Algorithms for Graph Visualization

Force-Directed Algorithms

INSTITUT FÜR THEORETISCHE INFORMATIK · FAKULTÄT FÜR INFORMATIK

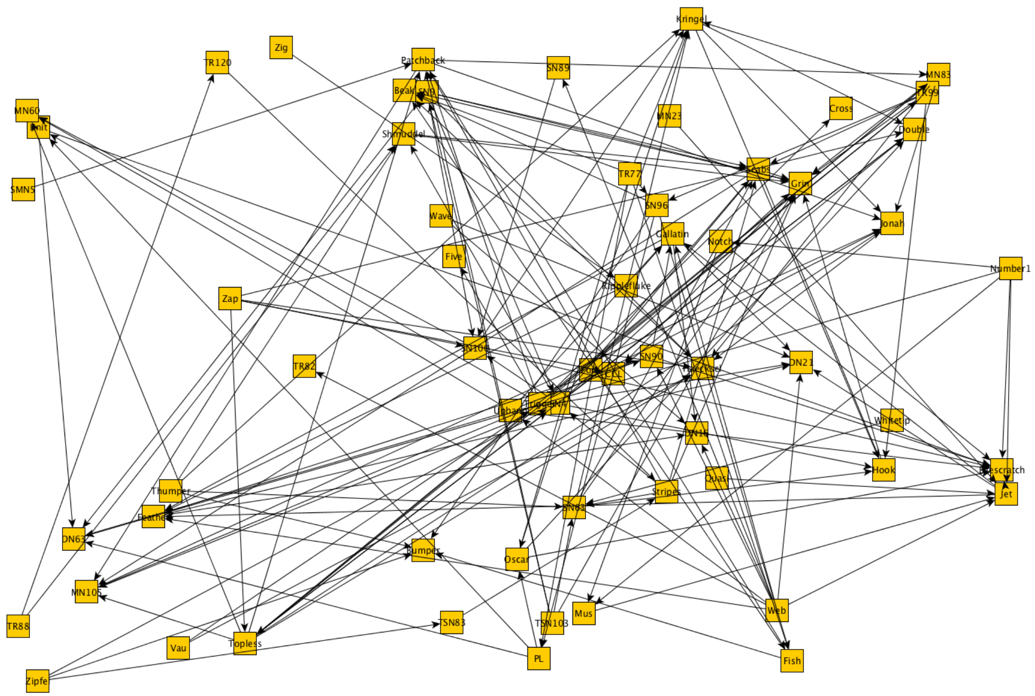
Torsten Ueckerdt

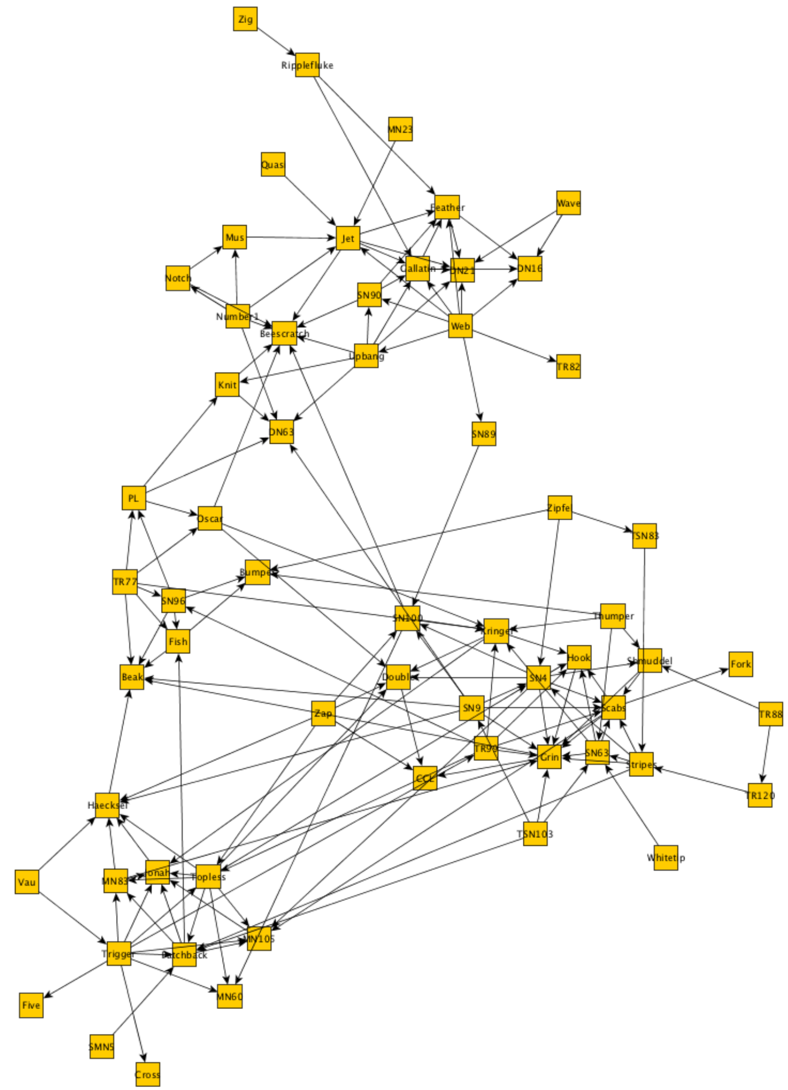
29.01.2020

General Layout Problem

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

Find: Clear and readable drawing of G





Which aesthetic criteria would you optimize?

General Layout Problem

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

Find: Clear and readable drawing of G

Criteria:

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

⚠ Optimization criteria partially contradict each other

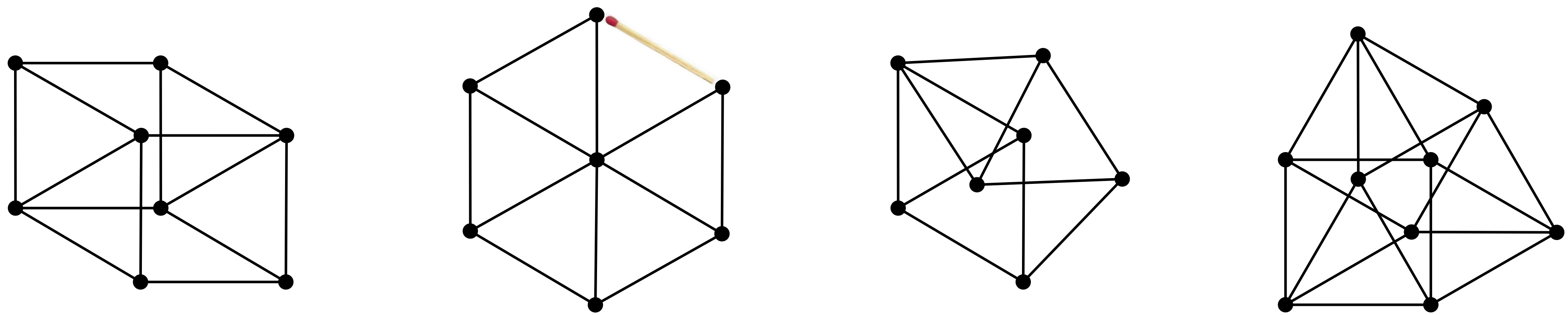
Example: Fixed edge-length

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

Find: Drawing of G which realizes all the edge lengths

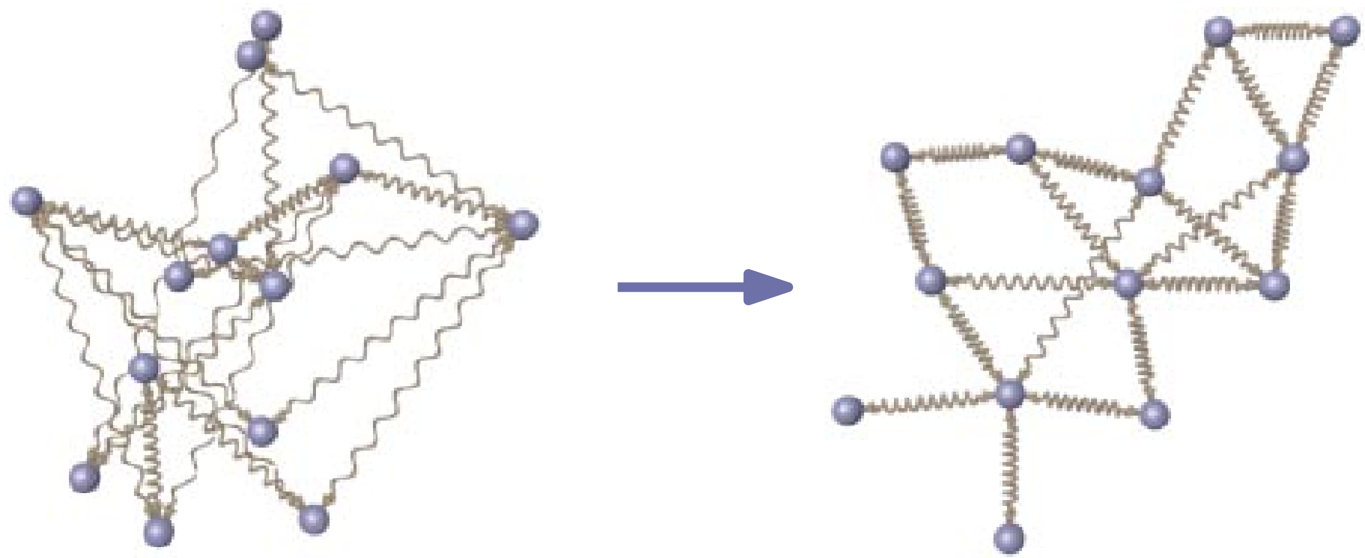
NP-hard for

- edge lengths $\{1, 2\}$ [Saxe, '80]
- planar drawing with unit edge length [Eades, Wormald, '90]



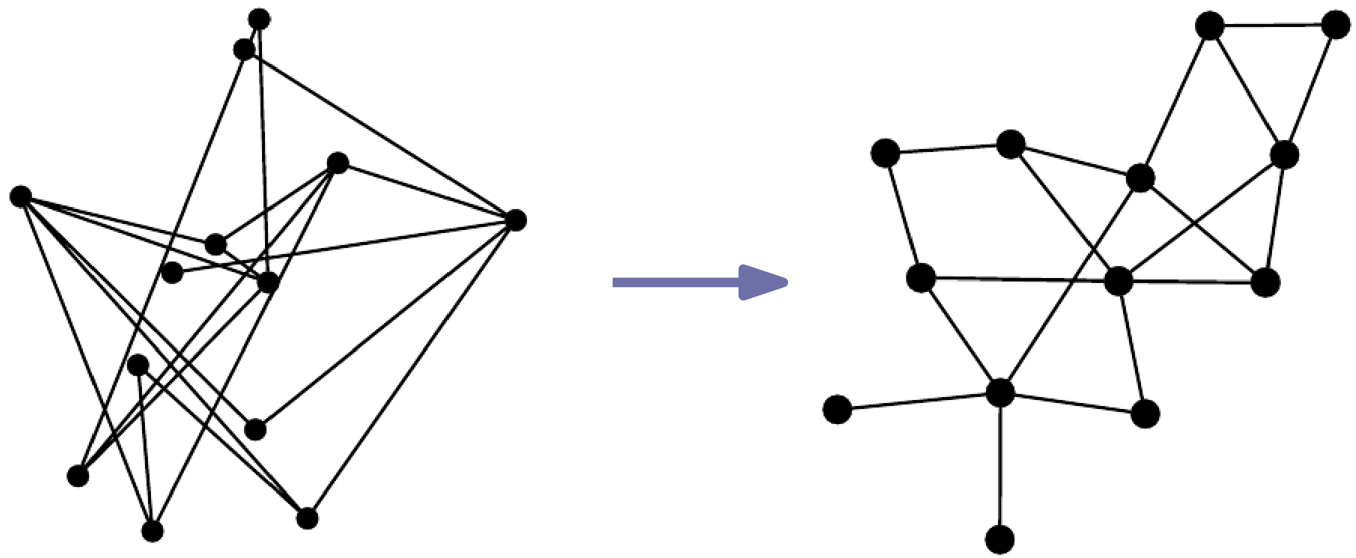
Physical Model

Karlsruhe Institute of Technology



"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." [Eades, '84]

Physical Model



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

“T… eac… are pla… e rings move the system to a minimal energy state.” [Eades, ’84]

Notation

$\ell = \ell(e)$	ideal spring length for edge e
$p_v = (x_v, y_v)$	position of node v
$\|\|p_u - p_v\|\|$	Euclidean distance between u and v
$\overrightarrow{p_u p_v}$	unit vector pointing from u to v

