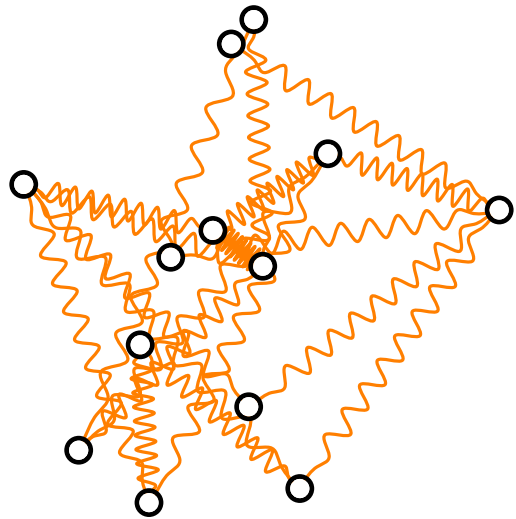


CS F402: Computational Geometry

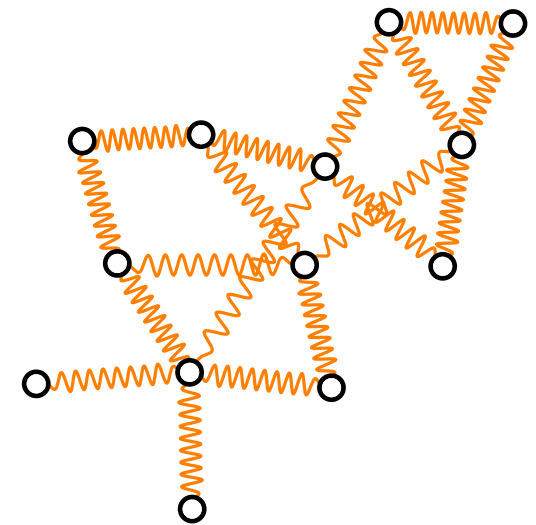
Lecture 9:

GD - Force-Directed Drawing Algorithms-I



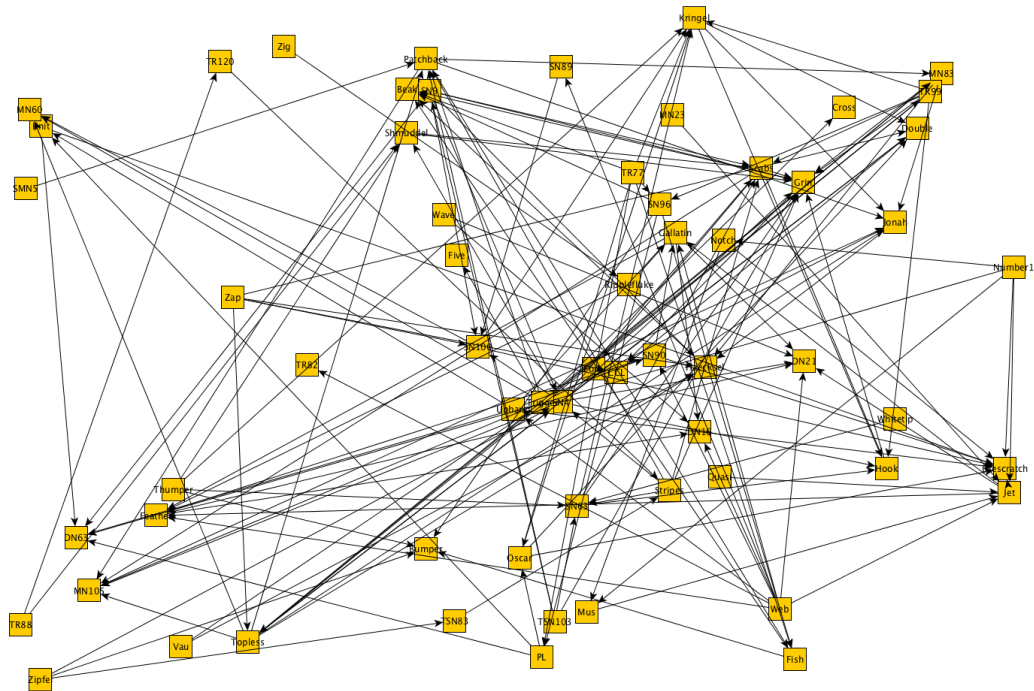
Siddharth Gupta

February 12+14, 2025



General Layout Problem

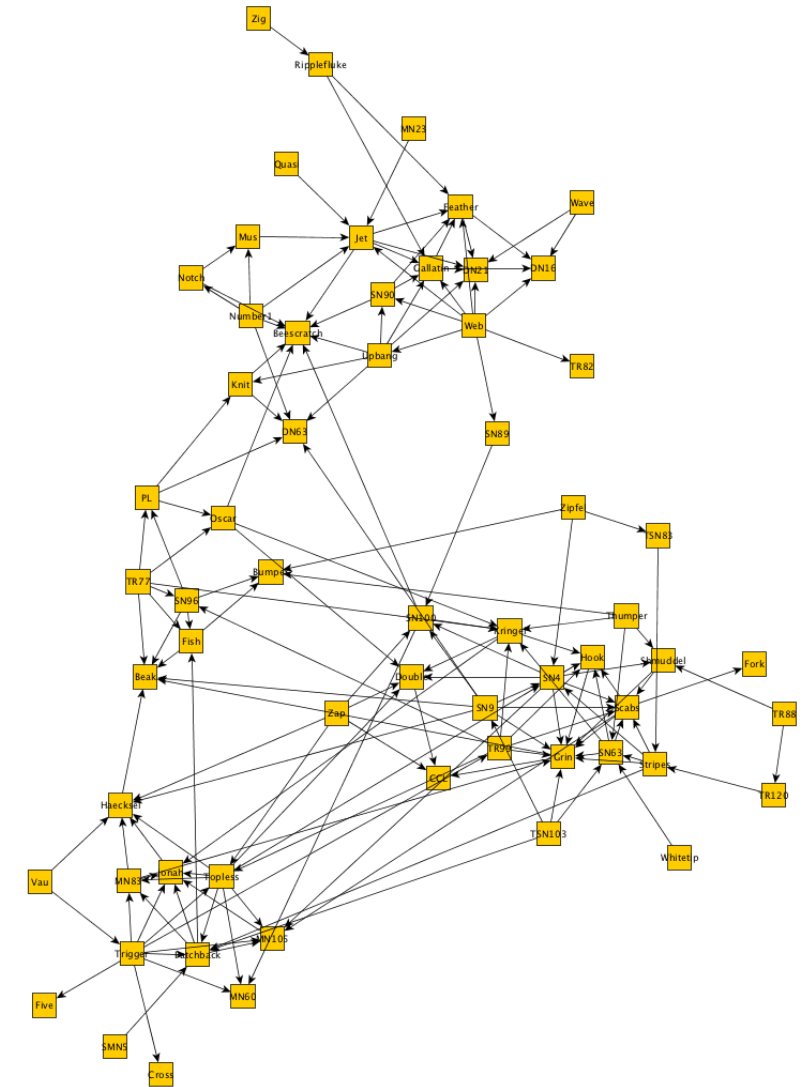
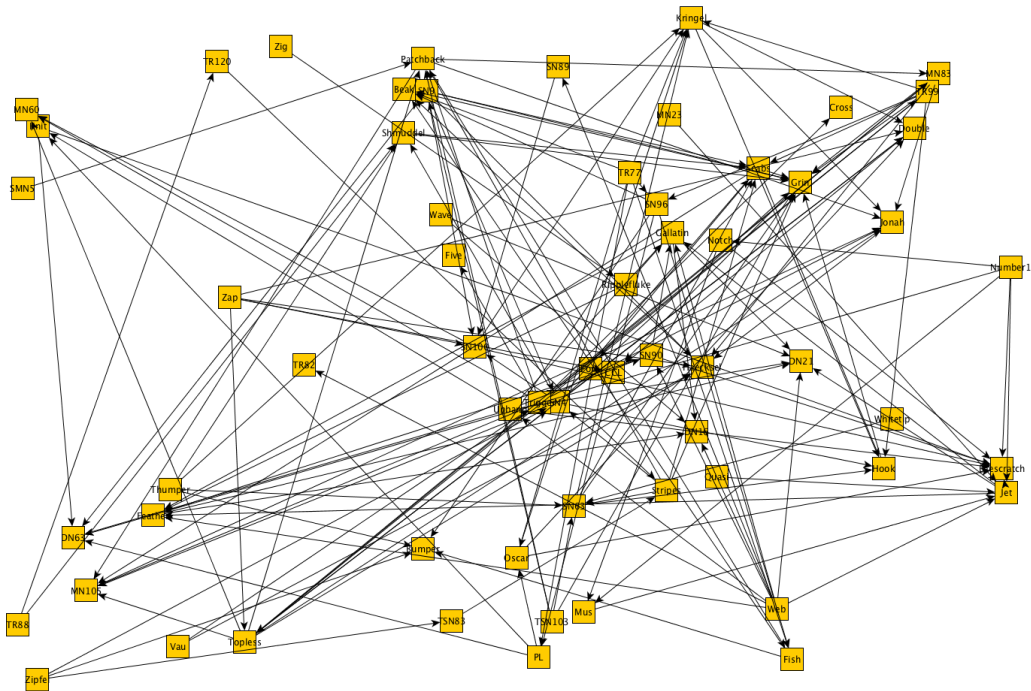
Input: Graph $G = (V, E)$



General Layout Problem

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

Output: Clear and readable straight-line drawing of G

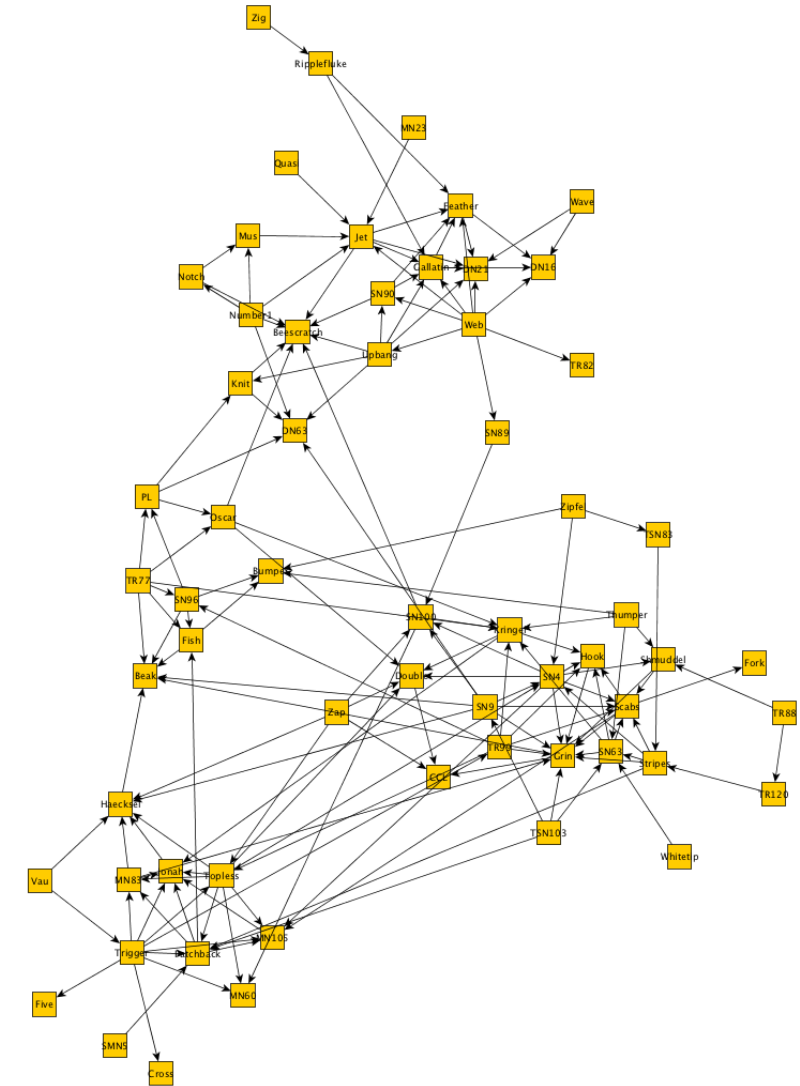


General Layout Problem

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

Output: Clear and readable straight-line drawing of G

Drawing aesthetics:



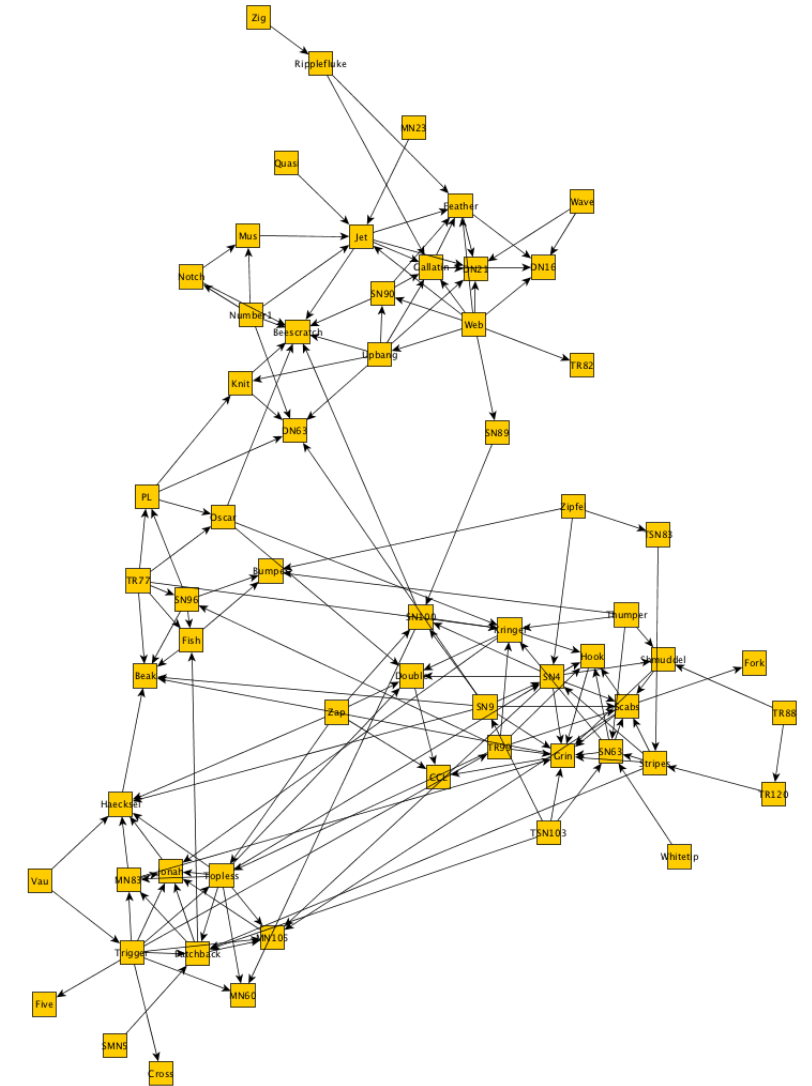
General Layout Problem

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

Output: Clear and readable straight-line drawing of G

Drawing aesthetics:

- adjacent vertices are close



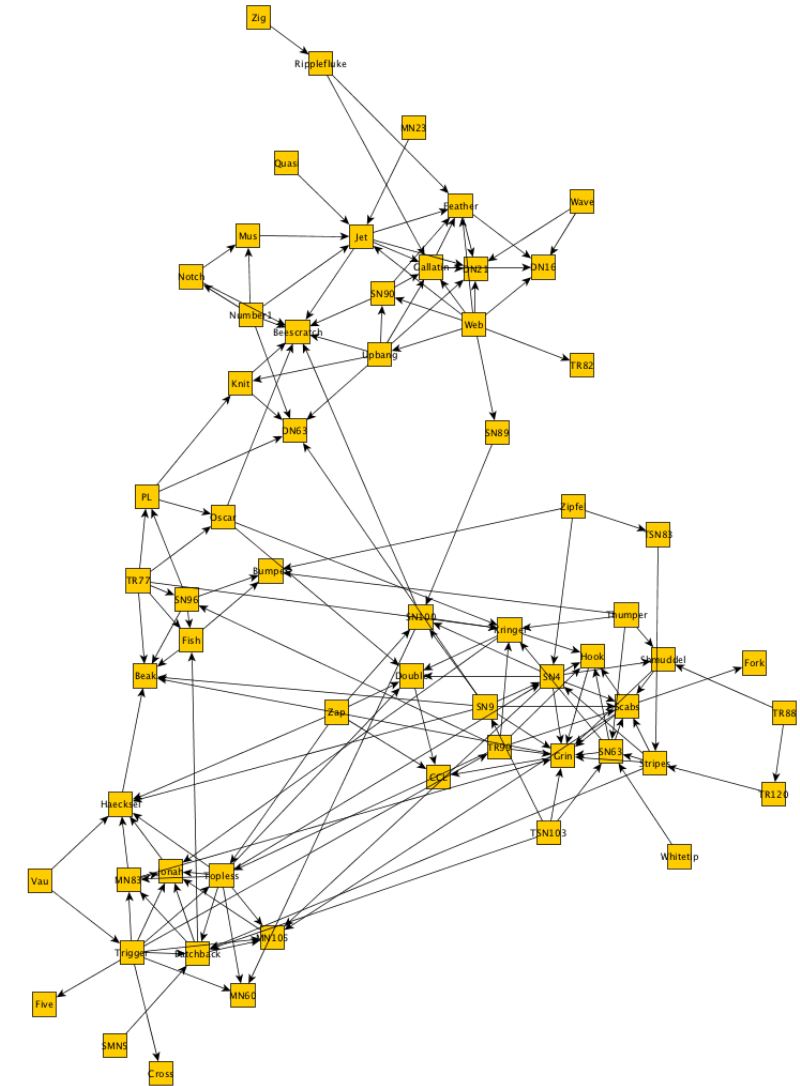
General Layout Problem

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

Output: Clear and readable straight-line drawing of G

Drawing aesthetics:

- adjacent vertices are close
- non-adjacent vertices are far apart



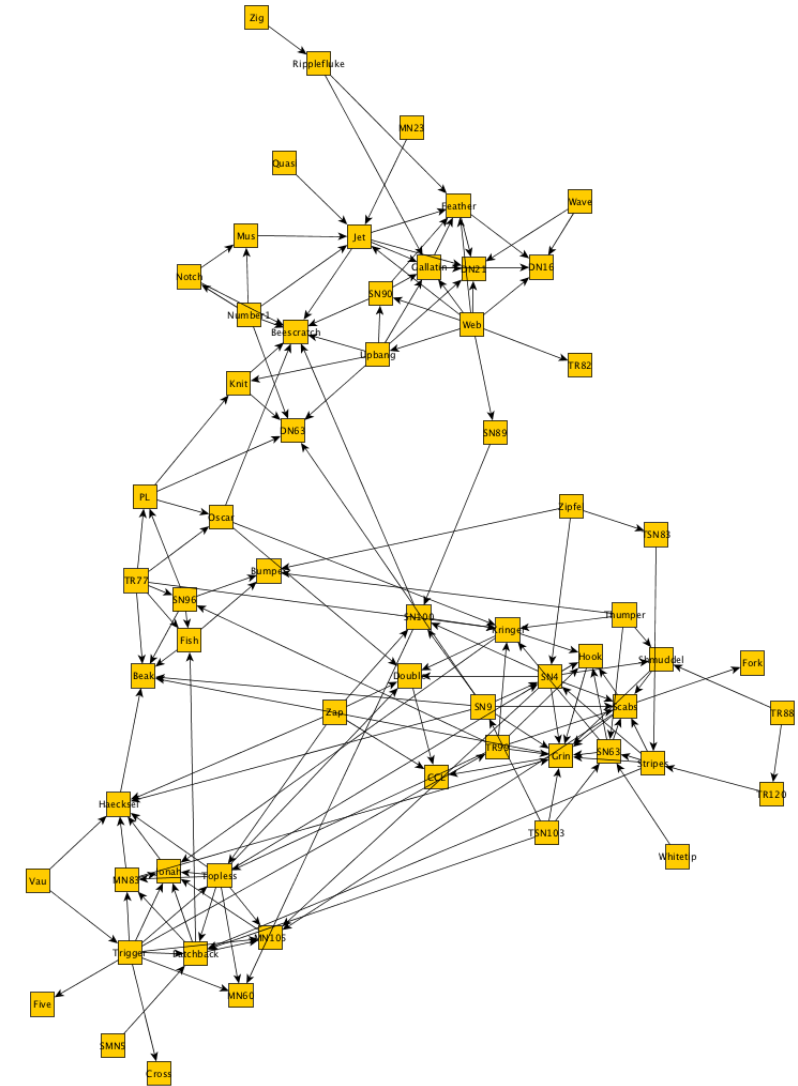
General Layout Problem

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

Output: Clear and readable straight-line drawing of G

Drawing aesthetics:

- adjacent vertices are close
- non-adjacent vertices are far apart
- edges short, straight-line, **similar length**



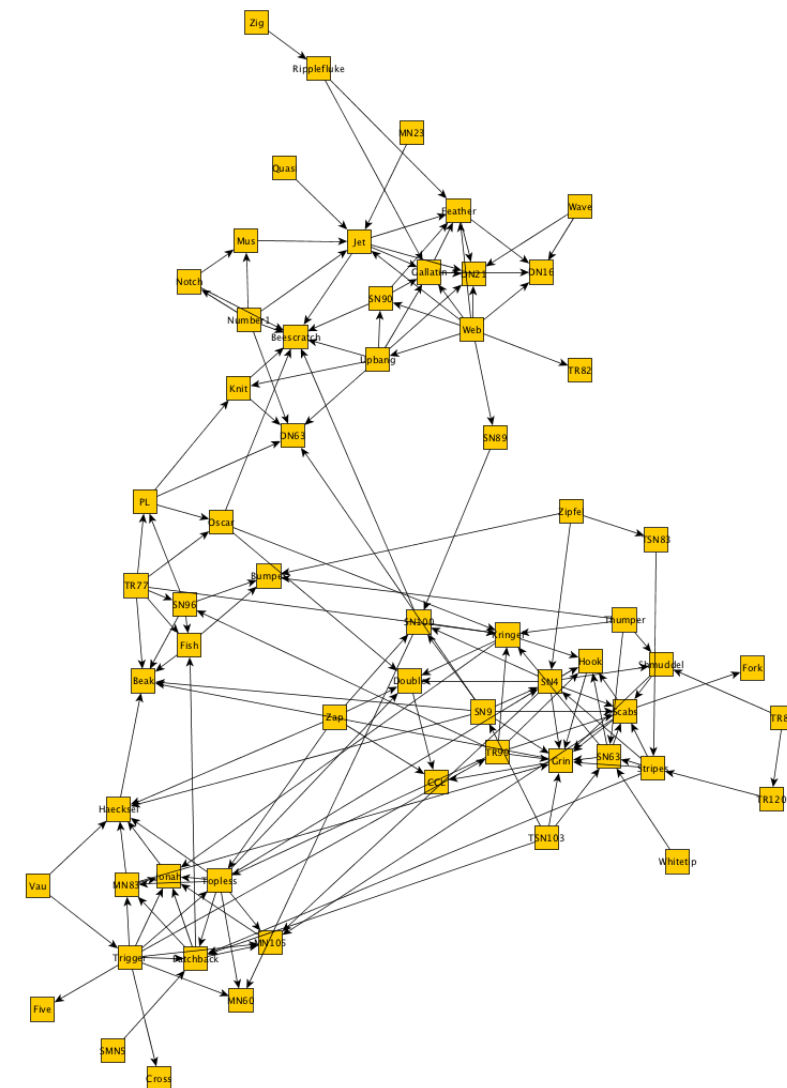
General Layout Problem

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

Output: Clear and readable straight-line drawing of G

Drawing aesthetics:

- adjacent vertices are close
- non-adjacent vertices are far apart
- edges short, straight-line, **similar length**
- densely connected parts (clusters) form communities



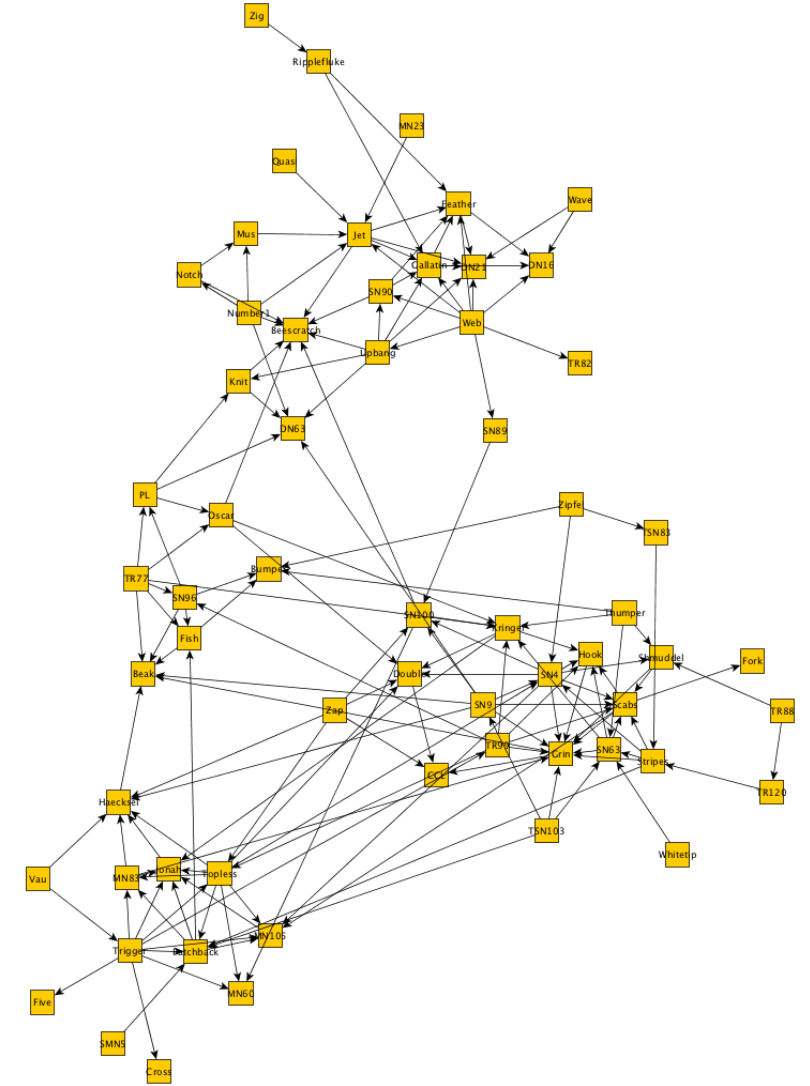
General Layout Problem

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

Output: Clear and readable straight-line drawing of G

Drawing aesthetics:

- adjacent vertices are close
- non-adjacent vertices are far apart
- edges short, straight-line, **similar length**
- densely connected parts (clusters) form communities
- as few crossings as possible



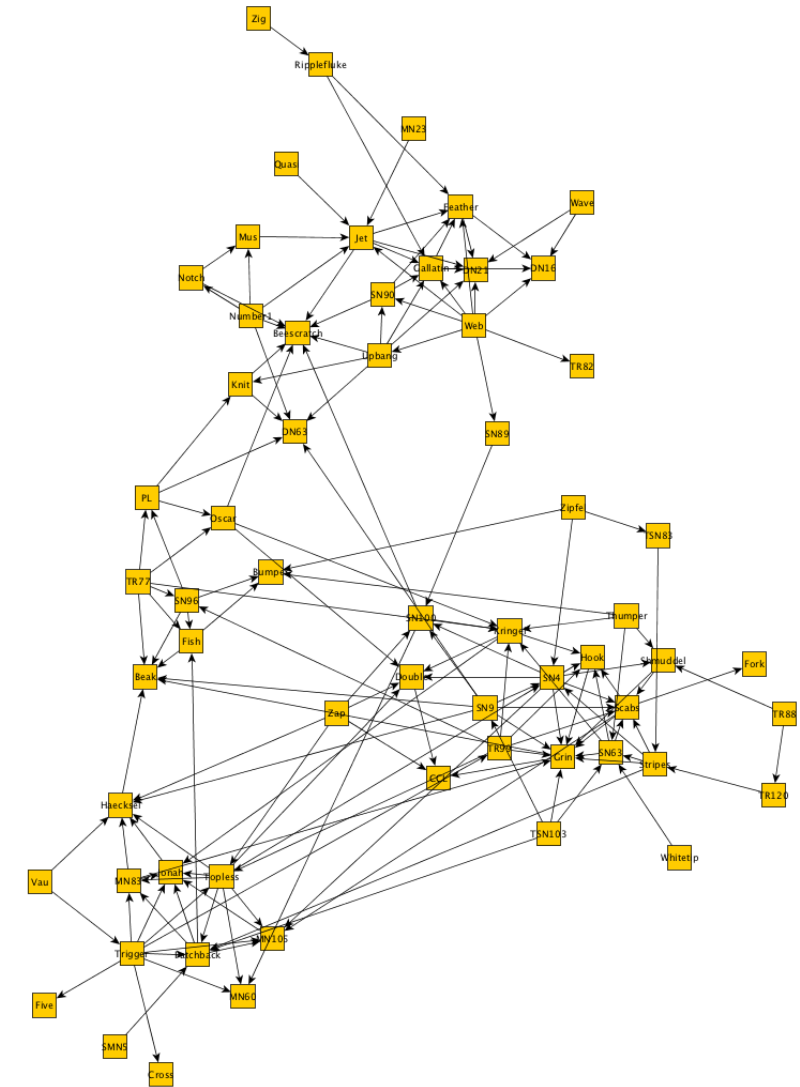
General Layout Problem

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

Output: Clear and readable straight-line drawing of G

Drawing aesthetics:

- adjacent vertices are close
- non-adjacent vertices are far apart
- edges short, straight-line, **similar length**
- densely connected parts (clusters) form communities
- as few crossings as possible
- nodes distributed evenly



General Layout Problem

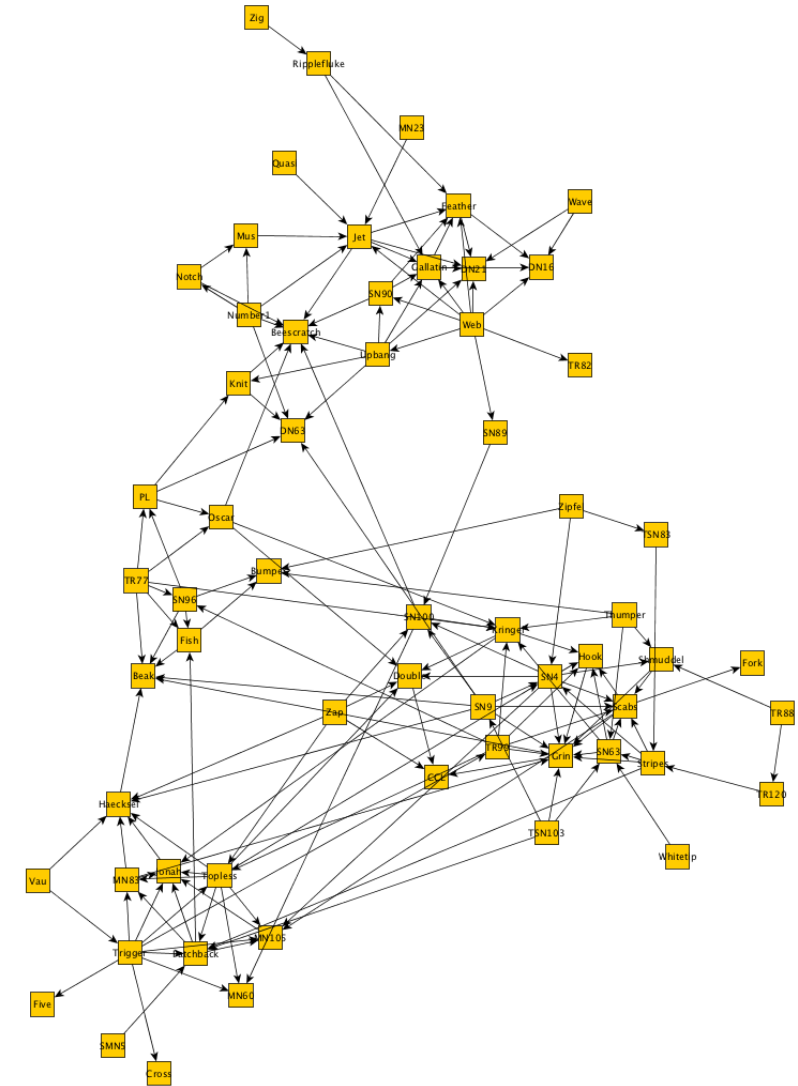
Input: Graph $G = (V, E)$

Output: Clear and readable straight-line drawing of G

Drawing aesthetics:

- adjacent vertices are close
- non-adjacent vertices are far apart
- edges short, straight-line, **similar length**
- densely connected parts (clusters) form communities
- as few crossings as possible
- nodes distributed evenly

Optimization criteria partially contradict each other



Fixed Edge Lengths?

