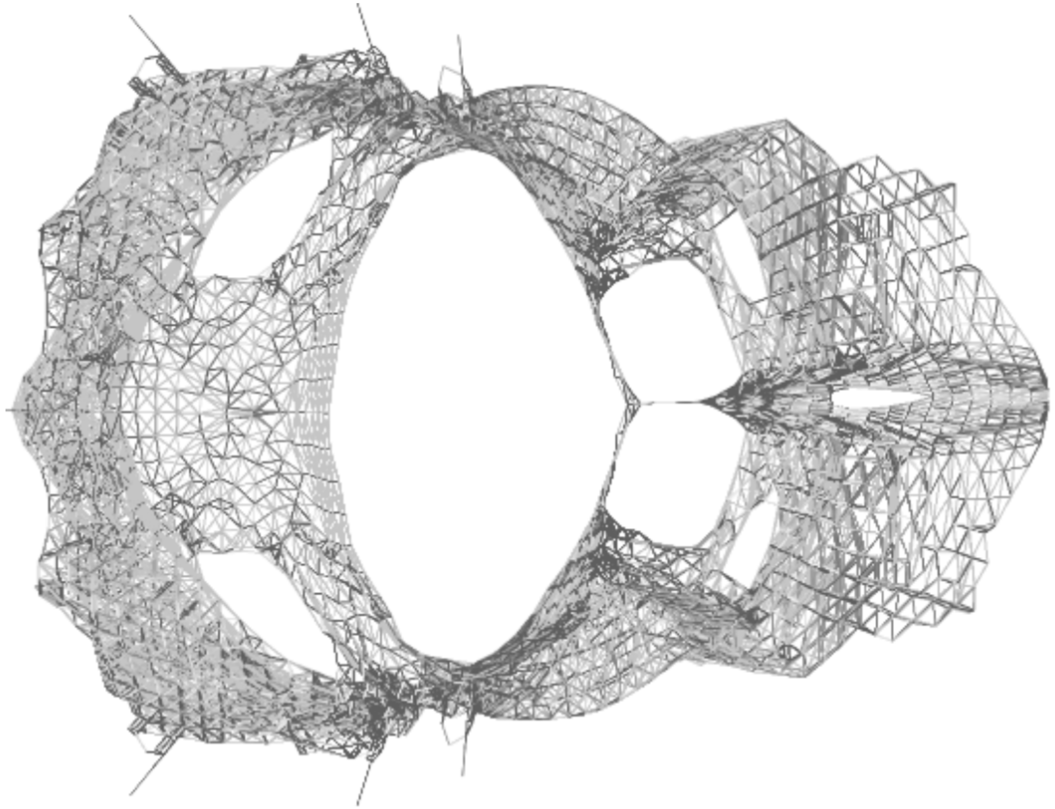
- $\Gamma \colon V \to \mathbb{R}^2$, vertex $v \mapsto$ point $\Gamma(v)$
- $\Gamma \colon E \to$ curves in $\mathbb{R}^2$, edge $\{u, v\} \mapsto$ simple, open curve $\Gamma(\{u, v\})$ with endpoints $\Gamma(u)$ und $\Gamma(v)$

But what is a **nice** drawing?

Examples







- See slides with more examples.

