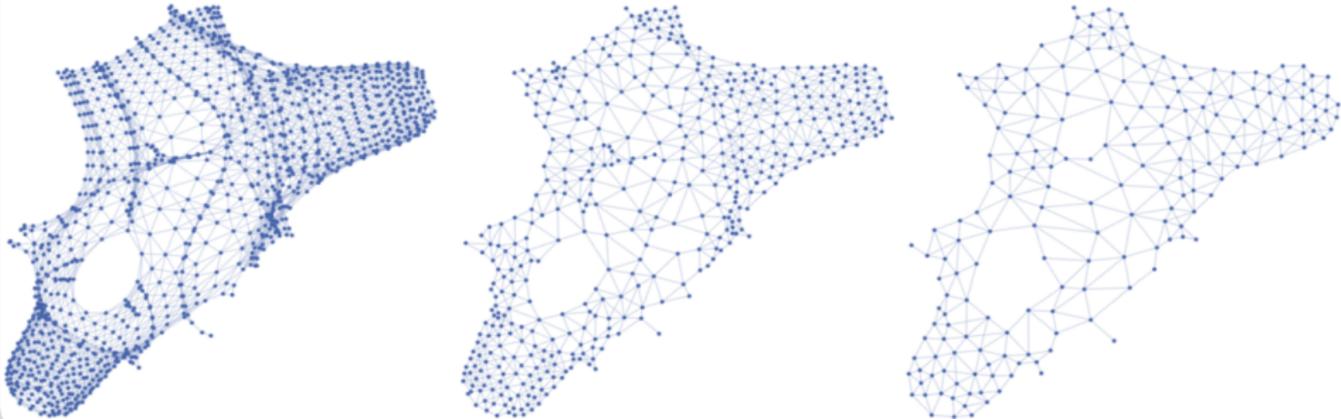


Algorithmen zur Visualisierung von Graphen

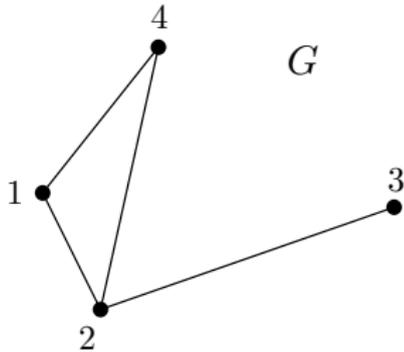
Globale und Lokale Optimierung

Marcus Krug | WS 2011/12

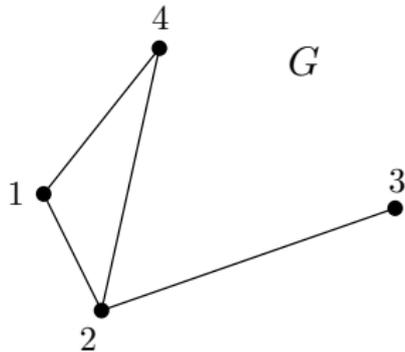
INSTITUTE OF THEORETICAL INFORMATICS
KARLSRUHE INSTITUTE OF TECHNOLOGY (KIT)



Matrizen eines Graphen

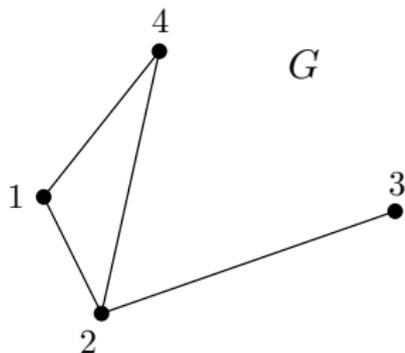


Matrizen eines Graphen



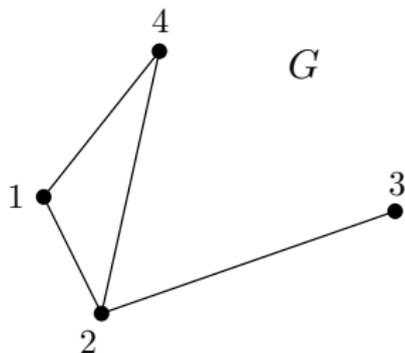
$$A(G) = \begin{matrix} & \begin{matrix} 0 & 1 & 0 & 1 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{matrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{matrix} \end{matrix}$$