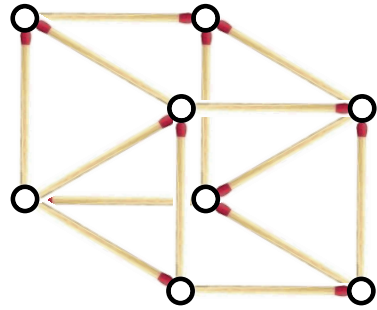
Input: Graph $G = (V, E)$, required edge length $\ell(e)$, $\forall e \in E$

Output: Drawing of G which realizes all the edge lengths

Fixed Edge Lengths?

Input: Graph $G = (V, E)$, required edge length $\ell(e)$, $\forall e \in E$

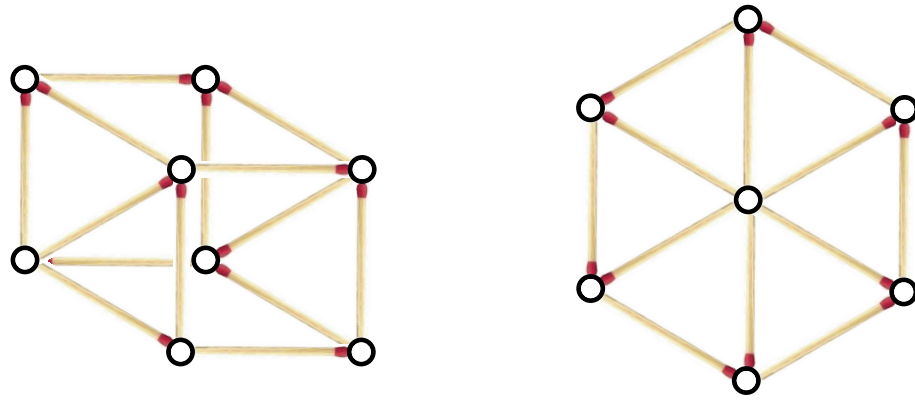
Output: Drawing of G which realizes all the edge lengths



Fixed Edge Lengths?

Input: Graph $G = (V, E)$, required edge length $\ell(e)$, $\forall e \in E$

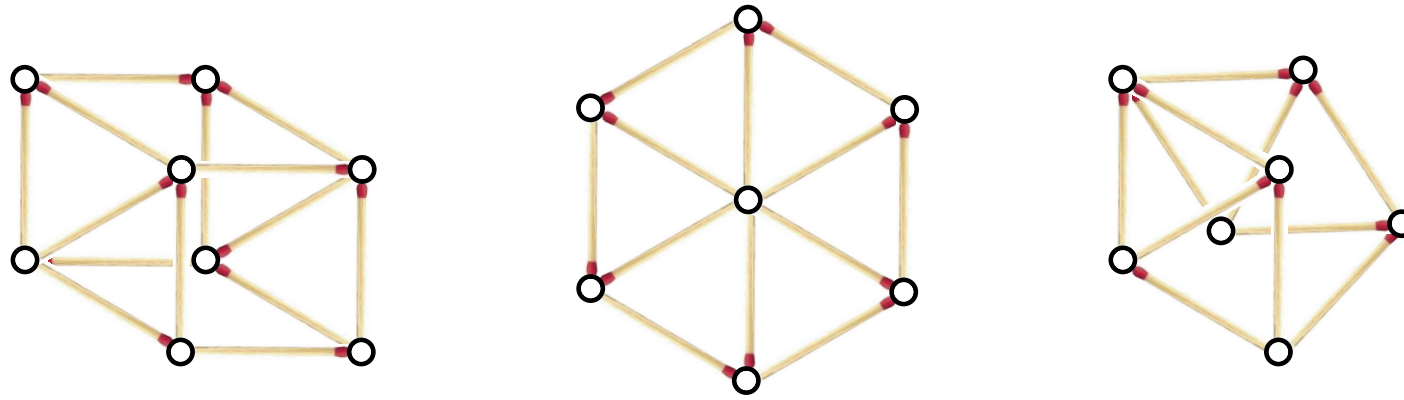
Output: Drawing of G which realizes all the edge lengths



Fixed Edge Lengths?

Input: Graph $G = (V, E)$, required edge length $\ell(e)$, $\forall e \in E$

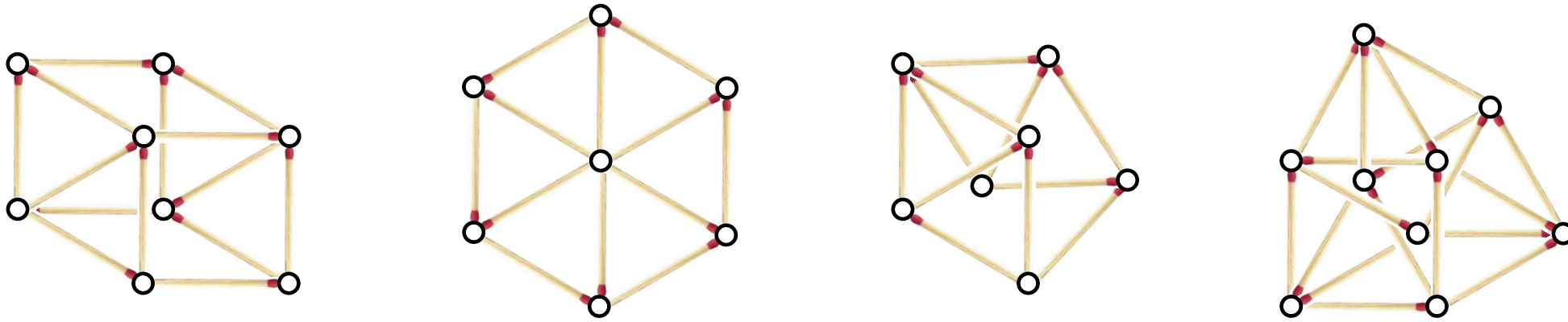
Output: Drawing of G which realizes all the edge lengths



Fixed Edge Lengths?

Input: Graph $G = (V, E)$, required edge length $\ell(e)$, $\forall e \in E$

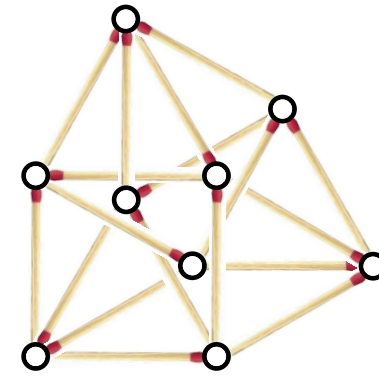
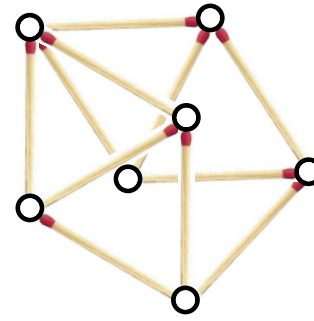
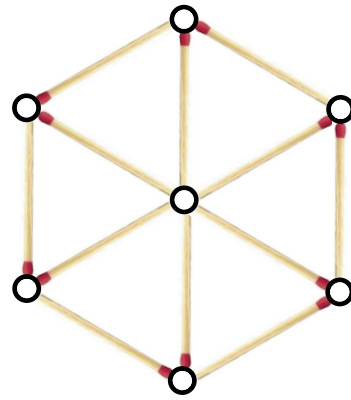
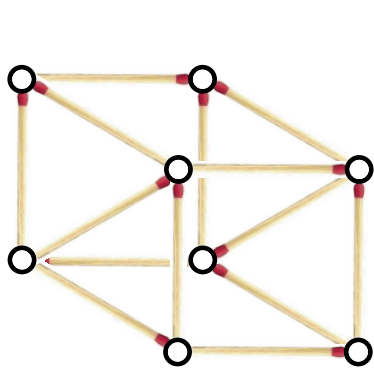
Output: Drawing of G which realizes all the edge lengths



Fixed Edge Lengths?

Input: Graph $G = (V, E)$, required edge length $\ell(e)$, $\forall e \in E$

Output: Drawing of G which realizes all the edge lengths

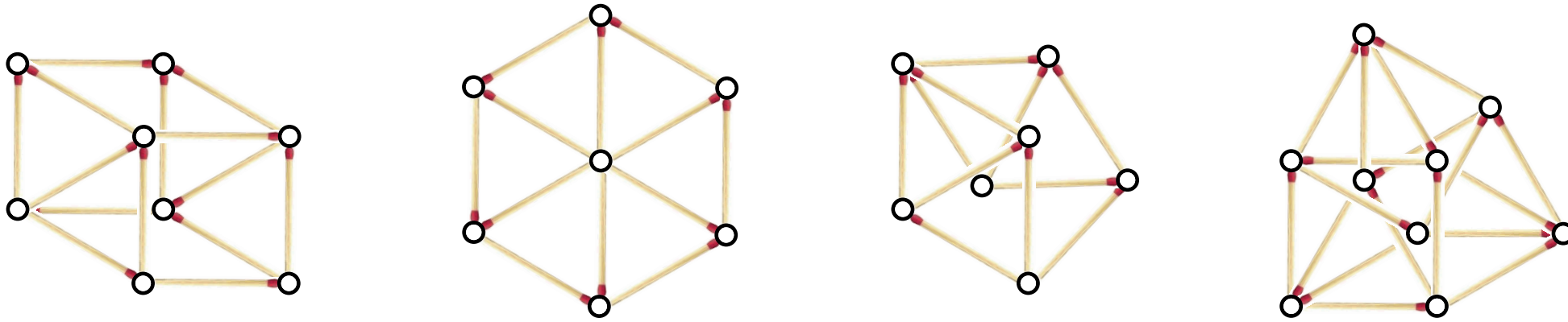


NP-hard for

Fixed Edge Lengths?

Input: Graph $G = (V, E)$, required edge length $\ell(e)$, $\forall e \in E$

Output: Drawing of G which realizes all the edge lengths



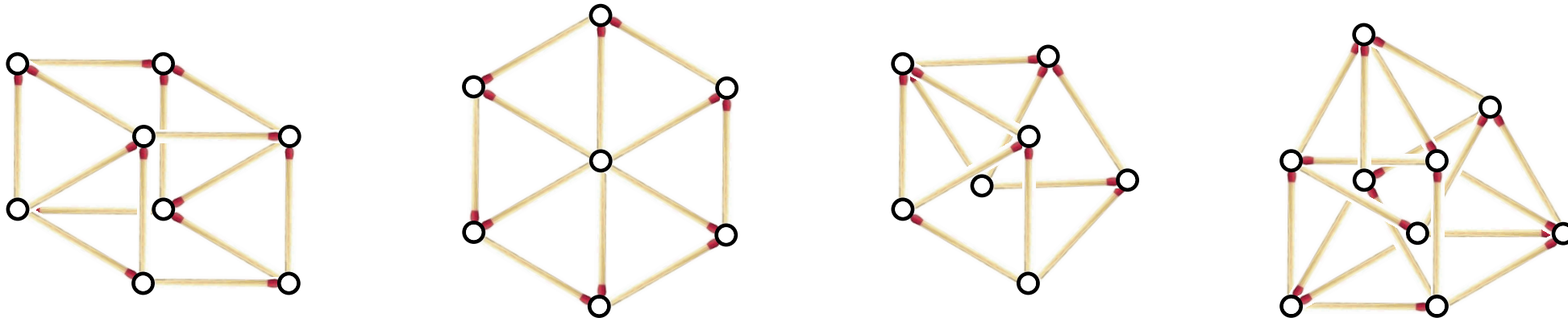
NP-hard for

- uniform edge lengths in any dimension [Johnson '82]

Fixed Edge Lengths?

Input: Graph $G = (V, E)$, required edge length $\ell(e)$, $\forall e \in E$

Output: Drawing of G which realizes all the edge lengths



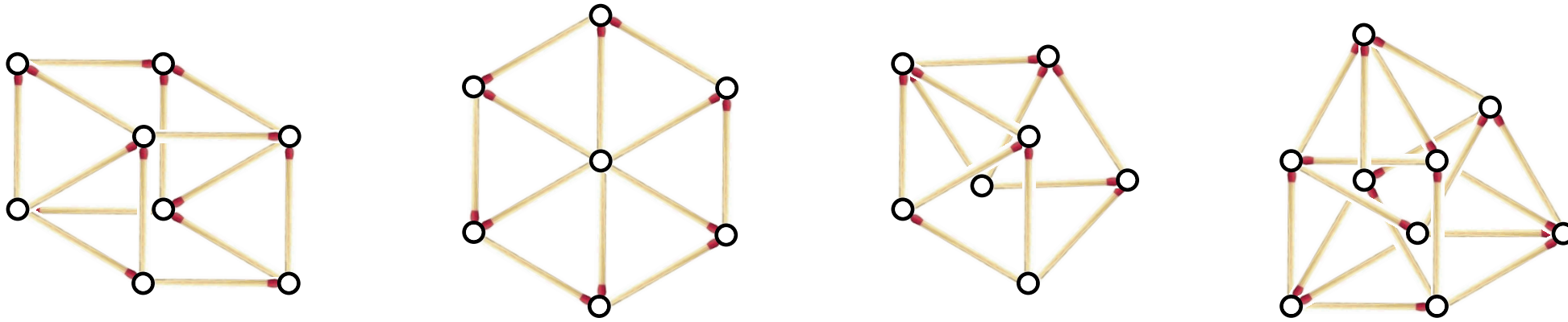
NP-hard for

- uniform edge lengths in any dimension [Johnson '82]
- uniform edge lengths in planar drawings [Eades, Wormald '90]

Fixed Edge Lengths?

Input: Graph $G = (V, E)$, required edge length $\ell(e)$, $\forall e \in E$

Output: Drawing of G which realizes all the edge lengths



NP-hard for

- uniform edge lengths in any dimension [Johnson '82]
- uniform edge lengths in planar drawings [Eades, Wormald '90]
- edge lengths $\{1, 2\}$ [Saxe '80]

Physical Analogy

Idea.

[Eades '84]

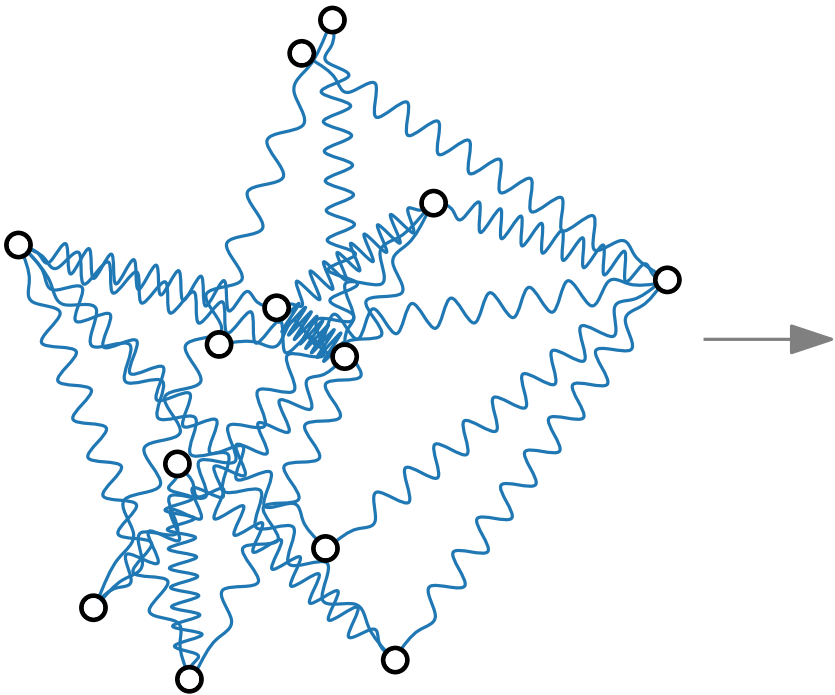
“To embed a graph we replace the vertices by steel rings and replace each edge with a **spring** to form a mechanical system .