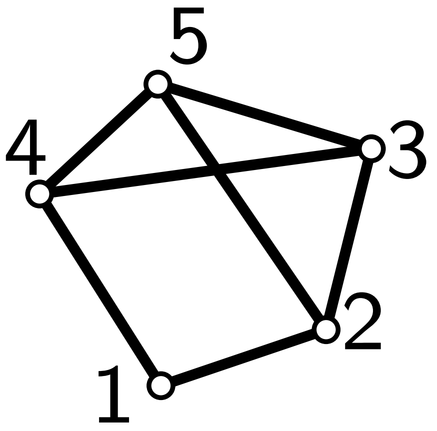
Requirements of a graph layout

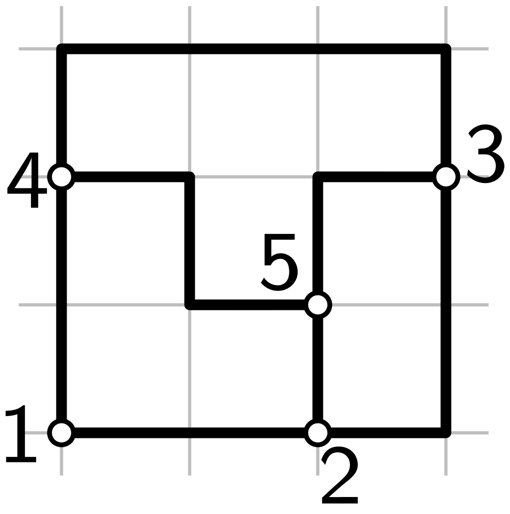
1. Drawing conventions and requirements, e.g.,

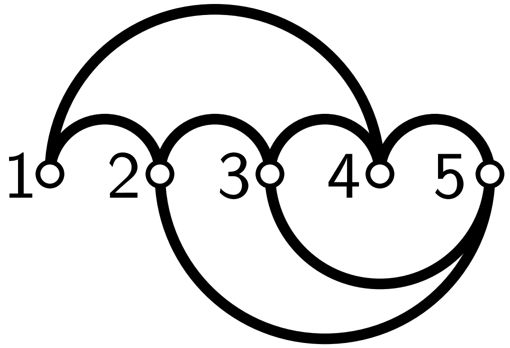
Requirements of a graph layout

1. Drawing conventions and requirements, e.g.,

- straight edges with $\Gamma(uv) = \overline{\Gamma(u)\Gamma(v)}$
- orthogonal edges (i.e. with bends)
- grid drawings
- without crossing

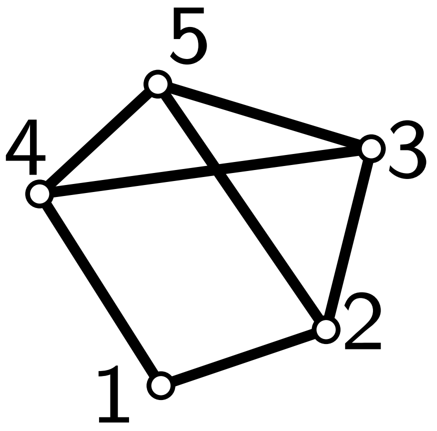


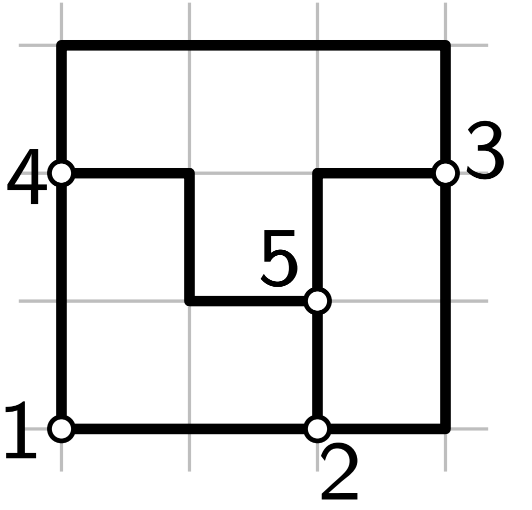


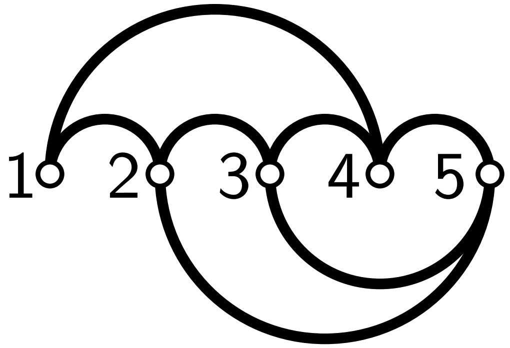


Requirements of a graph layout

1. Drawing conventions and requirements, e.g.,
   - straight edges with $\Gamma(uv) = \overline{\Gamma(u)\Gamma(v)}$
   - orthogonal edges (i.e. with bends)
   - grid drawings
   - without crossing
2. Aesthetics to be optimised, e.g.