Matrizen eines Graphen



$$A(G) = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$
$$D(G) = \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix}$$

Matrizen eines Graphen

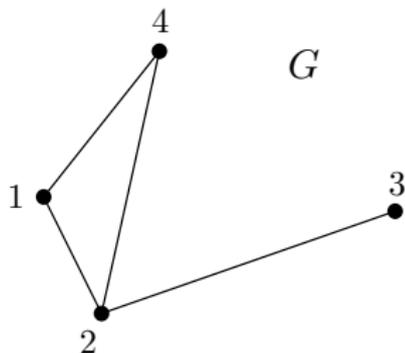


$$A(G) = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

$$D(G) = \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix}$$

$$L(G) = D(G) - A(G) = \begin{pmatrix} 2 & -1 & 0 & -1 \\ -1 & 3 & -1 & -1 \\ 0 & -1 & 1 & 0 \\ -1 & -1 & 0 & 2 \end{pmatrix}$$

Matrizen eines Graphen



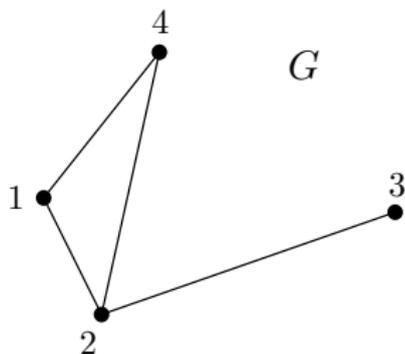
$$A(G) = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

$$D(G) = \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix}$$

$$L(G) = D(G) - A(G) = \begin{pmatrix} 2 & -1 & 0 & -1 \\ -1 & 3 & -1 & -1 \\ 0 & -1 & 1 & 0 \\ -1 & -1 & 0 & 2 \end{pmatrix}$$

The matrix $L(G)$ is shown with red annotations: a vertical line labeled v through the second column, a horizontal line labeled u through the third row, and their intersection at the element 3 in the second row, third column.

Matrizen eines Graphen



$$A(G) = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

$$D(G) = \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix}$$

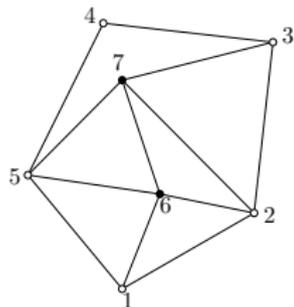
$$L(G) \overset{V_0}{=} D(G) - A(G) = \begin{pmatrix} 2 & -1 & 0 & -1 \\ -1 & 3 & -1 & -1 \\ 0 & -1 & 1 & 0 \\ -1 & -1 & 0 & 2 \end{pmatrix}$$

Laplace-Matrix und Spann­b­ume

Theorem (Matrix-Ger­ust-Satz; Kirchhoff, 1847)

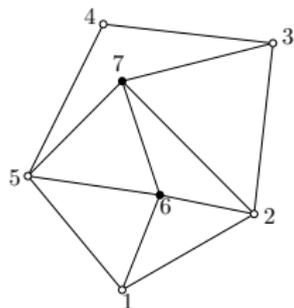
F­ur jeden Multigraphen G und einen beliebigen Knoten v in G ist $|L(G)^{vv}|$ gleich der Anzahl $t(G)$ der aufspannenden B­ume von G .