Physical Analogy

Idea.

[Eades '84]

“To embed a graph we replace the vertices by steel rings and replace each edge with a **spring** to form a mechanical system .

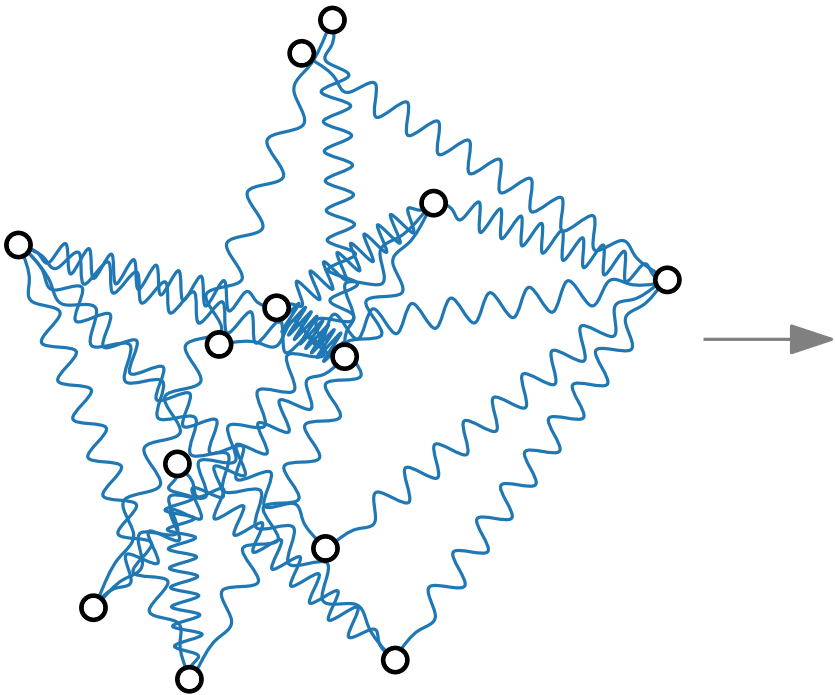


Physical Analogy

Idea.

[Eades '84]

“To embed a graph we replace the vertices by steel rings and replace each edge with a **spring** to form a mechanical system . . . The vertices are placed in some initial layout and let go so that the spring forces on the rings move the system to a minimal energy state.”

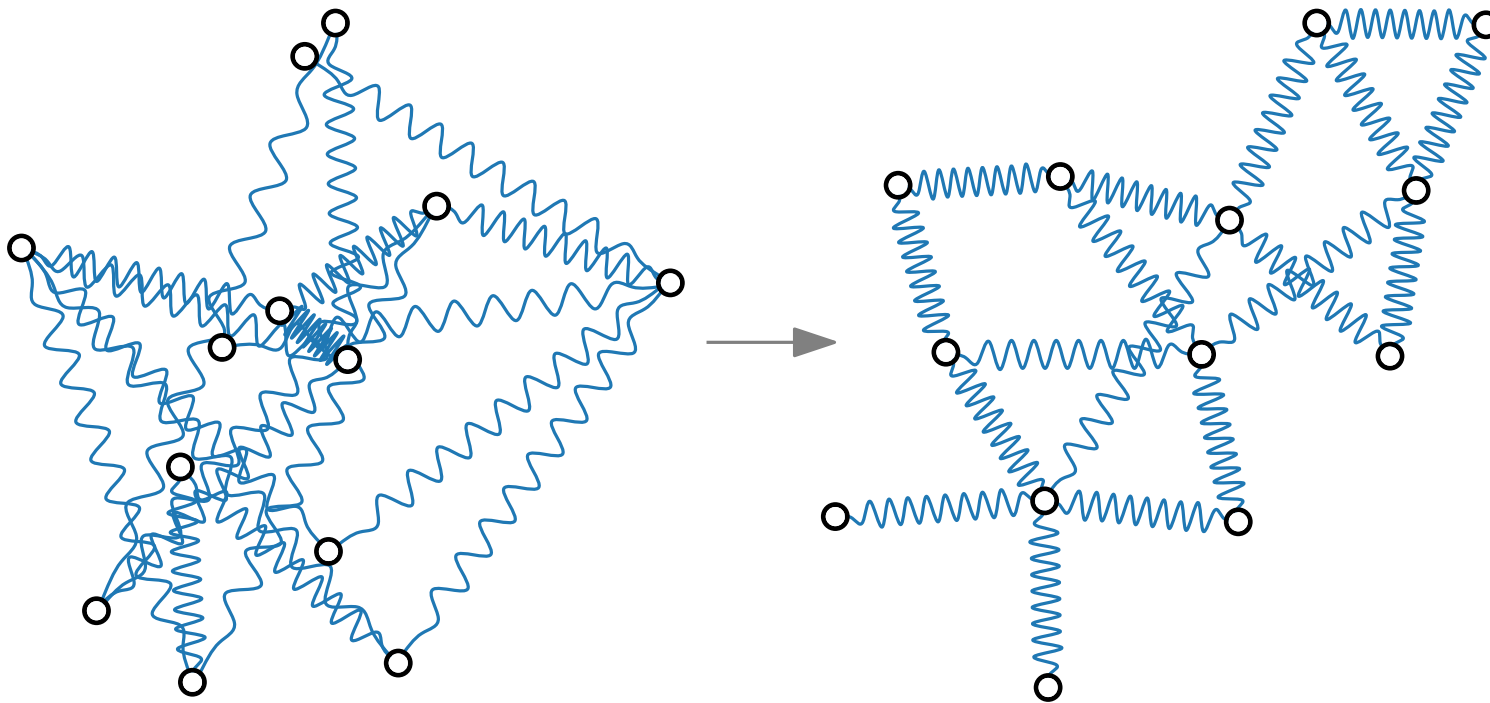


Physical Analogy

Idea.

[Eades '84]

“To embed a graph we replace the vertices by steel rings and replace each edge with a **spring** to form a mechanical system . . . The vertices are placed in some initial layout and let go so that the spring forces on the rings move the system to a minimal energy state.”

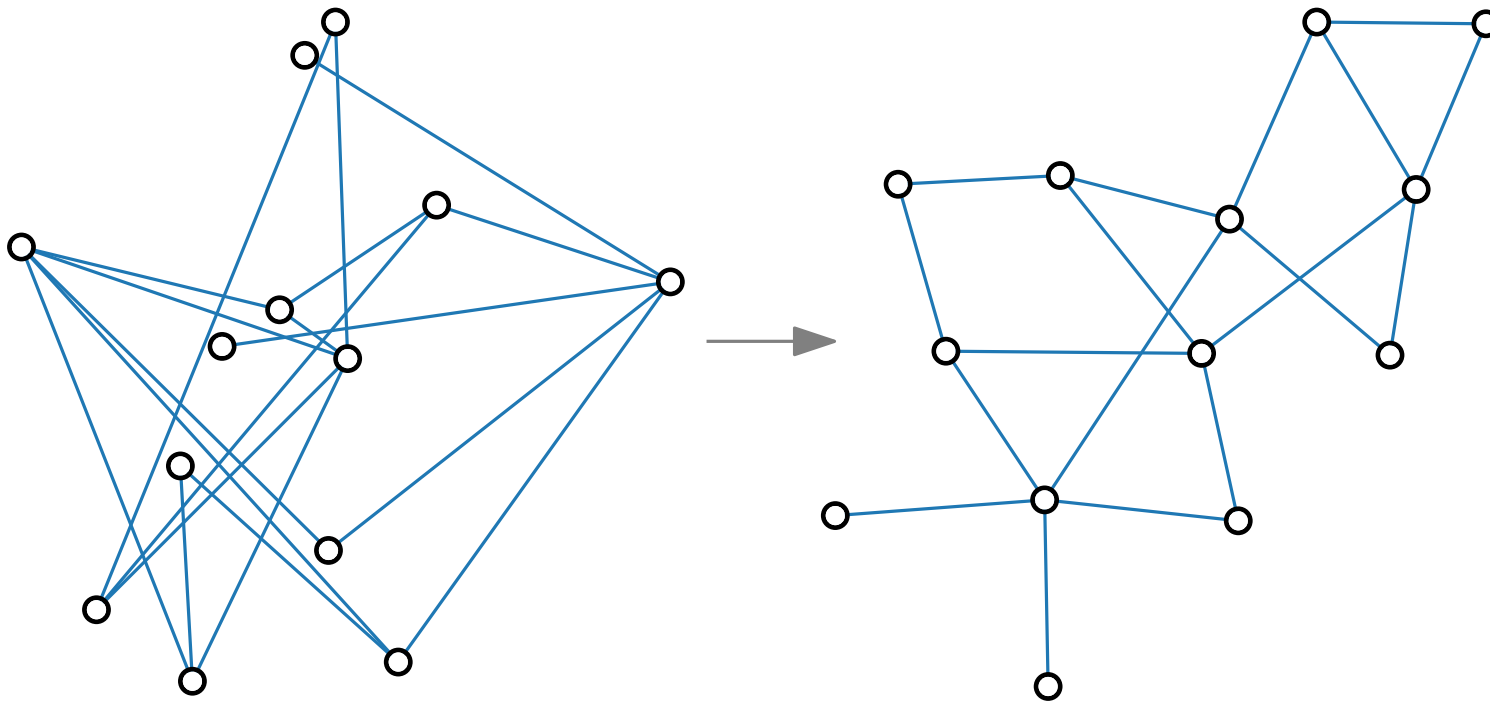


Physical Analogy

Idea.

[Eades '84]

“To embed a graph we replace the vertices by steel rings and replace each edge with a **spring** to form a mechanical system ... The vertices are placed in some initial layout and let go so that the spring forces on the rings move the system to a minimal energy state.”



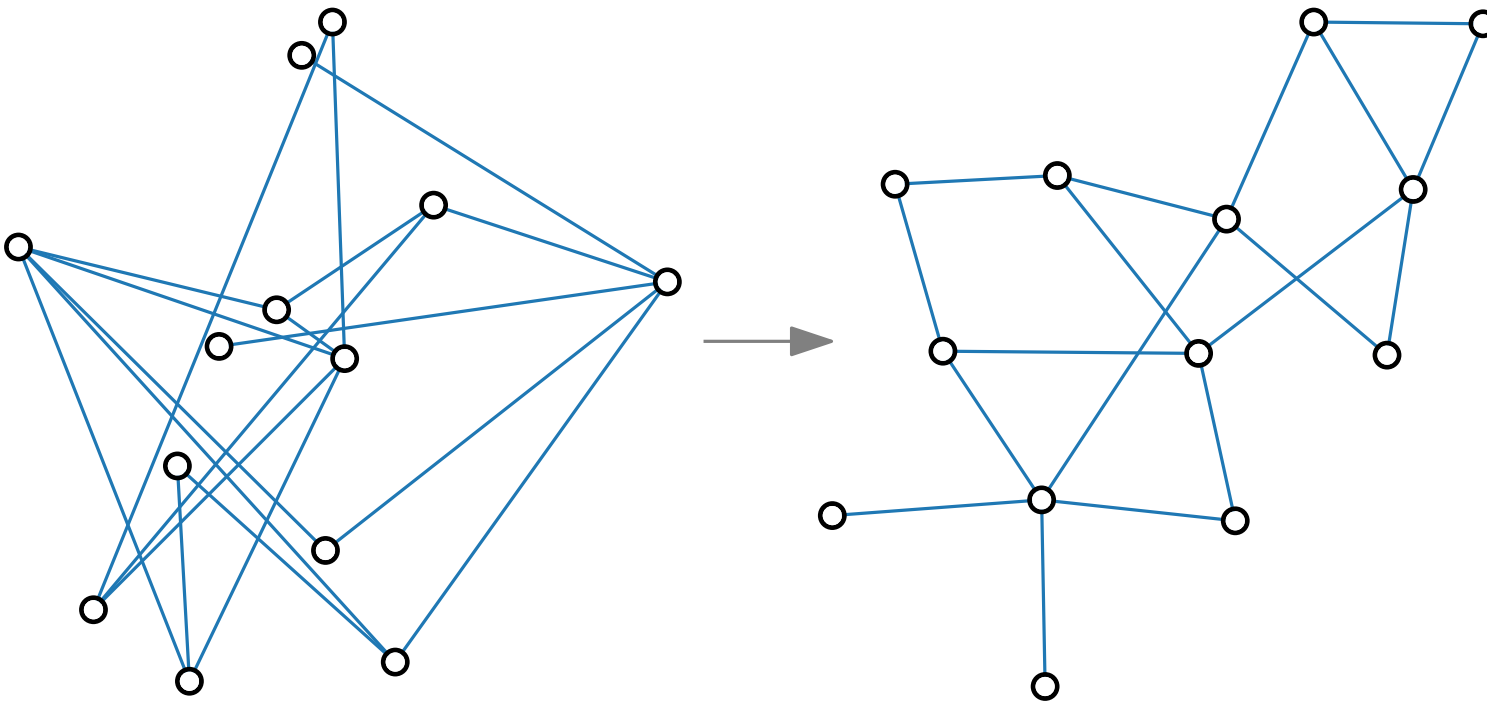
Physical Analogy

Idea.

[Eades '84]

“To embed a graph we replace the vertices by steel rings and replace each edge with a **spring** to form a mechanical system ... The vertices are placed in some initial layout and let go so that the spring forces on the rings move the system to a minimal energy state.”

Attractive forces.

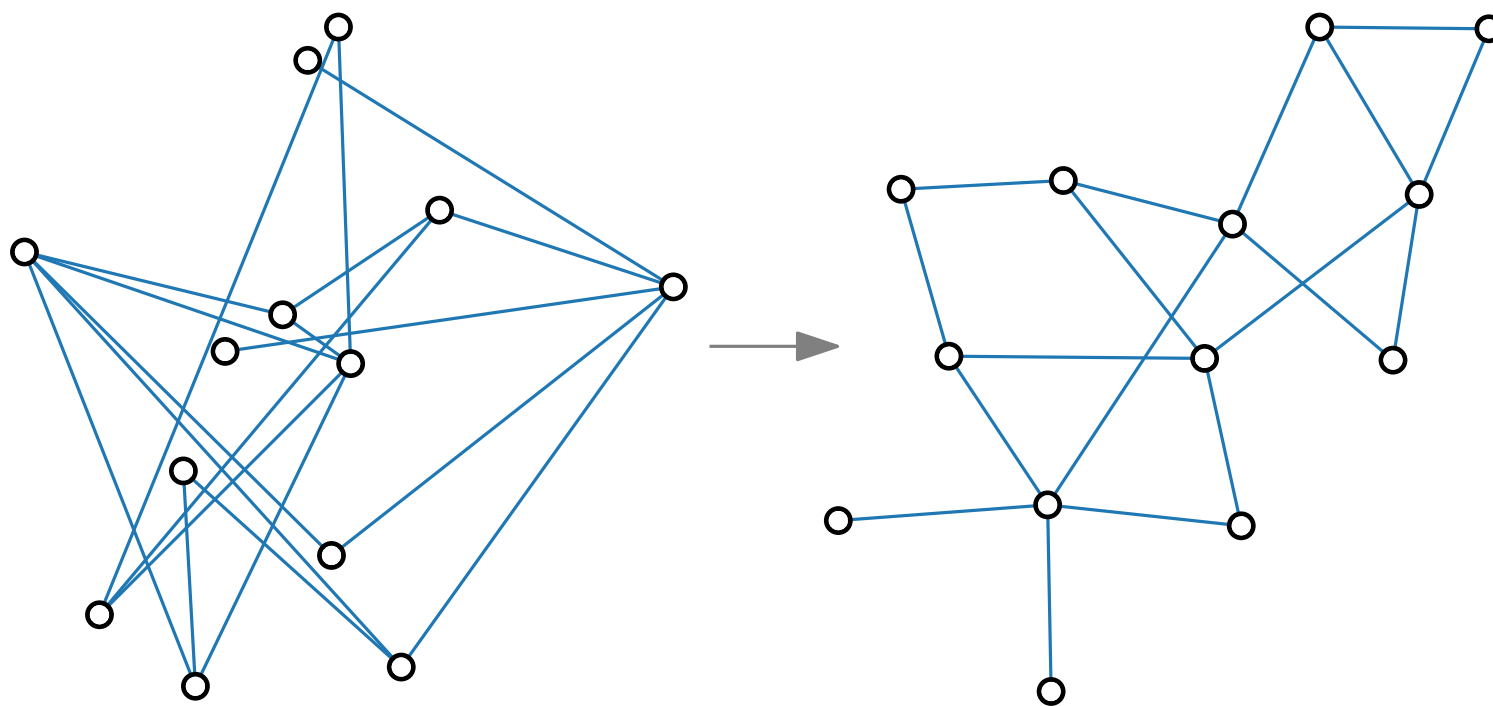


Physical Analogy

Idea.