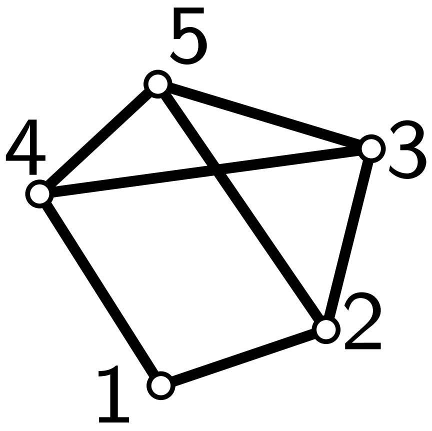


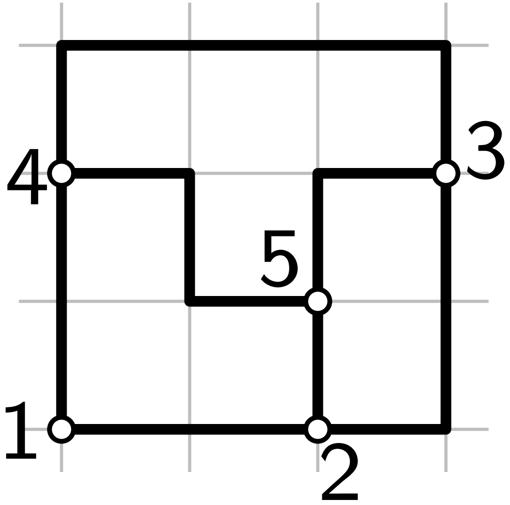


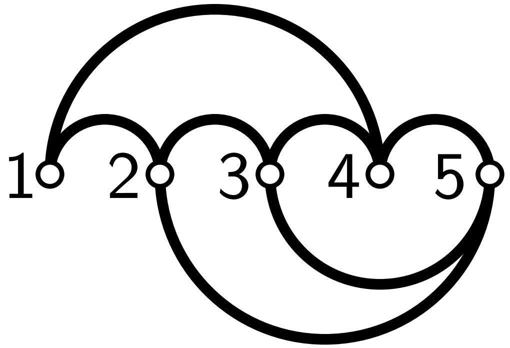


Requirements of a graph layout

1. Drawing conventions and requirements, e.g.,
   - straight edges with $\Gamma(uv) = \overline{\Gamma(u)\Gamma(v)}$
   - orthogonal edges (i.e. with bends)
   - grid drawings
   - without crossing
2. Aesthetics to be optimised, e.g.
   - crossing/bend minimisation
   - edge length uniformity
   - minimising total edge length/drawing area
   - angular resolution
   - symmetry/structure