Kontraktion

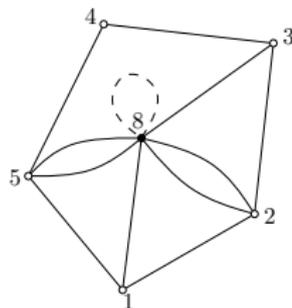


$$L(G) = \begin{pmatrix} 3 & -1 & 0 & 0 & -1 & 0 & -1 \\ -1 & 4 & -1 & 0 & 0 & -1 & -1 \\ 0 & -1 & 3 & -1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ -1 & 0 & 0 & -1 & 4 & -1 & -1 \\ 0 & -1 & -1 & 0 & -1 & 4 & -1 \\ -1 & -1 & 0 & 0 & -1 & -1 & 4 \end{pmatrix}$$

Kontraktion



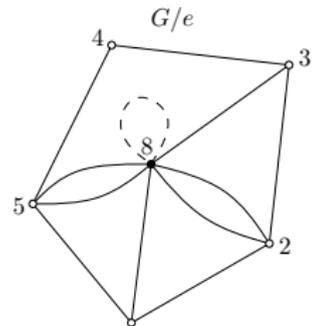
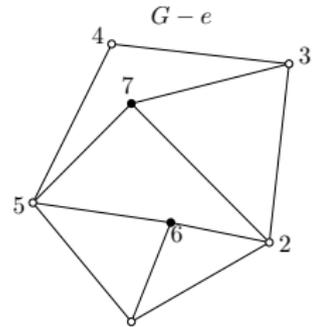
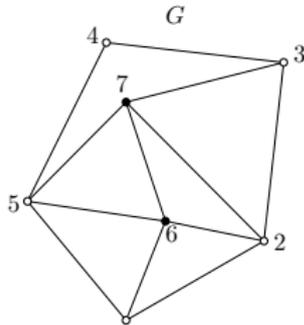
Kontraktion →



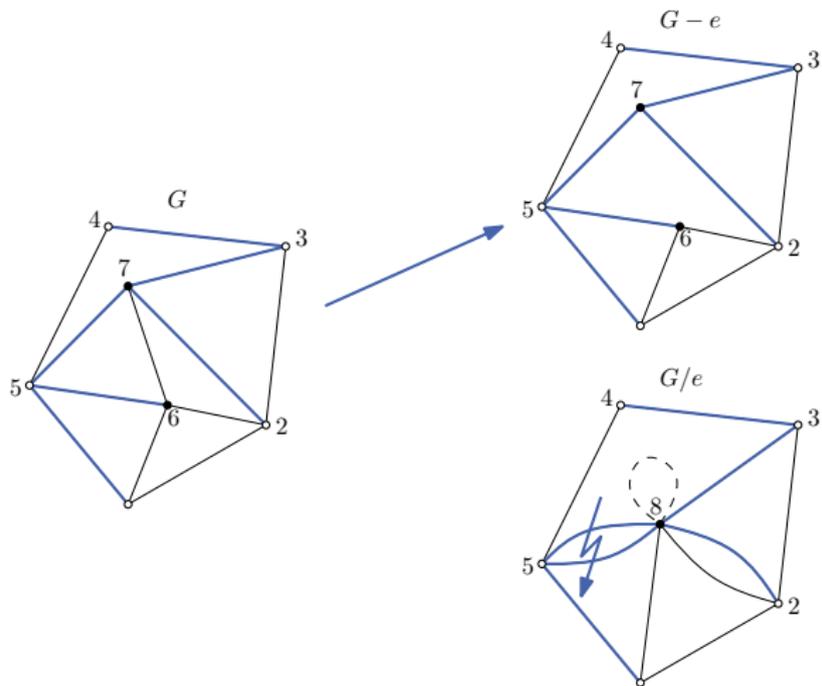
$$L(G) = \begin{pmatrix} 3 & -1 & 0 & 0 & -1 & 0 & -1 \\ -1 & 4 & -1 & 0 & 0 & -1 & -1 \\ 0 & -1 & 3 & -1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ -1 & 0 & 0 & -1 & 4 & -1 & -1 \\ 0 & -1 & -1 & 0 & -1 & 4 & -1 \\ -1 & -1 & 0 & 0 & -1 & -1 & 4 \end{pmatrix}$$

$$L(G/\{6,7\}) = \begin{pmatrix} 3 & -1 & 0 & 0 & -1 & -1 \\ -1 & 4 & -1 & 0 & 0 & -2 \\ 0 & -1 & 3 & -1 & 0 & -1 \\ 0 & 0 & -1 & 2 & -1 & 0 \\ -1 & 0 & 0 & -1 & 4 & -2 \\ -1 & -2 & -1 & 0 & -2 & 6 \end{pmatrix}$$