[Eades '84]

“To embed a graph we replace the vertices by steel rings and replace each edge with a **spring** to form a mechanical system ... The vertices are placed in some initial layout and let go so that the spring forces on the rings move the system to a minimal energy state.”



Attractive forces.

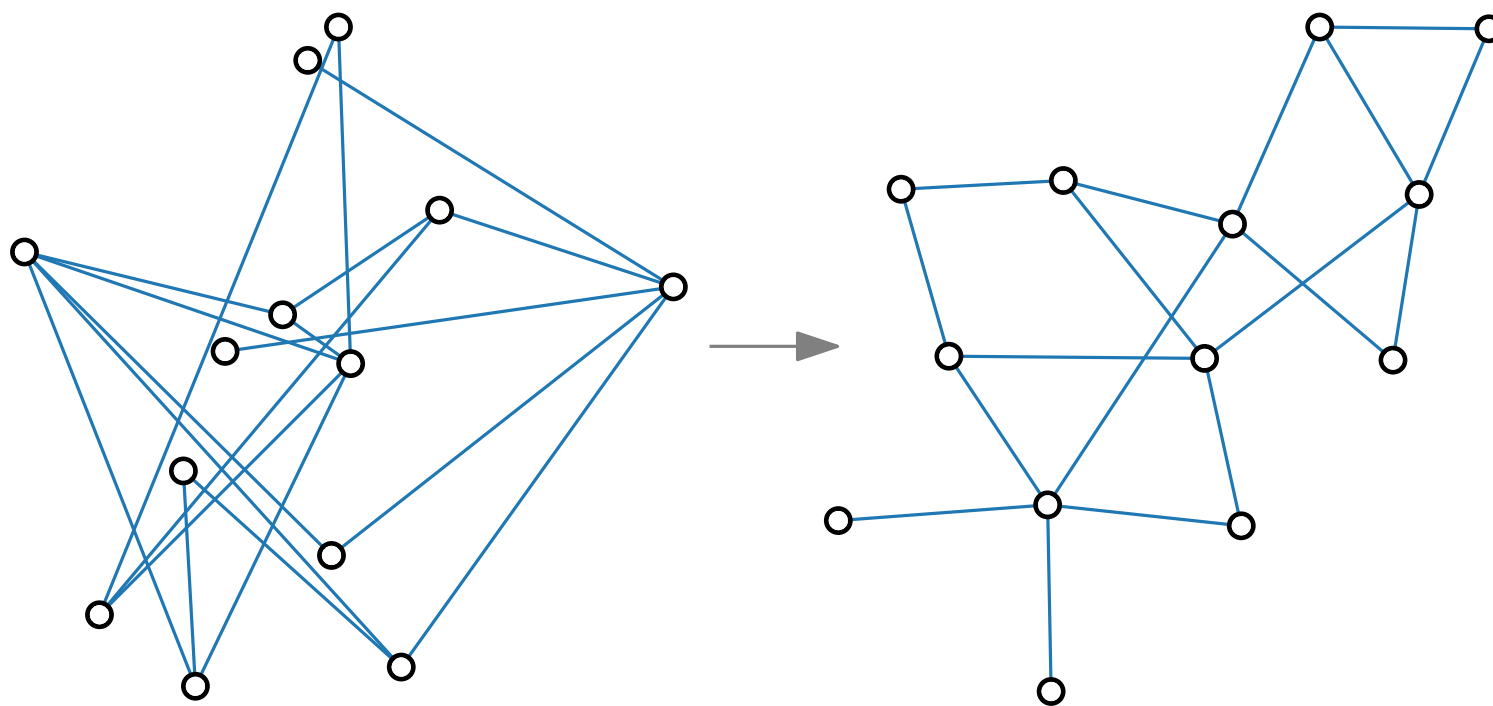
adjacent vertices u and v :

Physical Analogy

Idea.

[Eades '84]

“To embed a graph we replace the vertices by steel rings and replace each edge with a **spring** to form a mechanical system ... The vertices are placed in some initial layout and let go so that the spring forces on the rings move the system to a minimal energy state.”



Attractive forces.

adjacent vertices u and v :

$$u \circ \text{spring} \circ v$$

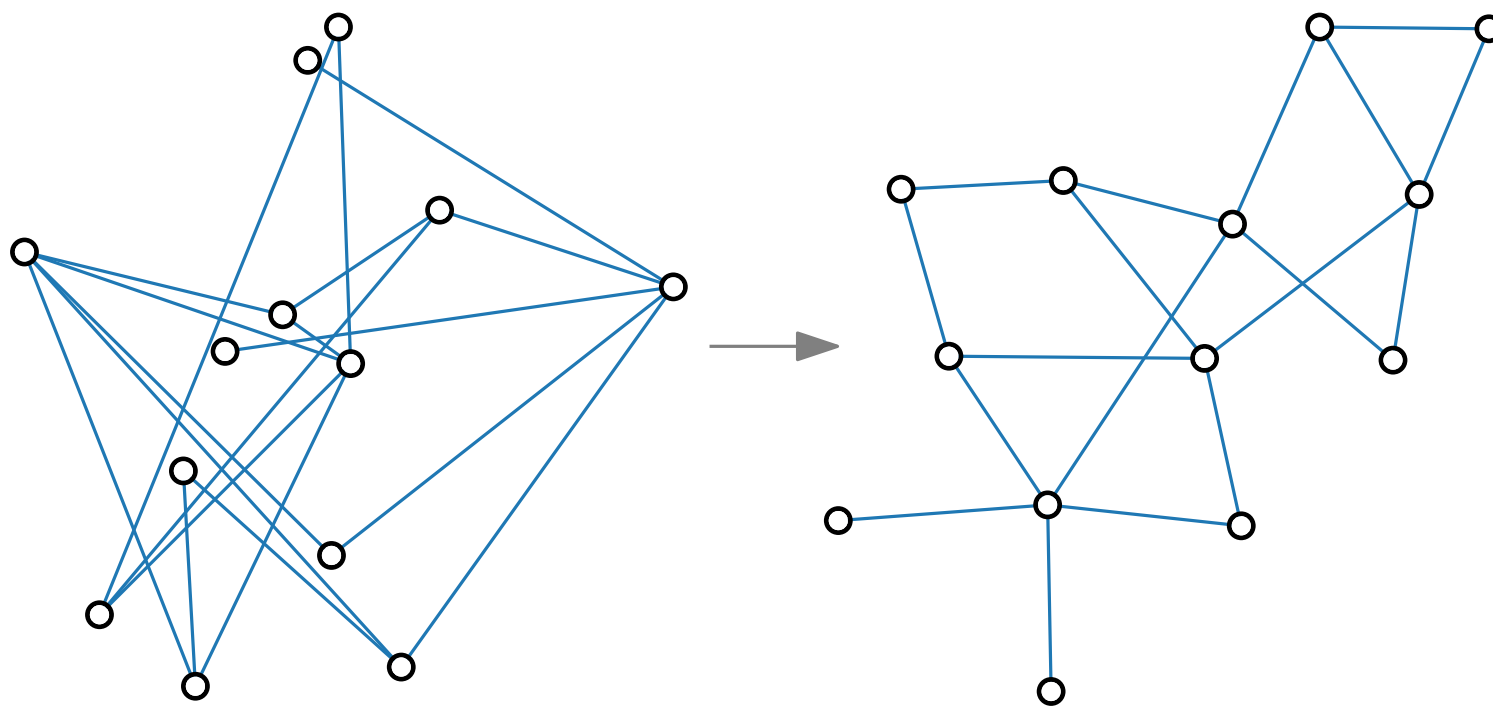
f_{attr}

Physical Analogy

Idea.

[Eades '84]

“To embed a graph we replace the vertices by steel rings and replace each edge with a **spring** to form a mechanical system ... The vertices are placed in some initial layout and let go so that the spring forces on the rings move the system to a minimal energy state.”



Attractive forces.

adjacent vertices u and v :

$$u \circ \text{spring} \circ v$$

f_{attr}

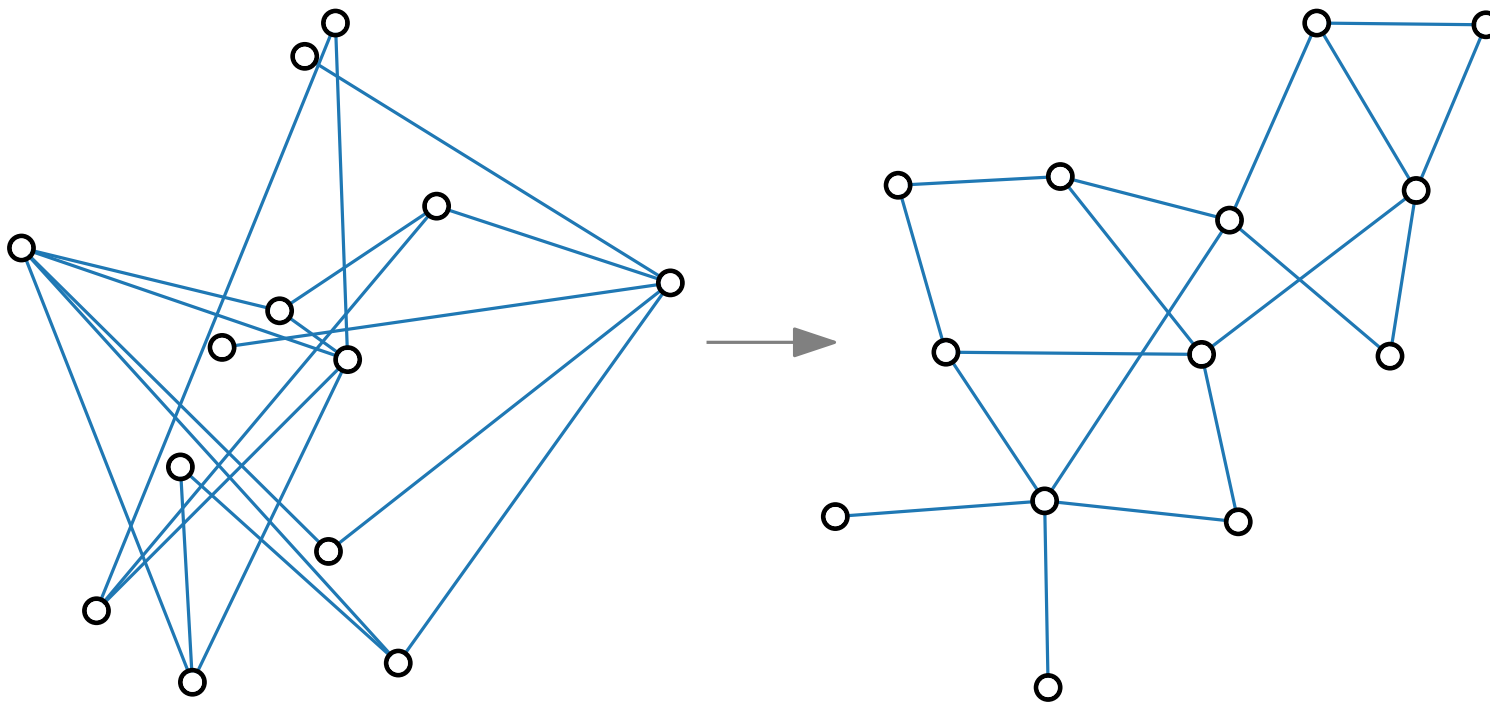
Repulsive forces.

Physical Analogy

Idea.

[Eades '84]

“To embed a graph we replace the vertices by steel rings and replace each edge with a **spring** to form a mechanical system ... The vertices are placed in some initial layout and let go so that the spring forces on the rings move the system to a minimal energy state.”



Attractive forces.

adjacent vertices u and v :

$$u \circ \text{spring} \circ v$$

f_{attr}

Repulsive forces.

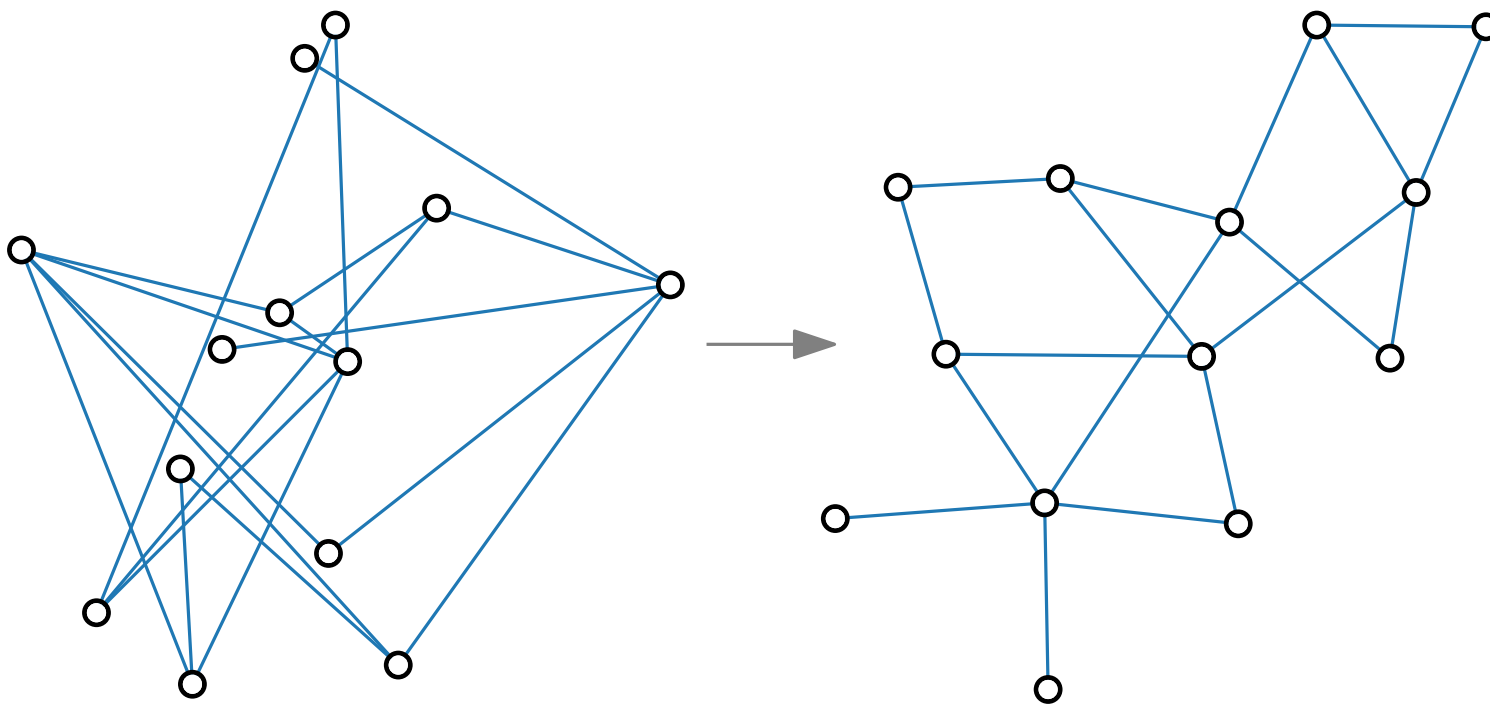
all vertices x and y :

Physical Analogy

Idea.

[Eades '84]

“To embed a graph we replace the vertices by steel rings and replace each edge with a **spring** to form a mechanical system ... The vertices are placed in some initial layout and let go so that the spring forces on the rings move the system to a minimal energy state.”