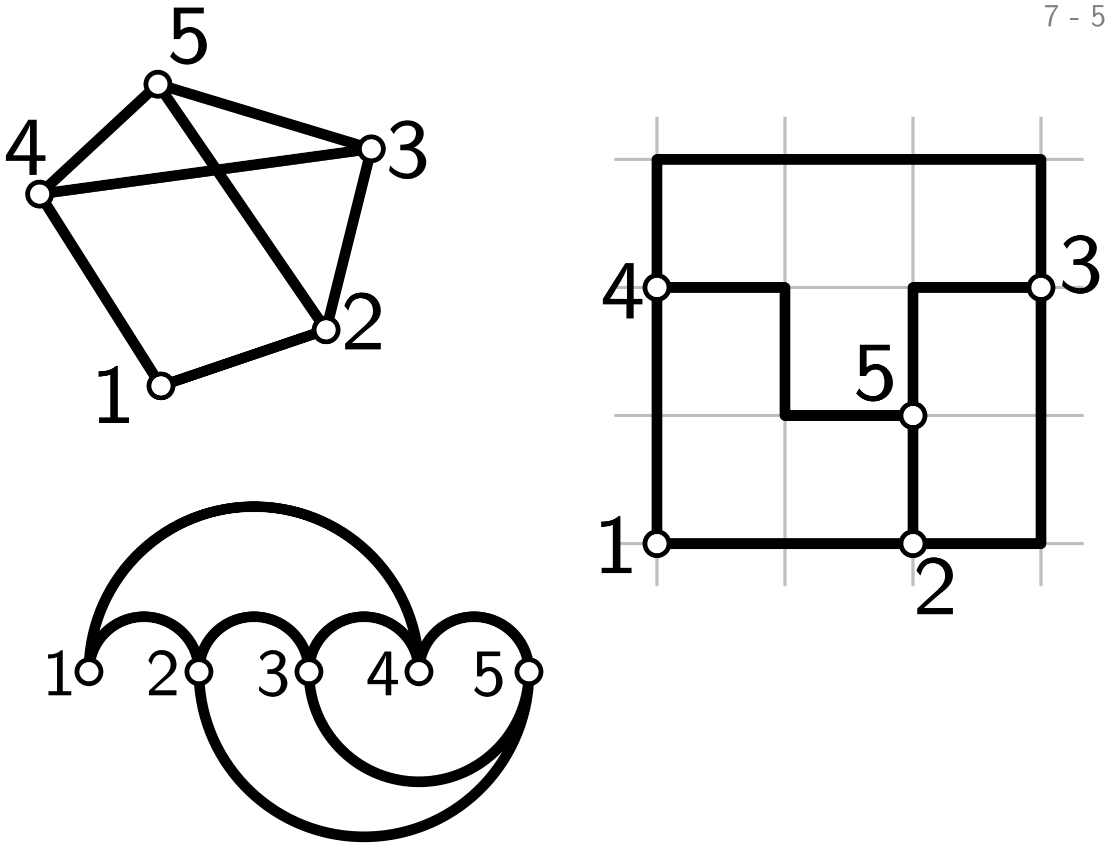






Requirements of a graph layout

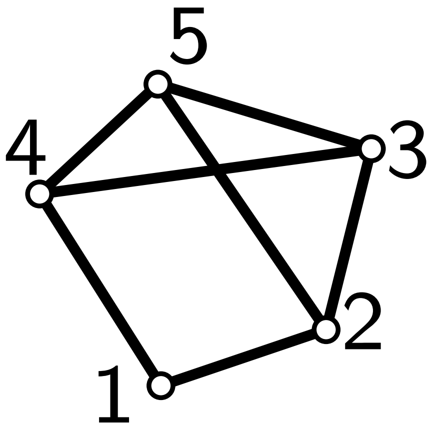
1. Drawing conventions and requirements, e.g.,
   - straight edges with $\Gamma(uv) = \overline{\Gamma(u)\Gamma(v)}$
   - orthogonal edges (i.e. with bends)
   - grid drawings
   - without crossing
2. Aesthetics to be optimised, e.g.
   - crossing/bend minimisation
   - edge length uniformity
   - minimising total edge length/drawing area
   - angular resolution
   - symmetry/structure