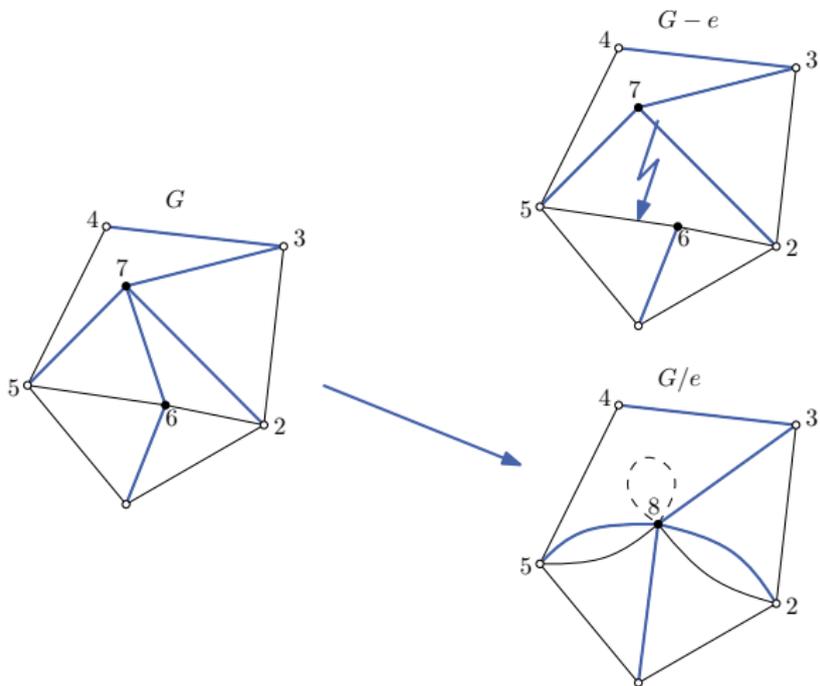
Spannbaum-Kontraktion



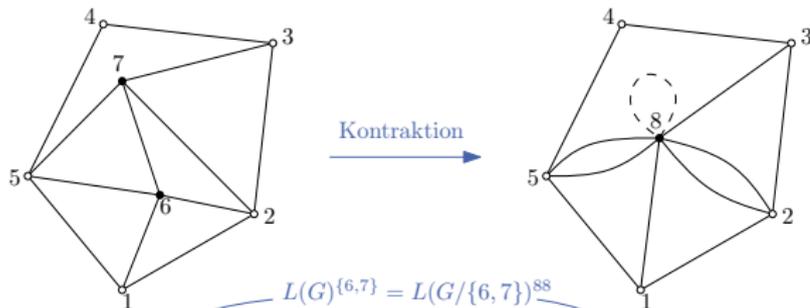
Spannbaum-Kontraktion



Spannbaum-Kontraktion



Kontraktion



$$L(G) = \begin{bmatrix} 3 & -1 & 0 & 0 & -1 & 0 & -1 \\ -1 & 4 & -1 & 0 & 0 & -1 & -1 \\ 0 & -1 & 3 & -1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ -1 & 0 & 0 & -1 & 4 & -1 & -1 \\ 0 & -1 & -1 & 0 & -1 & 4 & -1 \\ -1 & -1 & 0 & 0 & -1 & -1 & 4 \end{bmatrix}$$