Attractive forces.

adjacent vertices u and v :



Repulsive forces.

all vertices x and y :



[Eades '84]

adjacent vertices u and v :

Repulsive forces.

all vertices x and y :

A diagram showing two nodes, x and y , connected by a red double-headed arrow. Below the arrow is the label f_{rep} .

So-called **spring embedders** or **force-directed** algorithms that work according to this or similar principles are among the most frequently used graph-drawing methods in practice.

Force-Directed Algorithms

$\text{ForceDirected}(G = (V, E), p = (p_v)_{v \in V}, \varepsilon > 0, K \in \mathbb{N})$

return p

Force-Directed Algorithms

initial layout

ForceDirected($G = (V, E)$, $p = (p_v)_{v \in V}$, $\varepsilon > 0$, $K \in \mathbb{N}$)

return p

Force-Directed Algorithms

initial layout

ForceDirected($G = (V, E)$, $p = (p_v)_{v \in V}$, $\varepsilon > 0$, $K \in \mathbb{N}$)

return p

end layout

Force-Directed Algorithms

ForceDirected($G = (V, E)$, $p = (p_v)_{v \in V}$, $\varepsilon > 0$, $K \in \mathbb{N}$)

return p

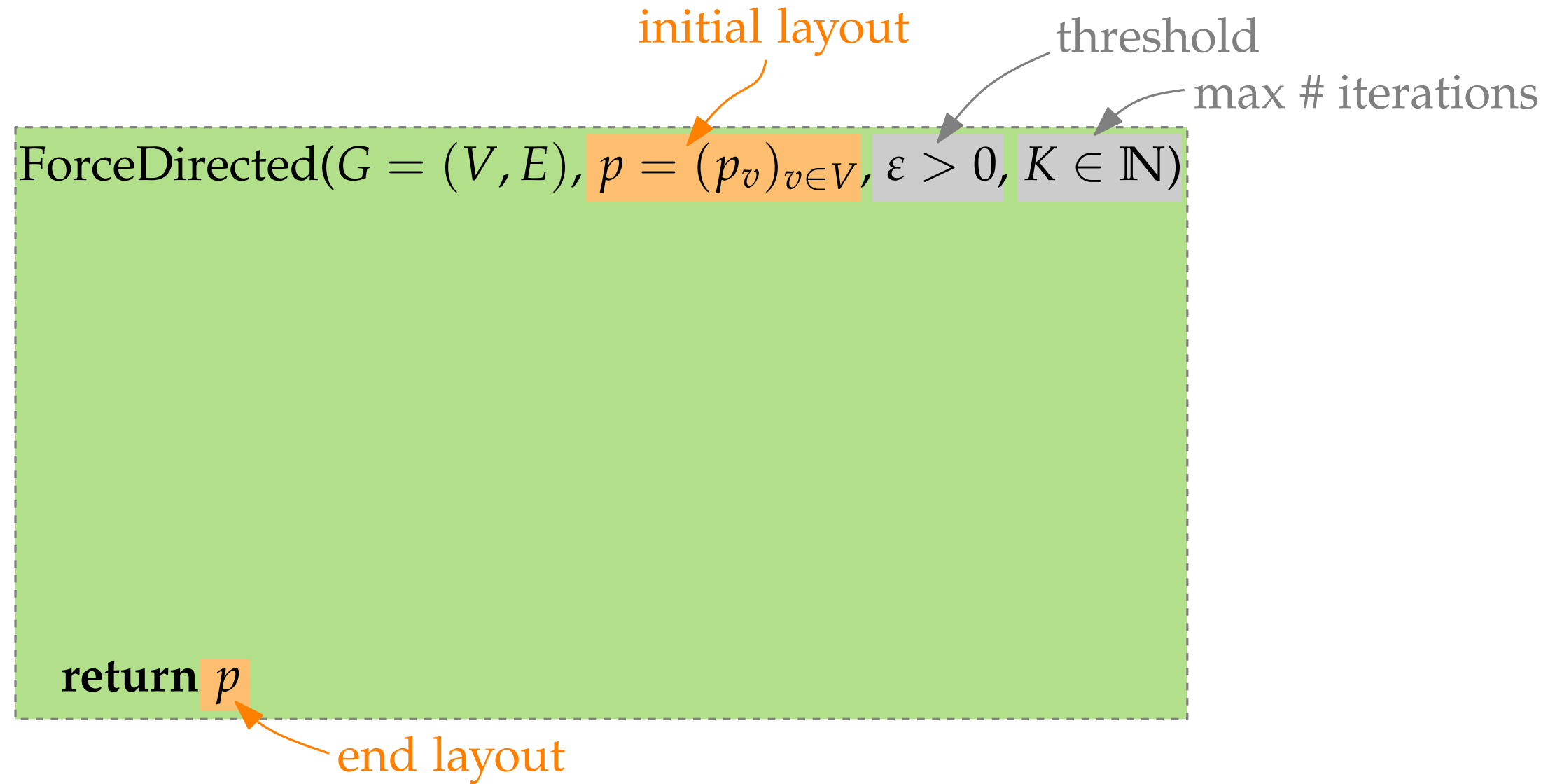
initial layout

threshold

end layout

The diagram shows the signature and return statement of the ForceDirected algorithm. The signature is ForceDirected(G = (V, E), p = (p_v)_{v in V}, ε > 0, K in N). The return statement is return p. Annotations include: 'initial layout' pointing to the initial position parameter p, 'threshold' pointing to the ε > 0 parameter, and 'end layout' pointing to the returned position p.

Force-Directed Algorithms



Force-Directed Algorithms

ForceDirected($G = (V, E)$, $p = (p_v)_{v \in V}$, $\varepsilon > 0$, $K \in \mathbb{N}$)

initial layout (points to p)

threshold (points to ε)

max # iterations (points to K)

```
 $t \leftarrow 1$   
while  $t < K$  and  $\max_{v \in V} \|F_v(t)\| > \varepsilon$  do  
     $t \leftarrow t + 1$   
return  $p$ 
```

end layout (points to p)

Force-Directed Algorithms

ForceDirected($G = (V, E)$, $p = (p_v)_{v \in V}$, $\varepsilon > 0$, $K \in \mathbb{N}$)

initial layout (points to p)

threshold (points to ε)

max # iterations (points to K)

```

 $t \leftarrow 1$ 
while  $t < K$  and  $\max_{v \in V} \|F_v(t)\| > \varepsilon$  do
  foreach  $u \in V$  do
     $\quad$ 
   $t \leftarrow t + 1$ 
return  $p$ 
end layout (points to  $p$ )
  
```

$u \circ$

Force-Directed Algorithms

ForceDirected($G = (V, E)$, $p = (p_v)_{v \in V}$, $\varepsilon > 0$, $K \in \mathbb{N}$)

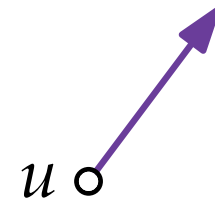
initial layout (points to p)

threshold (points to ε)

max # iterations (points to K)

```

 $t \leftarrow 1$ 
while  $t < K$  and  $\max_{v \in V} \|F_v(t)\| > \varepsilon$  do
  foreach  $u \in V$  do
     $F_u(t) \leftarrow$ 
   $t \leftarrow t + 1$ 
return  $p$ 
end layout (points to  $p$ )
  
```



Force-Directed Algorithms

ForceDirected($G = (V, E)$, $p = (p_v)_{v \in V}$, $\varepsilon > 0$, $K \in \mathbb{N}$)

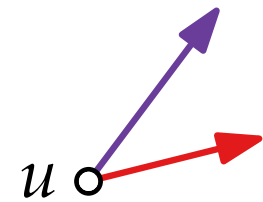
initial layout (points to p)

threshold (points to ε)

max # iterations (points to K)

```

 $t \leftarrow 1$ 
while  $t < K$  and  $\max_{v \in V} \|F_v(t)\| > \varepsilon$  do
  foreach  $u \in V$  do
     $F_u(t) \leftarrow \sum_{v \in V} f_{\text{rep}}(u, v) +$ 
    ...
   $t \leftarrow t + 1$ 
return  $p$ 
end layout (points to  $p$ )
  
```



Force-Directed Algorithms

ForceDirected($G = (V, E)$, $p = (p_v)_{v \in V}$, $\varepsilon > 0$, $K \in \mathbb{N}$)

initial layout → $p = (p_v)_{v \in V}$

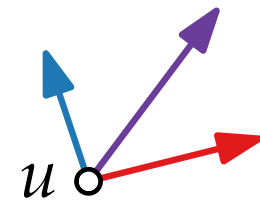
threshold → $\varepsilon > 0$

max # iterations → $K \in \mathbb{N}$

```

 $t \leftarrow 1$ 
while  $t < K$  and  $\max_{v \in V} \|F_v(t)\| > \varepsilon$  do
  foreach  $u \in V$  do
     $F_u(t) \leftarrow \sum_{v \in V} f_{\text{rep}}(u, v) + \sum_{uv \in E} f_{\text{attr}}(u, v)$ 
   $t \leftarrow t + 1$ 
return  $p$ 
  
```

end layout → p



Force-Directed Algorithms

ForceDirected($G = (V, E)$, $p = (p_v)_{v \in V}$, $\varepsilon > 0$, $K \in \mathbb{N}$)

initial layout → $p = (p_v)_{v \in V}$

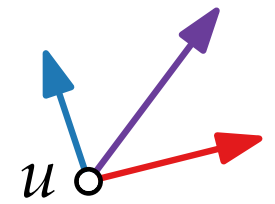
threshold → $\varepsilon > 0$

max # iterations → $K \in \mathbb{N}$

```

 $t \leftarrow 1$ 
while  $t < K$  and  $\max_{v \in V} \|F_v(t)\| > \varepsilon$  do
  foreach  $u \in V$  do
     $F_u(t) \leftarrow \sum_{v \in V} f_{\text{rep}}(u, v) + \sum_{uv \in E} f_{\text{attr}}(u, v)$ 
  foreach  $u \in V$  do
     $p_u \leftarrow p_u + \delta(t) \cdot F_u(t)$ 
   $t \leftarrow t + 1$ 
return  $p$ 
  
```

end layout → p



Force-Directed Algorithms

ForceDirected($G = (V, E)$, $p = (p_v)_{v \in V}$, $\varepsilon > 0$, $K \in \mathbb{N}$)

initial layout → p

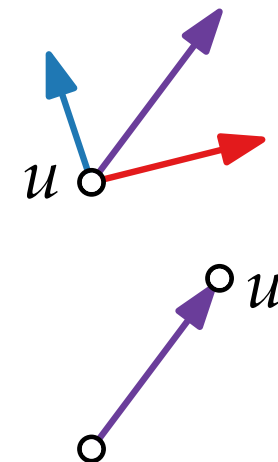
threshold → ε

max # iterations → K

```

 $t \leftarrow 1$ 
while  $t < K$  and  $\max_{v \in V} \|F_v(t)\| > \varepsilon$  do
  foreach  $u \in V$  do
     $F_u(t) \leftarrow \sum_{v \in V} f_{\text{rep}}(u, v) + \sum_{uv \in E} f_{\text{attr}}(u, v)$ 
  foreach  $u \in V$  do
     $p_u \leftarrow p_u + \delta(t) \cdot F_u(t)$ 
   $t \leftarrow t + 1$ 
return  $p$ 
  
```

end layout → p



Force-Directed Algorithms

ForceDirected($G = (V, E)$, $p = (p_v)_{v \in V}$, $\varepsilon > 0$, $K \in \mathbb{N}$)

initial layout → p

threshold → ε

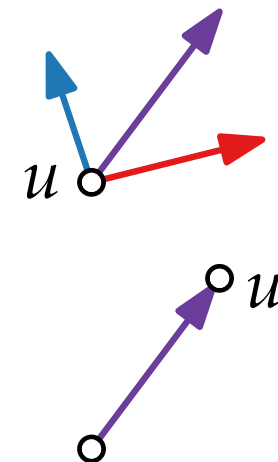
max # iterations → K

```

 $t \leftarrow 1$ 
while  $t < K$  and  $\max_{v \in V} \|F_v(t)\| > \varepsilon$  do
  foreach  $u \in V$  do
     $F_u(t) \leftarrow \sum_{v \in V} f_{\text{rep}}(u, v) + \sum_{uv \in E} f_{\text{attr}}(u, v)$ 
  foreach  $u \in V$  do
     $p_u \leftarrow p_u + \delta(t) \cdot F_u(t)$ 
   $t \leftarrow t + 1$ 
return  $p$ 
  
```

cooling factor → $\delta(t)$

end layout → p




$$\text{ForceDirected}(G = (V, E), p = (p_v)_{v \in V}, \varepsilon > 0, K \in \mathbb{N})$$

while $t < K$ **and** $\max_{v \in V} \|F_v(t)\| > \varepsilon$ **do**

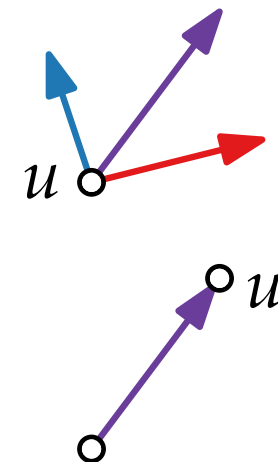
$$\lfloor F_u(t) \leftarrow \sum_{v \in V} f_{\text{rep}}(u, v) + \sum_{uv \in E} f_{\text{attr}}(u, v)$$
$$p_u \leftarrow p_u + \delta(t) \cdot F_u(t)$$

- cooling factor



A graph showing the function $\delta(t)$ on the vertical axis versus t on the horizontal axis. The curve starts at a high value on the vertical axis and decays exponentially towards the horizontal axis as t increases.

end layout



Force-Directed Algorithms

ForceDirected($G = (V, E)$, $p = (p_v)_{v \in V}$, $\varepsilon > 0$, $K \in \mathbb{N}$)

initial layout → $p = (p_v)_{v \in V}$

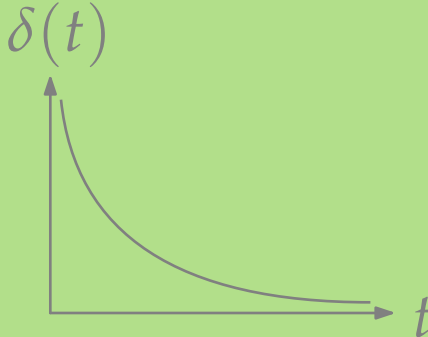
threshold → $\varepsilon > 0$

max # iterations → $K \in \mathbb{N}$

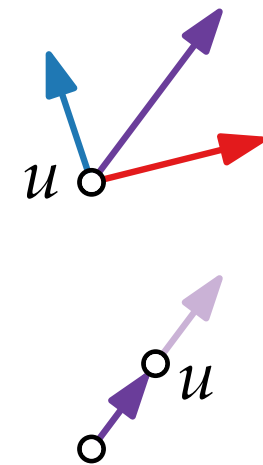
```

 $t \leftarrow 1$ 
while  $t < K$  and  $\max_{v \in V} \|F_v(t)\| > \varepsilon$  do
  foreach  $u \in V$  do
     $F_u(t) \leftarrow \sum_{v \in V} f_{\text{rep}}(u, v) + \sum_{uv \in E} f_{\text{attr}}(u, v)$ 
  foreach  $u \in V$  do
     $p_u \leftarrow p_u + \delta(t) \cdot F_u(t)$ 
     $t \leftarrow t + 1$ 
return  $p$ 
  
```

cooling factor → $\delta(t)$



end layout → p



Spring Embedder by Eades – Model

```

ForceDirected( $G = (V, E)$ ,  $p = (p_v)_{v \in V}$ ,  $\varepsilon > 0$ ,  $K \in \mathbb{N}$ )
   $t \leftarrow 1$ 
  while  $t < K$  and  $\max_{v \in V} \|F_v(t)\| > \varepsilon$  do
    foreach  $u \in V$  do
       $F_u(t) \leftarrow \sum_{v \in V} f_{\text{rep}}(u, v) + \sum_{uv \in E} f_{\text{attr}}(u, v)$ 
    foreach  $u \in V$  do
       $p_u \leftarrow p_u + \delta(t) \cdot F_u(t)$ 
     $t \leftarrow t + 1$ 
  return  $p$ 