Spring-Embedder (Eades, 1984)

Model:

- repulsive force between two non-adjacent nodes u and v

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

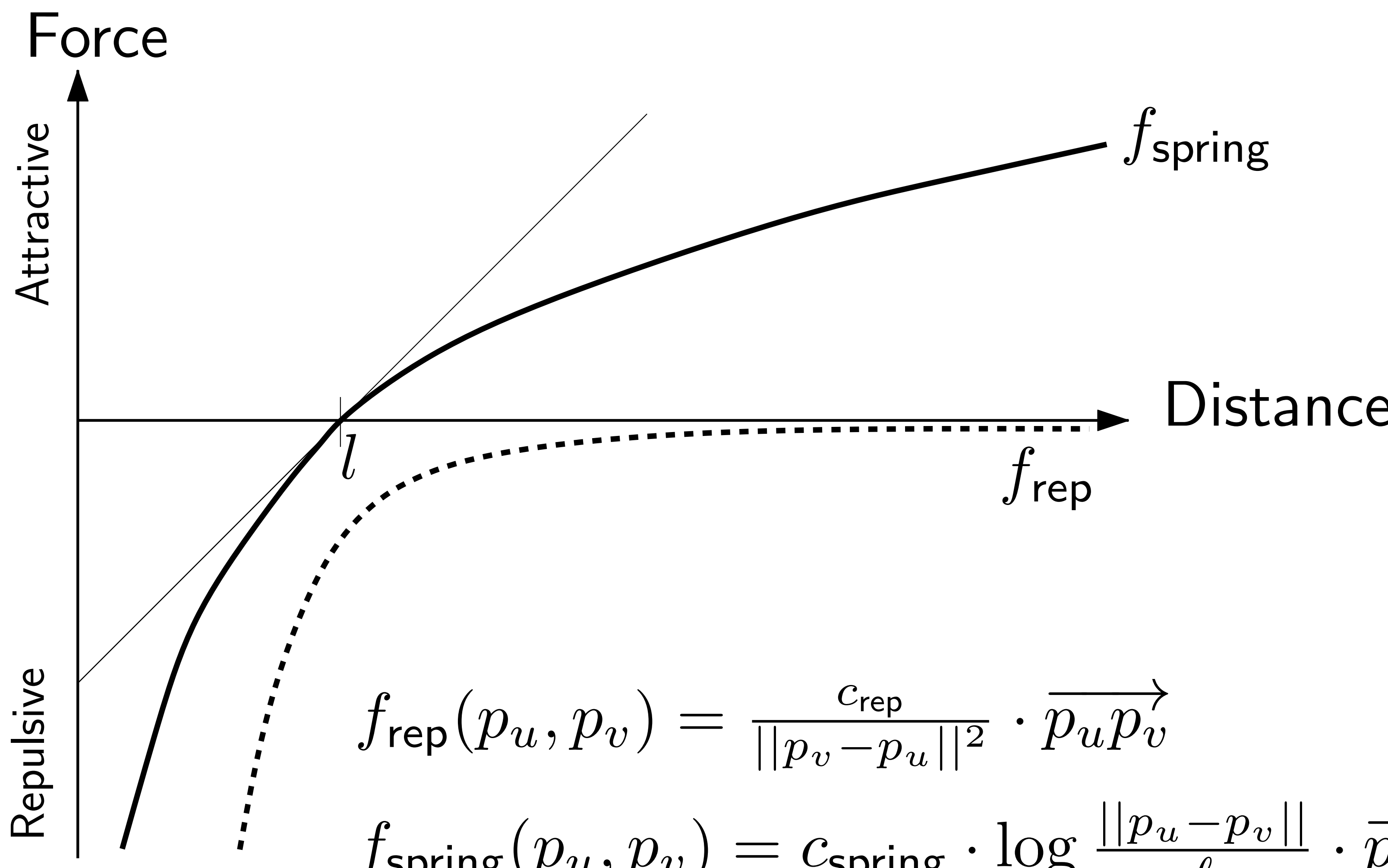
- attractive force between adjacent vertices u and v

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

- resulting displacement vector for node v

$$F_v = \sum_{u:\{u,v\} \notin E} f_{\text{rep}}(p_u, p_v) + \sum_{u:\{u,v\} \in E} f_{\text{spring}}(p_u, p_v)$$