$$L(G)^{\{6,7\}} = L(G/\{6,7\})^{88}$$

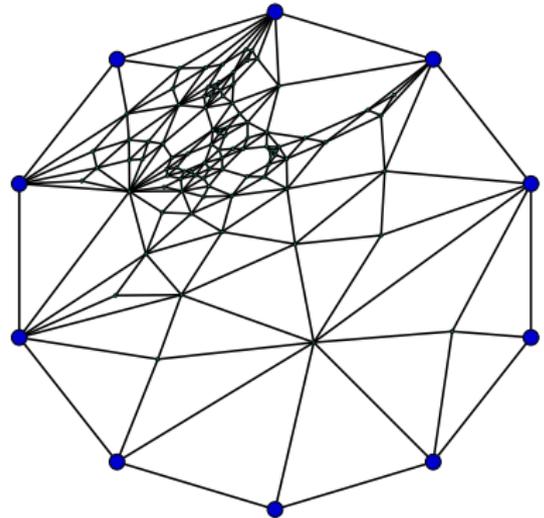
$$L(G/\{6,7\}) = \begin{bmatrix} 3 & -1 & 0 & 0 & -1 & -1 \\ -1 & 4 & -1 & 0 & 0 & -2 \\ 0 & -1 & 3 & -1 & 0 & -1 \\ 0 & 0 & -1 & 2 & -1 & 0 \\ -1 & 0 & 0 & -1 & 4 & -2 \\ -1 & -2 & -1 & 0 & -2 & 6 \end{bmatrix}$$

Eigenschaften von Schwerpunkt-Layouts

Theorem (Tutte, 1963)

Sei G ein planarer Graph

- 3-fach zusammenhängend
 - Knoten der äußeren Facette in konvexer Lage
- ⇒ Schwerpunktlayout kreuzungsfrei und innere Facetten konvex.



Eigenschaften von Schwerpunkt-Layouts

Theorem

Die *worst-case* Auflösung in Schwerpunktlayouts ist $\mathcal{O}(1/\lambda^n)$ für ein $\lambda > 1$.

Berechnung mit Gauß-Seidel Iteration

Algorithmus 1: Schwerpunktlayout nach Gauß-Seidel

Input: $G = (V, E)$, \hat{p}_v , $v \in V_0 \subset V$

Output: p_v , $v \in V$ ($p_v = \hat{p}_v$ für alle $v \in V_0$)

- 1 **foreach** $v \in V_0$ **do** $p_v \leftarrow \hat{p}_v$
- 2 **while** p ändert sich noch nennenswert **do**
- 3 **foreach** $v \in V \setminus V_0$ **do**
- 4
$$p_v \leftarrow \frac{1}{d_G(v)} \cdot \sum_{u \in V: \{u,v\} \in E} p_u$$

Spektralmethoden

Definition (Spektrallayout)

Sei $G = (V, E)$ ein zusammenhängender Graph und $L(G)$ seine Laplace-Matrix mit zugehörigen Eigenpaaren $(\lambda_1, v_1), \dots, (\lambda_n, v_n)$, und

$$\lambda_i \leq \lambda_{i+1} \text{ und } v_i \perp v_{i+1} \quad (1 \leq i < n) .$$

Dann ist das (Laplace'sche) Spektrallayout von G in 2D definiert als:

$$x := v_2 \quad \text{und} \quad y := v_3 .$$

Andere Eigenvektoren auch verwendbar (siehe Demo).

Berechnung von Eigenvektoren

Idee

- starte mit Zufallsvektor x_0
- berechne

$$x_{i+1} = \frac{Ax_i}{\|Ax_i\|}$$

- konvergiert gegen dominanten Eigenvektor v_n
- verwende $A' := 2 \max_{w \in V} \deg(w) I_n - L(G)$
- zusätzliche Orthogonalisierung zur Berechnung von v_i

$$x'_{i+1} = x_{i+1} - \sum_{k=1}^{i-1} \frac{\langle x_{i+1}, v_k \rangle}{\langle v_k, v_k \rangle} x_k$$

Berechnung von Eigenvektoren

Algorithmus 2: Poweriteration (für Laplace'sche Matrizen)

Eingabe : $G = (V, E)$, Eigenvektoren $\{v_1, \dots, v_i\}$

Ausgabe : Eigenvektor v_{i+1}

- 1 $x \leftarrow \text{random}(\text{span}(v_1, \dots, v_i)^\perp)$
- 2 **while** x keine gute Näherung an einen Eigenvektor **ist do**
- 3 $x' \leftarrow \mathbf{0}$
- 4 **foreach** $v \in V$ **do**
- 5 $x'_v \leftarrow (2 \cdot \max_{w \in V} \deg(w) - \deg(v)) \cdot x_v + \sum_{\{u,v\} \in E} x_u$
- 6 $x \leftarrow x' - \sum_{k=1}^i \frac{\langle x', v_k \rangle}{\langle v_k, v_k \rangle} v_k$
- 7 $x \leftarrow x / \|x\|$