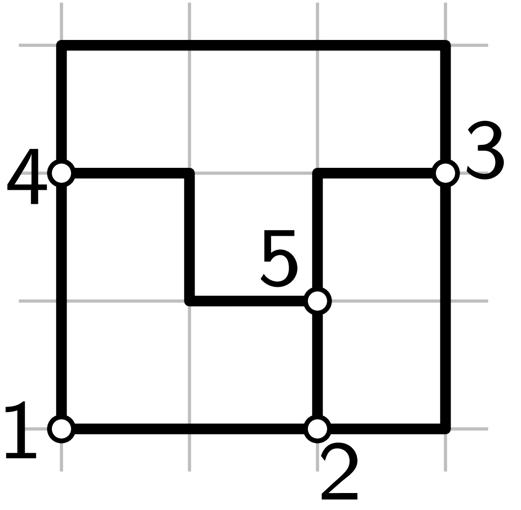


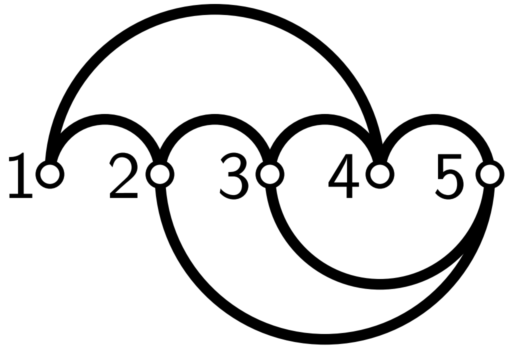
→ lead to NP-hard optimization problems

Requirements of a graph layout

1. Drawing conventions and requirements, e.g.,
   - straight edges with $\Gamma(uv) = \overline{\Gamma(u)\Gamma(v)}$
   - orthogonal edges (i.e. with bends)
   - grid drawings
   - without crossing
2. Aesthetics to be optimised, e.g.
   - crossing/bend minimisation
   - edge length uniformity
   - minimising total edge length/drawing area
   - angular resolution
   - symmetry/structure



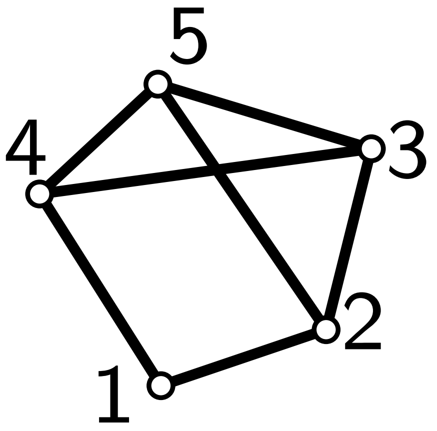


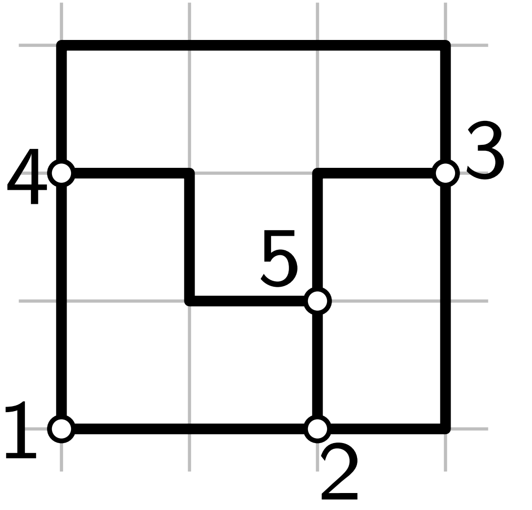


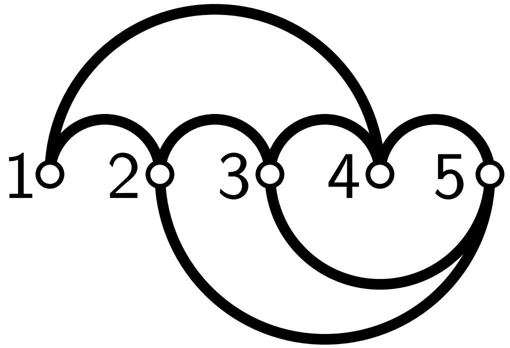
→ lead to NP-hard optimization problems
→ such criteria are often inversely related

Requirements of a graph layout

1. Drawing conventions and requirements, e.g.,
   - straight edges with $\Gamma(uv) = \overline{\Gamma(u)\Gamma(v)}$
   - orthogonal edges (i.e. with bends)
   - grid drawings
   - without crossing
2. Aesthetics to be optimised, e.g.
   - crossing/bend minimisation
   - edge length uniformity
   - minimising total edge length/drawing area
   - angular resolution
   - symmetry/structure
3. Local Constraints, e.g.