Diagram of Spring-Embedder Forces (Eades, 1984)



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

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

Algorithm Spring-Embedder (Eades, 1984)

Input: $G = (V, E)$ connected undirected graph with initial placement $p = (p_v)_{v \in V}$, number of interations $K \in \mathbb{N}$, threshold $\varepsilon > 0$, constant $\delta > 0$

Output: Layout p with "low internal stress"

$t \leftarrow 1$

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

  foreach $v \in V$ **do**

    $F_v(t) \leftarrow \sum_{u:\{u,v\} \notin E} f_{\text{rep}}(p_u, p_v) + \sum_{u:\{u,v\} \in E} f_{\text{spring}}(p_u, p_v)$

  foreach $v \in V$ **do**

    $p_v \leftarrow p_v + \delta \cdot F_v(t)$

  $t \leftarrow t + 1$

Algorithm Spring-Embedder (Eades, 1984)

Input: $G = (V, E)$ connected undirected graph with initial placement $p = (p_v)_{v \in V}$, number of interations $K \in \mathbb{N}$, threshold $\varepsilon > 0$, constant $\delta > 0$

Output: Layout p with "low

$t \leftarrow 1$

while $t < K$ **and** $\max_{v \in V} \|F$

foreach $v \in V$ **do**

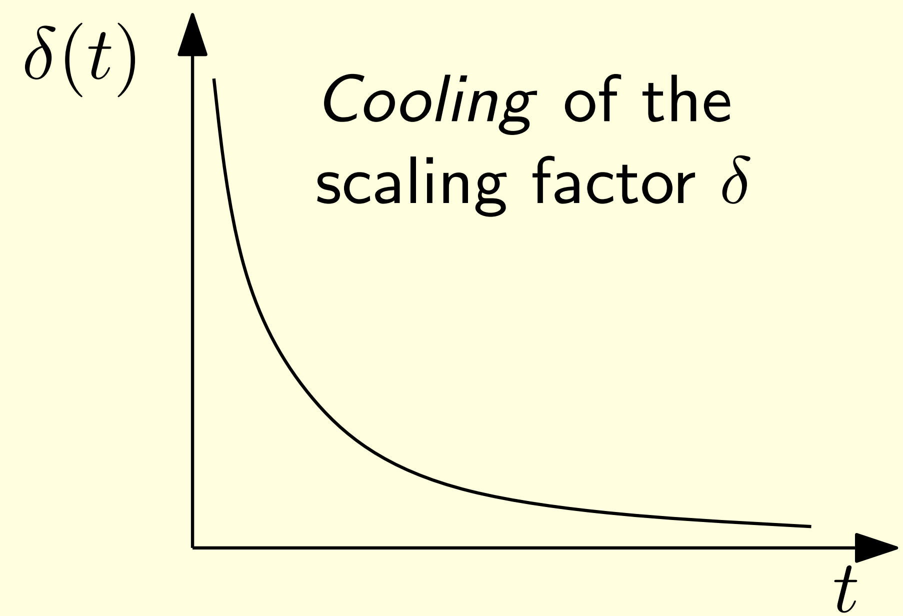
$F_v(t) \leftarrow \sum_{u:\{u,v\} \notin E} f$

$\sum_{u:\{u,v\} \in E}$

foreach $v \in V$ **do**

$p_v \leftarrow p_v + \delta(t) \cdot F_v(t)$

$t \leftarrow t + 1$



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 got 1800 citations
- basis for many further ideas

Variant: Fruchterman & Reingold (1991)

Model:

- repulsive force between **all** node pairs u and v

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

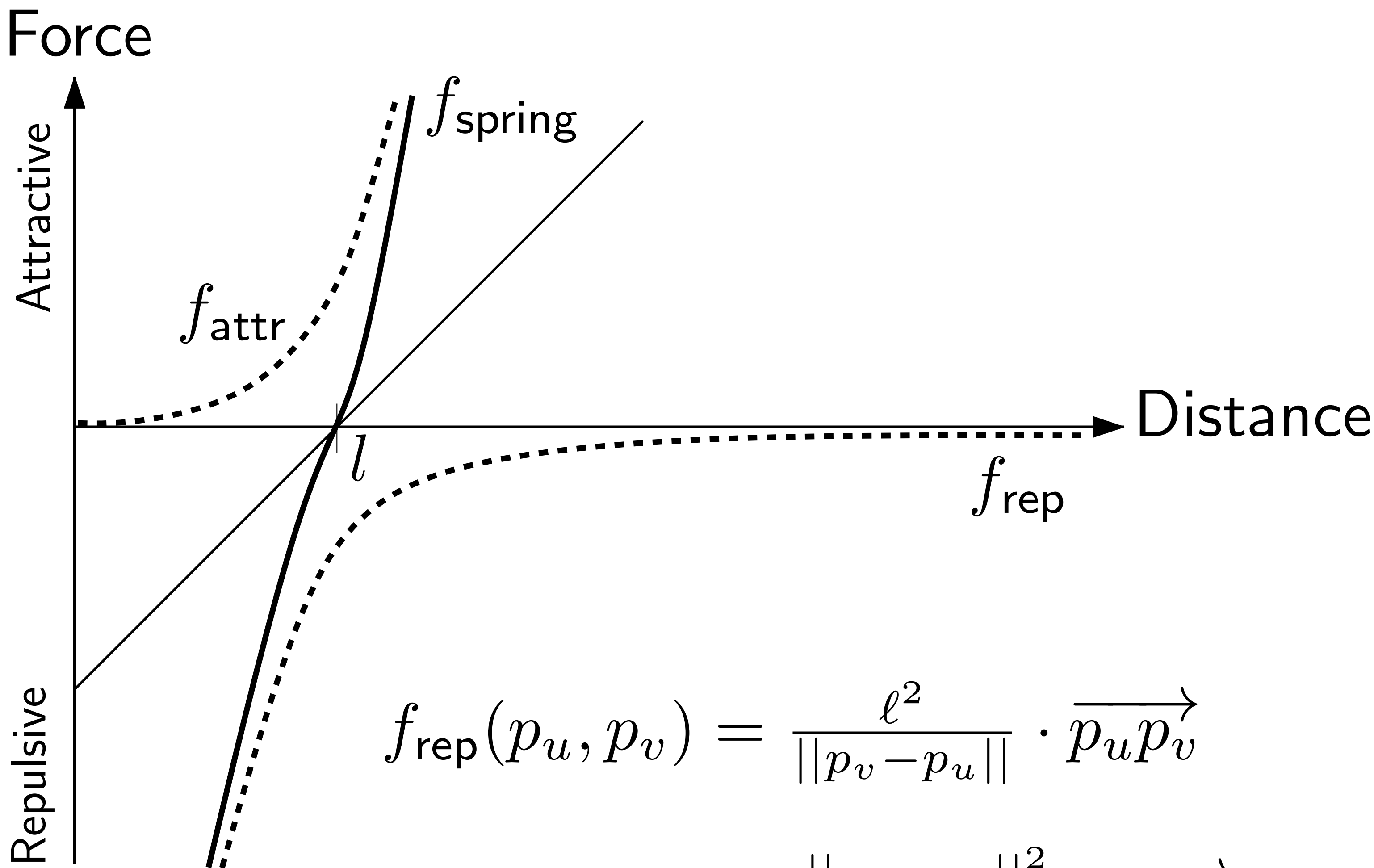
- attractive force between two adjacent nodes u and v

$$f_{\text{attr}}(p_u, p_v) = \frac{||p_u - p_v||^2}{\ell} \cdot \overrightarrow{p_v p_u}$$

- resulting force between adjacent nodes u and v

$$f_{\text{spring}}(p_u, p_v) = f_{\text{rep}}(p_u, p_v) + f_{\text{attr}}(p_u, p_v)$$

Diagramm of Fruchtermann & Reingold Forces



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

$$f_{\text{attr}}(p_u, p_v) = \frac{||p_u - p_v||^2}{\ell} \cdot \overrightarrow{p_v p_u}$$

$$f_{\text{spring}}(p_u, p_v) = f_{\text{rep}}(p_u, p_v) + f_{\text{attr}}(p_u, p_v)$$

Tutte's Barycenter Method

Karlsruhe Institute of Technology

- historically the first method (1963)
- computes crossing-free drawings for 3-connected planar graphs with convex faces
- actually a system of linear equations
- but can be considered a force-directed method

Advantages

- exact computation
- unique global minimum

Disadvantages

- poor vertex resolution
- may require exponential area