Multidimensional Scaling

Idee

- lege Ideal-Distanz zwischen allen Knotenpaaren fest
- z.B. Distanz im Graphen $d_G(u, v)$
- finde Einbettung in \mathbb{R}^2 , die Distanzen möglichst gut approximiert
- minimiere

$$U(p) = \sum_{u, v \in V} \frac{(\|p_u - p_v\| - d_G(u, v))^2}{d_G(u, v)^2}$$

Multidimensional Scaling

Iterative Berechnung mit Gradientenverfahren

$$\frac{\partial U(p)}{\partial x_v} = \sum_{u \neq v} \frac{2}{d_G(u, v)^2} \left((x_u - x_v) - \frac{d_G(u, v) \cdot (x_u - x_v)}{\|p_u - p_v\|} \right)$$

$$\frac{\partial U(p)}{\partial y_v} = \sum_{u \neq v} \frac{2}{d_G(u, v)^2} \left((y_u - y_v) - \frac{d_G(u, v) \cdot (y_u - y_v)}{\|p_u - p_v\|} \right)$$

- starte mit zufälligem Vektor p_0
- setze

$$p_{i+1} = p_i + \alpha \cdot \nabla U(p_i)$$

Simulated Annealing

Layouteigenschaften als globale Zielfunktion

- gleichmäßige Verteilung der Knoten

$$\sum_{u,v \in V} \lambda_1 \cdot \frac{1}{\|p_u - p_v\|^2}$$

Simulated Annealing

Layouteigenschaften als globale Zielfunktion

- gleichmäßige Verteilung der Knoten
- Kantenlängen ähnlich zu einer vorgegebenen Länge

$$\sum_{\{u,v\} \in E} \lambda_3 \cdot \|p_u - p_v\|^2$$

Simulated Annealing

Layouteigenschaften als globale Zielfunktion

- gleichmäßige Verteilung der Knoten
- Kantenlängen ähnlich zu einer vorgegebenen Länge
- Minimierung von Kreuzungen

$$\sum_{\{u,v\},\{x,y\} \in E} \lambda_4 \cdot \chi((p_u, p_v), (p_x, p_y))$$

Simulated Annealing

Layouteigenschaften als globale Zielfunktion

- gleichmäßige Verteilung der Knoten
- Kantenlängen ähnlich zu einer vorgegebenen Länge
- Minimierung von Kreuzungen
- angemessener Abstand zwischen Knoten und Kanten

$$\sum_{\substack{u \in V, \{v, w\} \in E \\ u \notin \{v, w\}}} \begin{cases} \lambda_5 & , \text{ falls } \text{dist}(p_u, \overline{p_x p_y}) < \text{dist}_{\min} \\ \frac{\lambda_6}{\text{dist}(p_u, \overline{p_x p_y})^2} & , \text{ sonst} \end{cases}$$

Simulated Annealing

Layouteigenschaften als globale Zielfunktion

- gleichmäßige Verteilung der Knoten
- Kantenlängen ähnlich zu einer vorgegebenen Länge
- Minimierung von Kreuzungen
- angemessener Abstand zwischen Knoten und Kanten
- gute Winkelauflösung

$$\sum_{\substack{\{u,v\}, \{v,w\} \in E, \\ u \neq w}} \lambda_7 \cdot \frac{1}{\angle(p_u p_v p_w)}$$

Simulated Annealing

Layouteigenschaften als globale Zielfunktion

- gleichmäßige Verteilung der Knoten
- Kantenlängen ähnlich zu einer vorgegebenen Länge
- Minimierung von Kreuzungen
- angemessener Abstand zwischen Knoten und Kanten
- gute Winkelauflösung
- globale Zielfunktion ist gewichtete Summe

Ausblick

weitere globale Methoden

- Lineare/Quadratische Programmierung
- Constraint Programming