```


Spring Embedder by Eades – Model

- Repulsive forces

- Attractive forces

- Resulting displacement vector

```
ForceDirected( $G = (V, E)$ ,  $p = (p_v)_{v \in V}$ ,  $\varepsilon > 0$ ,  $K \in \mathbb{N}$ )  
   $t \leftarrow 1$   
  while  $t < K$  and  $\max_{v \in V} \|F_v(t)\| > \varepsilon$  do  
    foreach  $u \in V$  do  
       $F_u(t) \leftarrow \sum_{v \in V} f_{\text{rep}}(u, v) + \sum_{uv \in E} f_{\text{attr}}(u, v)$   
    foreach  $u \in V$  do  
       $p_u \leftarrow p_u + \delta(t) \cdot F_u(t)$   
     $t \leftarrow t + 1$   
  return  $p$ 
```

Spring Embedder by Eades – Model

■ Repulsive forces

■ Attractive forces

■ Resulting displacement vector

$$F_u = \sum_{v \in V} f_{\text{rep}}(u, v) + \sum_{uv \in E} f_{\text{attr}}(u, v)$$

```

ForceDirected( $G = (V, E)$ ,  $p = (p_v)_{v \in V}$ ,  $\varepsilon > 0$ ,  $K \in \mathbb{N}$ )
 $t \leftarrow 1$ 
while  $t < K$  and  $\max_{v \in V} \|F_v(t)\| > \varepsilon$  do
    foreach  $u \in V$  do
         $F_u(t) \leftarrow \sum_{v \in V} f_{\text{rep}}(u, v) + \sum_{uv \in E} f_{\text{attr}}(u, v)$ 
    foreach  $u \in V$  do
         $p_u \leftarrow p_u + \delta(t) \cdot F_u(t)$ 
     $t \leftarrow t + 1$ 
return  $p$ 
  
```

Spring Embedder by Eades – Model

■ Repulsive forces

$$f_{\text{rep}}(u, v) = \frac{c_{\text{rep}}}{\|p_v - p_u\|^2} \cdot \overrightarrow{p_v p_u}$$

■ Attractive forces

■ Resulting displacement vector

$$F_u = \sum_{v \in V} f_{\text{rep}}(u, v) + \sum_{uv \in E} f_{\text{attr}}(u, v)$$

```

ForceDirected( $G = (V, E)$ ,  $p = (p_v)_{v \in V}$ ,  $\varepsilon > 0$ ,  $K \in \mathbb{N}$ )
 $t \leftarrow 1$ 
while  $t < K$  and  $\max_{v \in V} \|F_v(t)\| > \varepsilon$  do
    foreach  $u \in V$  do
         $F_u(t) \leftarrow \sum_{v \in V} f_{\text{rep}}(u, v) + \sum_{uv \in E} f_{\text{attr}}(u, v)$ 
    foreach  $u \in V$  do
         $p_u \leftarrow p_u + \delta(t) \cdot F_u(t)$ 
     $t \leftarrow t + 1$ 
return  $p$ 

```

Spring Embedder by Eades – Model

■ Repulsive forces

$$f_{\text{rep}}(u, v) = \frac{c_{\text{rep}}}{||p_v - p_u||^2} \cdot \overrightarrow{p_v p_u}$$

■ Attractive forces

■ Resulting displacement vector

$$F_u = \sum_{v \in V} f_{\text{rep}}(u, v) + \sum_{uv \in E} f_{\text{attr}}(u, v)$$

```

ForceDirected( $G = (V, E)$ ,  $p = (p_v)_{v \in V}$ ,  $\varepsilon > 0$ ,  $K \in \mathbb{N}$ )
 $t \leftarrow 1$ 
while  $t < K$  and  $\max_{v \in V} ||F_v(t)|| > \varepsilon$  do
    foreach  $u \in V$  do
         $F_u(t) \leftarrow \sum_{v \in V} f_{\text{rep}}(u, v) + \sum_{uv \in E} f_{\text{attr}}(u, v)$ 
    foreach  $u \in V$  do
         $p_u \leftarrow p_u + \delta(t) \cdot F_u(t)$ 
     $t \leftarrow t + 1$ 
return  $p$ 

```

Notation.

- $||p_u - p_v||$ = Euclidean distance between u and v

Spring Embedder by Eades – Model

■ Repulsive forces

$$f_{\text{rep}}(u, v) = \frac{c_{\text{rep}}}{\|p_v - p_u\|^2} \cdot \overrightarrow{p_v p_u}$$

■ Attractive forces

■ Resulting displacement vector

$$F_u = \sum_{v \in V} f_{\text{rep}}(u, v) + \sum_{uv \in E} f_{\text{attr}}(u, v)$$

```

ForceDirected( $G = (V, E)$ ,  $p = (p_v)_{v \in V}$ ,  $\varepsilon > 0$ ,  $K \in \mathbb{N}$ )
 $t \leftarrow 1$ 
while  $t < K$  and  $\max_{v \in V} \|F_v(t)\| > \varepsilon$  do
  foreach  $u \in V$  do
     $F_u(t) \leftarrow \sum_{v \in V} f_{\text{rep}}(u, v) + \sum_{uv \in E} f_{\text{attr}}(u, v)$ 
  foreach  $u \in V$  do
     $p_u \leftarrow p_u + \delta(t) \cdot F_u(t)$ 
   $t \leftarrow t + 1$ 
return  $p$ 

```

Notation.

- $\|p_u - p_v\|$ = Euclidean distance between u and v
- $\overrightarrow{p_u p_v}$ = unit vector pointing from u to v

Spring Embedder by Eades – Model

■ Repulsive forces

repulsion constant (e.g. 2.0)

$$f_{\text{rep}}(u, v) = \frac{c_{\text{rep}}}{\|p_v - p_u\|^2} \cdot \overrightarrow{p_v p_u}$$

■ Attractive forces

■ Resulting displacement vector

$$F_u = \sum_{v \in V} f_{\text{rep}}(u, v) + \sum_{uv \in E} f_{\text{attr}}(u, v)$$

```
ForceDirected( $G = (V, E)$ ,  $p = (p_v)_{v \in V}$ ,  $\varepsilon > 0$ ,  $K \in \mathbb{N}$ )
 $t \leftarrow 1$ 
while  $t < K$  and  $\max_{v \in V} \|F_v(t)\| > \varepsilon$  do
  foreach  $u \in V$  do
     $F_u(t) \leftarrow \sum_{v \in V} f_{\text{rep}}(u, v) + \sum_{uv \in E} f_{\text{attr}}(u, v)$ 
  foreach  $u \in V$  do
     $p_u \leftarrow p_u + \delta(t) \cdot F_u(t)$ 
   $t \leftarrow t + 1$ 
return  $p$ 
```

Notation.

- $\|p_u - p_v\|$ = Euclidean distance between u and v
- $\overrightarrow{p_u p_v}$ = unit vector pointing from u to v

Spring Embedder by Eades – Model

■ Repulsive forces

repulsion constant (e.g. 2.0)

$$f_{\text{rep}}(u, v) = \frac{c_{\text{rep}}}{\|p_v - p_u\|^2} \cdot \overrightarrow{p_v p_u}$$

■ Attractive forces

$$f_{\text{spring}}(u, v) = c_{\text{spring}} \cdot \log \frac{\|p_v - p_u\|}{\ell} \cdot \overrightarrow{p_u p_v}$$

■ Resulting displacement vector

$$F_u = \sum_{v \in V} f_{\text{rep}}(u, v) + \sum_{uv \in E} f_{\text{attr}}(u, v)$$

```
ForceDirected( $G = (V, E)$ ,  $p = (p_v)_{v \in V}$ ,  $\varepsilon > 0$ ,  $K \in \mathbb{N}$ )
 $t \leftarrow 1$ 
while  $t < K$  and  $\max_{v \in V} \|F_v(t)\| > \varepsilon$  do
  foreach  $u \in V$  do
     $F_u(t) \leftarrow \sum_{v \in V} f_{\text{rep}}(u, v) + \sum_{uv \in E} f_{\text{attr}}(u, v)$ 
  foreach  $u \in V$  do
     $p_u \leftarrow p_u + \delta(t) \cdot F_u(t)$ 
   $t \leftarrow t + 1$ 
return  $p$ 
```

Notation.

- $\|p_u - p_v\|$ = Euclidean distance between u and v
- $\overrightarrow{p_u p_v}$ = unit vector pointing from u to v

Spring Embedder by Eades – Model

■ Repulsive forces

repulsion constant (e.g. 2.0)

$$f_{\text{rep}}(u, v) = \frac{c_{\text{rep}}}{\|p_v - p_u\|^2} \cdot \overrightarrow{p_v p_u}$$

■ Attractive forces

$$f_{\text{spring}}(u, v) = c_{\text{spring}} \cdot \log \frac{\|p_v - p_u\|}{\ell} \cdot \overrightarrow{p_u p_v}$$

■ Resulting displacement vector

$$F_u = \sum_{v \in V} f_{\text{rep}}(u, v) + \sum_{uv \in E} f_{\text{attr}}(u, v)$$

```
ForceDirected( $G = (V, E)$ ,  $p = (p_v)_{v \in V}$ ,  $\varepsilon > 0$ ,  $K \in \mathbb{N}$ )
 $t \leftarrow 1$ 
while  $t < K$  and  $\max_{v \in V} \|F_v(t)\| > \varepsilon$  do
  foreach  $u \in V$  do
     $F_u(t) \leftarrow \sum_{v \in V} f_{\text{rep}}(u, v) + \sum_{uv \in E} f_{\text{attr}}(u, v)$ 
  foreach  $u \in V$  do
     $p_u \leftarrow p_u + \delta(t) \cdot F_u(t)$ 
   $t \leftarrow t + 1$ 
return  $p$ 
```

Notation.

- $\|p_u - p_v\|$ = Euclidean distance between u and v
- $\overrightarrow{p_u p_v}$ = unit vector pointing from u to v
- ℓ = ideal spring length for edges

Spring Embedder by Eades – Model

■ Repulsive forces

repulsion constant (e.g. 2.0)

$$f_{\text{rep}}(u, v) = \frac{c_{\text{rep}}}{\|p_v - p_u\|^2} \cdot \overrightarrow{p_v p_u}$$

■ Attractive forces

spring constant (e.g. 1.0)

$$f_{\text{spring}}(u, v) = c_{\text{spring}} \cdot \log \frac{\|p_v - p_u\|}{\ell} \cdot \overrightarrow{p_u p_v}$$

■ Resulting displacement vector

$$F_u = \sum_{v \in V} f_{\text{rep}}(u, v) + \sum_{uv \in E} f_{\text{attr}}(u, v)$$

```
ForceDirected( $G = (V, E)$ ,  $p = (p_v)_{v \in V}$ ,  $\varepsilon > 0$ ,  $K \in \mathbb{N}$ )
 $t \leftarrow 1$ 
while  $t < K$  and  $\max_{v \in V} \|F_v(t)\| > \varepsilon$  do
  foreach  $u \in V$  do
     $F_u(t) \leftarrow \sum_{v \in V} f_{\text{rep}}(u, v) + \sum_{uv \in E} f_{\text{attr}}(u, v)$ 
  foreach  $u \in V$  do
     $p_u \leftarrow p_u + \delta(t) \cdot F_u(t)$ 
   $t \leftarrow t + 1$ 
return  $p$ 
```

Notation.

- $\|p_u - p_v\|$ = Euclidean distance between u and v
- $\overrightarrow{p_u p_v}$ = unit vector pointing from u to v
- ℓ = ideal spring length for edges

Spring Embedder by Eades – Model

■ Repulsive forces

repulsion constant (e.g. 2.0)

$$f_{\text{rep}}(u, v) = \frac{c_{\text{rep}}}{\|p_v - p_u\|^2} \cdot \overrightarrow{p_v p_u}$$

■ Attractive forces

spring constant (e.g. 1.0)

$$f_{\text{spring}}(u, v) = c_{\text{spring}} \cdot \log \frac{\|p_v - p_u\|}{\ell} \cdot \overrightarrow{p_u p_v}$$

$$f_{\text{attr}}(u, v) = f_{\text{spring}}(u, v) - f_{\text{rep}}(u, v)$$

■ Resulting displacement vector

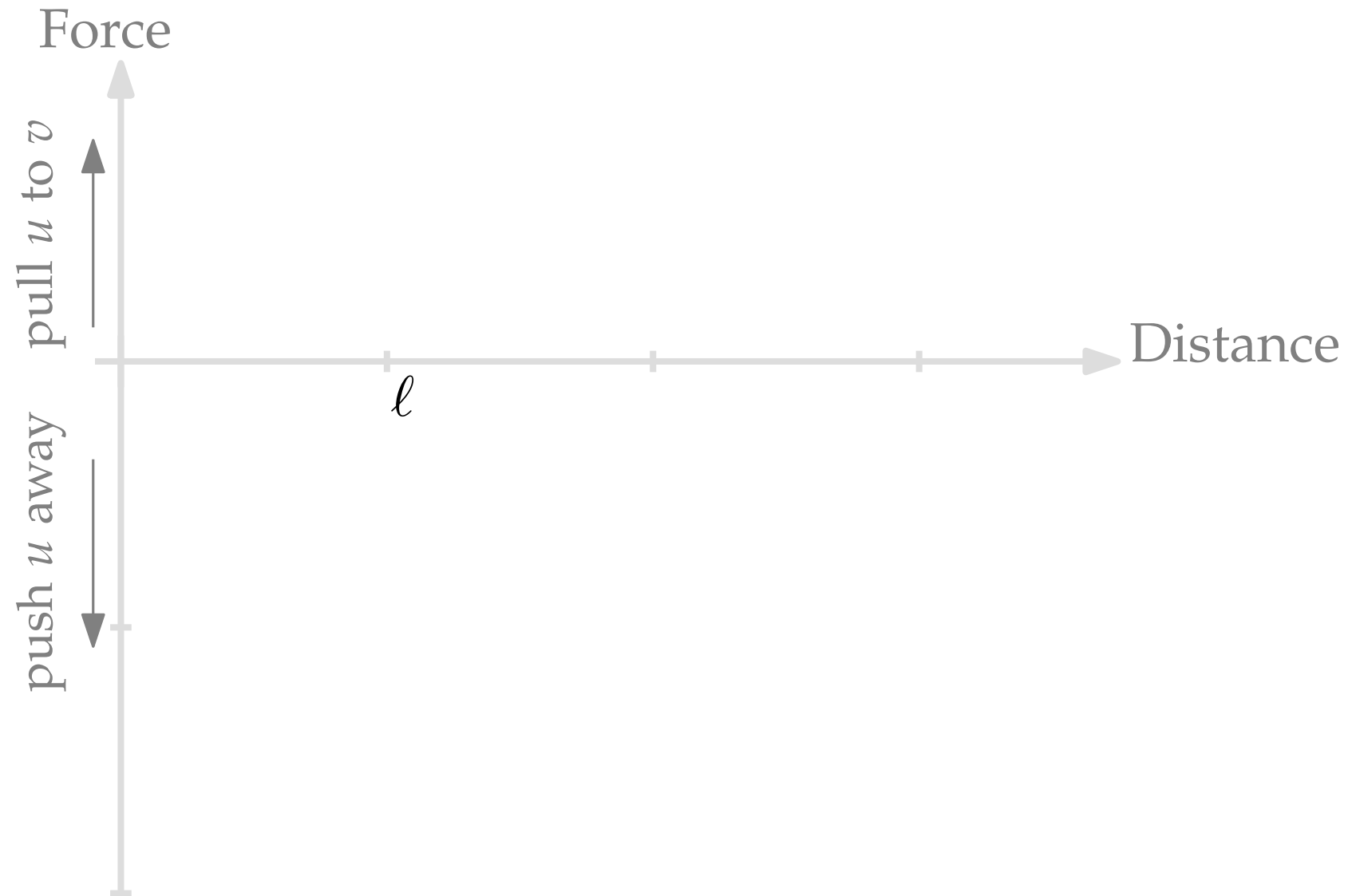
$$F_u = \sum_{v \in V} f_{\text{rep}}(u, v) + \sum_{uv \in E} f_{\text{attr}}(u, v)$$

```
ForceDirected( $G = (V, E)$ ,  $p = (p_v)_{v \in V}$ ,  $\varepsilon > 0$ ,  $K \in \mathbb{N}$ )
 $t \leftarrow 1$ 
while  $t < K$  and  $\max_{v \in V} \|F_v(t)\| > \varepsilon$  do
  foreach  $u \in V$  do
     $F_u(t) \leftarrow \sum_{v \in V} f_{\text{rep}}(u, v) + \sum_{uv \in E} f_{\text{attr}}(u, v)$ 
  foreach  $u \in V$  do
     $p_u \leftarrow p_u + \delta(t) \cdot F_u(t)$ 
   $t \leftarrow t + 1$ 
return  $p$ 
```