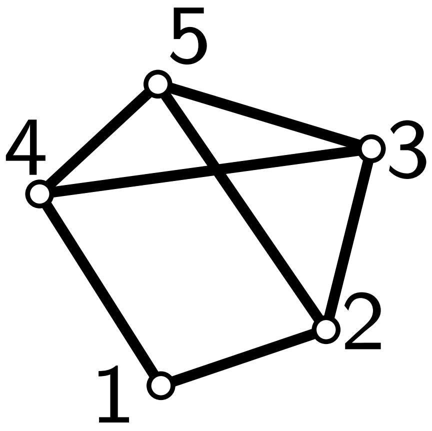


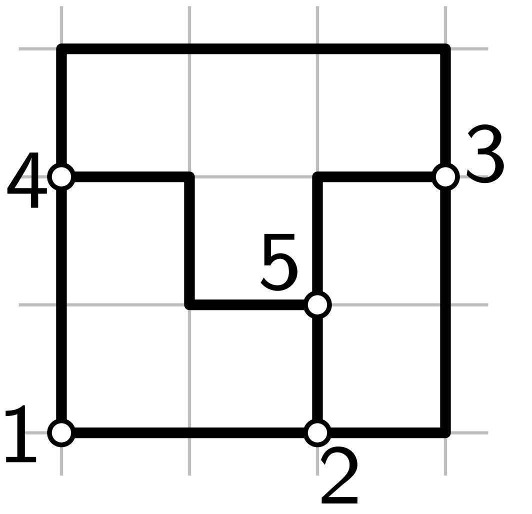


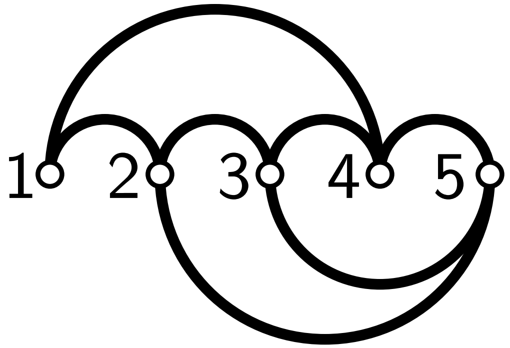
→ lead to NP-hard optimization problems

→ such criteria are often inversely related

Requirements of a graph layout







1. Drawing conventions and requirements, e.g.,
   - straight edges with $\Gamma(uv) = \overline{\Gamma(u)\Gamma(v)}$
   - orthogonal edges (i.e. with bends)
   - grid drawings
   - without crossing
2. Aesthetics to be optimised, e.g.
   - crossing/bend minimisation
   - edge length uniformity
   - minimising total edge length/drawing area
   - angular resolution
   - symmetry/structure

→ lead to NP-hard optimization problems
→ such criteria are often inversely related

3. Local Constraints, e.g.
   - restrictions on neighbouring vertices (e.g., “upward”).
   - restrictions on groups of vertices/edges (e.g., “clustered”).

The layout problem

Graph visualisation problem

in: Graph $G = (V, E)$

out: Drawing Γ of G such that

The layout problem

Graph visualisation problem

in: Graph $G = (V, E)$

out: Drawing Γ of G such that

- **drawing conventions** are met,

The layout problem

Graph visualisation problem

in: Graph $G = (V, E)$

out: Drawing Γ of G such that

- **drawing conventions** are met,
- **aesthetic criteria** are optimised, and

The layout problem

Graph visualisation problem

in: Graph $G = (V, E)$

out: Drawing Γ of G such that

- **drawing conventions** are met,
- **aesthetic criteria** are optimised, and
- some **additional constraints** are satisfied.

The layout problem

Graph visualisation problem

in: Graph $G = (V, E)$

out: Drawing Γ of G such that

- **drawing conventions** are met,
- **aesthetic criteria** are optimised, and
- some **additional constraints** are satisfied.

- Many algorithmically interesting questions arise.
- Rendering problem downstream is ignored.