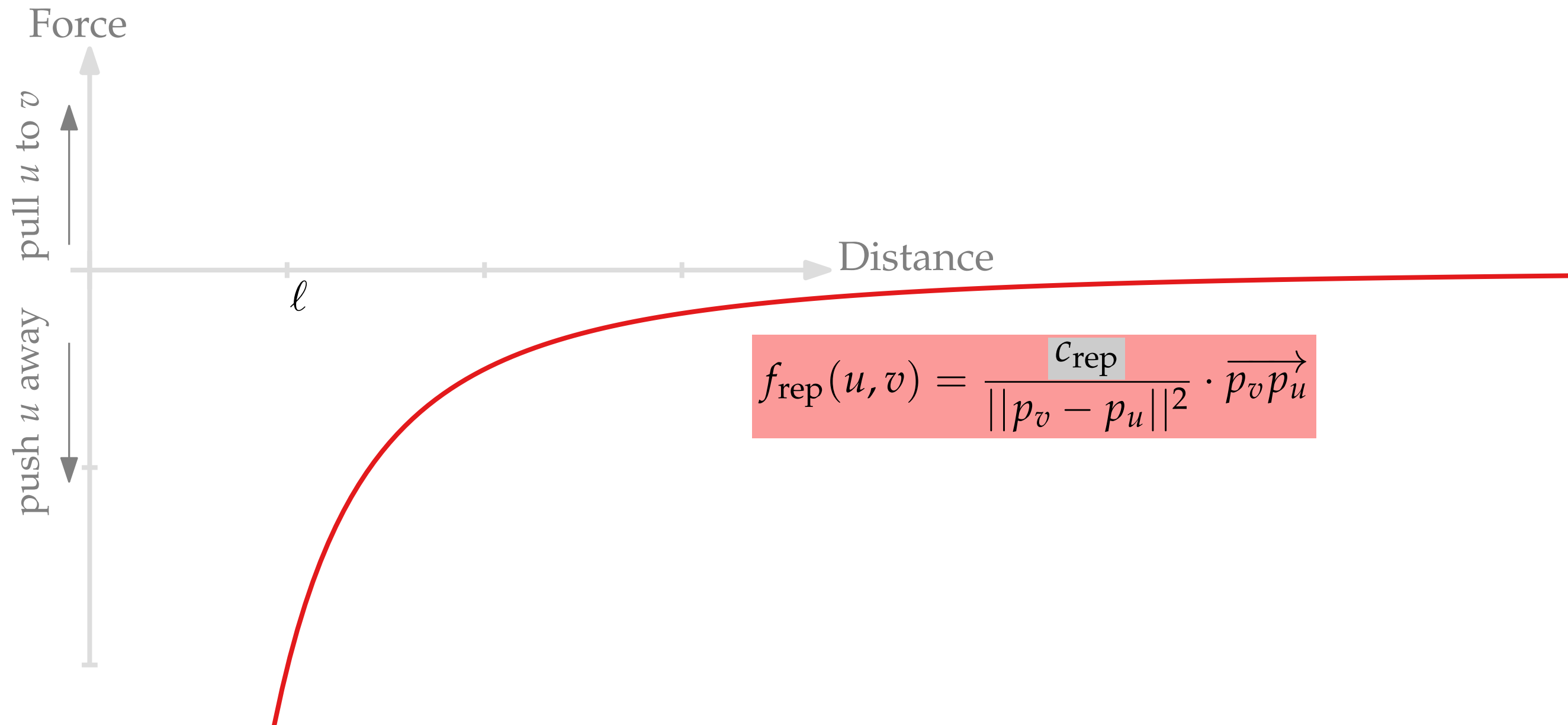
Notation.

- $\|p_u - p_v\|$ = Euclidean distance between u and v
- $\overrightarrow{p_u p_v}$ = unit vector pointing from u to v
- ℓ = ideal spring length for edges

Spring Embedder by Eades – Force Diagram



Spring Embedder by Eades – Force Diagram



$$f_{\text{rep}}(u, v) = \frac{c_{\text{rep}}}{||p_v - p_u||^2} \cdot \overrightarrow{p_v p_u}$$

Spring Embedder by Eades – Force Diagram

Force

pull u to v

push u away

ℓ

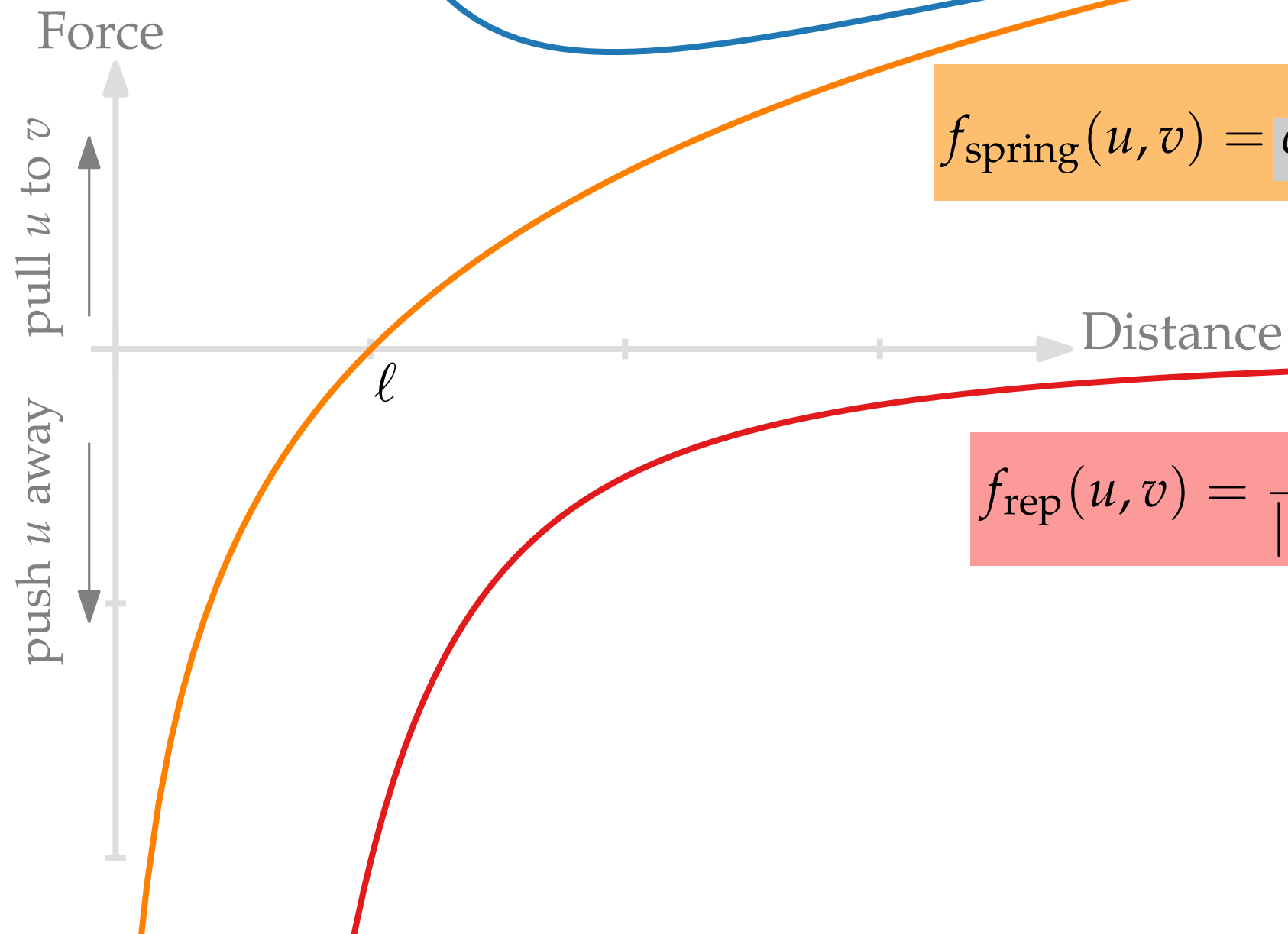
Distance

$$f_{\text{spring}}(u, v) = c_{\text{spring}} \cdot \log \frac{\|p_v - p_u\|}{\ell} \cdot \overrightarrow{p_u p_v}$$

$$f_{\text{rep}}(u, v) = \frac{c_{\text{rep}}}{\|p_v - p_u\|^2} \cdot \overrightarrow{p_v p_u}$$

Spring Embedder by Eades – Force Diagram

$$f_{\text{attr}}(u, v) = f_{\text{spring}}(u, v) - f_{\text{rep}}(u, v)$$

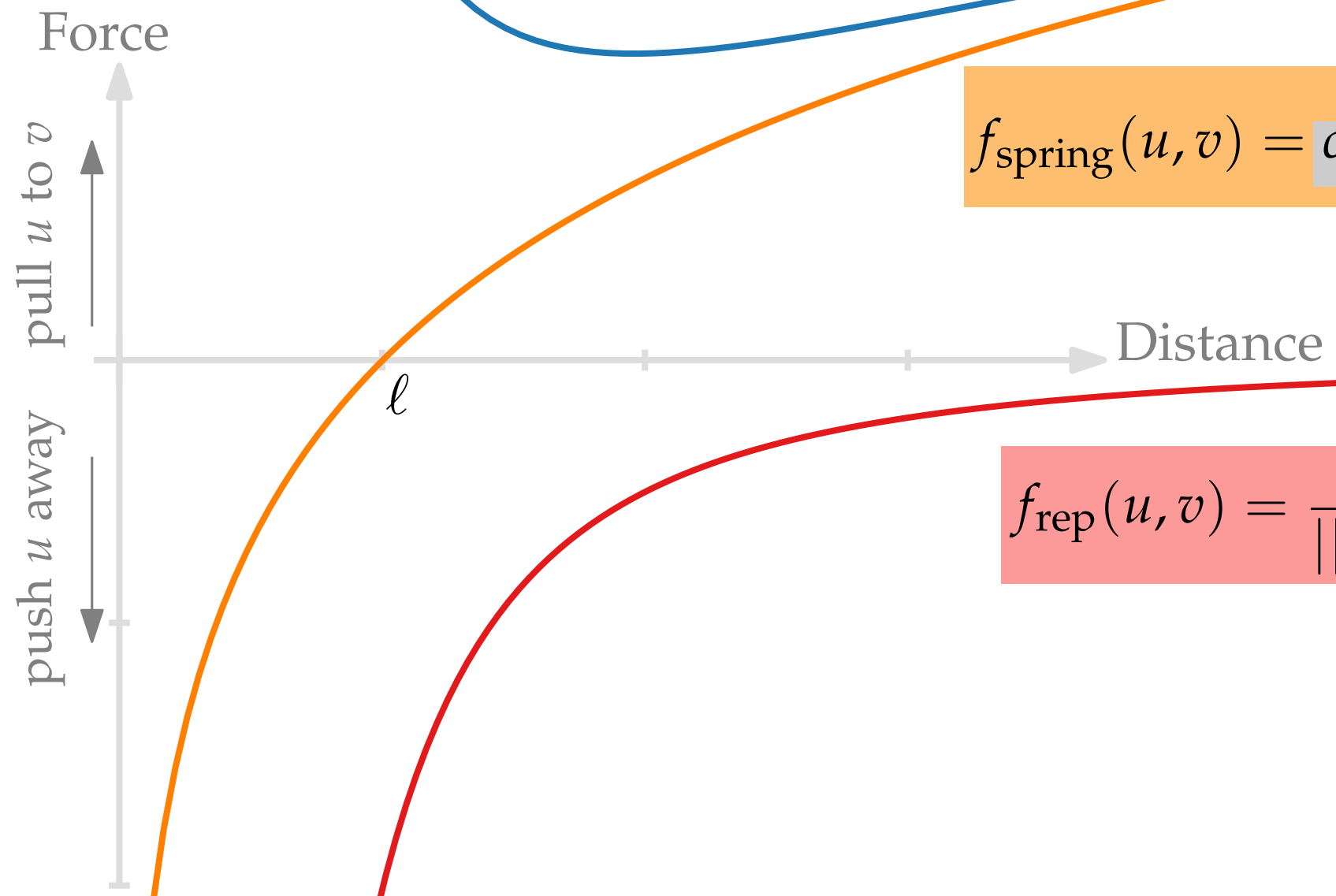


$$f_{\text{spring}}(u, v) = c_{\text{spring}} \cdot \log \frac{\|p_v - p_u\|}{\ell} \cdot \overrightarrow{p_u p_v}$$

$$f_{\text{rep}}(u, v) = \frac{c_{\text{rep}}}{\|p_v - p_u\|^2} \cdot \overrightarrow{p_v p_u}$$

Spring Embedder by Eades – Force Diagram

$$f_{\text{attr}}(u, v) = f_{\text{spring}}(u, v) - f_{\text{rep}}(u, v)$$



$$f_{\text{spring}}(u, v) = c_{\text{spring}} \cdot \log \frac{\|p_v - p_u\|}{\ell} \cdot \overrightarrow{p_u p_v}$$

$$f_{\text{rep}}(u, v) = \frac{c_{\text{rep}}}{\|p_v - p_u\|^2} \cdot \overrightarrow{p_v p_u}$$

Spring Embedder by Eades – Discussion

Advantages.

Spring Embedder by Eades – Discussion

Advantages.

- very simple algorithm

Spring Embedder by Eades – Discussion

Advantages.

- very simple algorithm
- good results for small and medium-sized graphs

Spring Embedder by Eades – Discussion

Advantages.

- very simple algorithm
- good results for small and medium-sized graphs
- empirically good representation of symmetry and structure

Spring Embedder by Eades – Discussion

Advantages.

- very simple algorithm
- good results for small and medium-sized graphs
- empirically good representation of symmetry and structure

Disadvantages.

Spring Embedder by Eades – Discussion

Advantages.

- very simple algorithm
- good results for small and medium-sized graphs
- empirically good representation of symmetry and structure

Disadvantages.

- system is not stable at the end

Spring Embedder by Eades – Discussion

Advantages.

- very simple algorithm
- good results for small and medium-sized graphs
- empirically good representation of symmetry and structure

Disadvantages.

- system is not stable at the end
- converging to local minima

Spring Embedder by Eades – Discussion

Advantages.

- very simple algorithm
- good results for small and medium-sized graphs
- empirically good representation of symmetry and structure

Disadvantages.

- system is not stable at the end
- converging to local minima
- timewise f_{spring} in $\mathcal{O}(|E|)$ and f_{rep} in $\mathcal{O}(|V|^2)$

Spring Embedder by Eades – Discussion

Advantages.

- very simple algorithm
- good results for small and medium-sized graphs
- empirically good representation of symmetry and structure

Disadvantages.

- system is not stable at the end
- converging to local minima
- timewise f_{spring} in $\mathcal{O}(|E|)$ and f_{rep} in $\mathcal{O}(|V|^2)$

Influence.

Spring Embedder by Eades – Discussion

Advantages.

- very simple algorithm
- good results for small and medium-sized graphs
- empirically good representation of symmetry and structure

Disadvantages.

- system is not stable at the end
- converging to local minima
- timewise f_{spring} in $\mathcal{O}(|E|)$ and f_{rep} in $\mathcal{O}(|V|^2)$

Influence.

- original paper by Peter Eades [Eades '84] got ~ 2000 citations

Spring Embedder by Eades – Discussion

Advantages.

- very simple algorithm
- good results for small and medium-sized graphs
- empirically good representation of symmetry and structure

Disadvantages.

- system is not stable at the end
- converging to local minima
- timewise f_{spring} in $\mathcal{O}(|E|)$ and f_{rep} in $\mathcal{O}(|V|^2)$

Influence.

- original paper by Peter Eades [Eades '84] got ~ 2000 citations
- basis for many further ideas