Crossing Angle Maximaztion

- Maximizing the smallest angle between any pair of crossing edges
- Straight-line embedding
- Input graphs are arbitrary undirected graphs

General idea

- Calculate force for each node
- Try to bring the graph into a minimum energy state
- Assume time discrete

$\Longrightarrow$ Each time step until convergence add force vectors to node positions

Forces

$G = (V, E)$

- Nodes are charged particles

$$\implies f_{el}(n_1, n_2) = \frac{\varepsilon_{el}}{\mathrm{d}\,(n_1, n_2)} \;\; \forall n_1, n_2 \in V$$

- Edges are springs

$$\implies f_{sp}(n_1, n_2) = \varepsilon_{sp} \cdot (\mathrm{d}\,(n_1, n_2) - l_{sp}) \;\; \forall e = (n_1, n_2) \in E$$

- Increase crossing angles

$$\implies f_{cr}(\vec{e}_1, \vec{e}_2, \theta) = (\hat{e_1}, \hat{e_2}) \cdot \varepsilon_{cr} \cos(\theta) \;\; \forall e_1, e_2 \in \text{Crossings} \subseteq E \times E$$

- Evenly distribute edges around nodes

$$\implies f_{n_1}(n_2, n_3, \phi) = \varepsilon_{nb} \cdot \sin\left(\frac{\phi - \theta}{2}\right) \;\; \forall n_1 \in V, n_2, n_3 \in \text{Neighbours}(n_1)$$

Problems and solutions

- For large graphs this produces small drawings
- Springs are too inflexible

$$l \sim \log(|N|), \ \ \varepsilon'_{sp} = \frac{\mathrm{d}\,(n_1, n_2) - l}{l}$$

$$\implies f'_{sp}(n_1, n_2) = \varepsilon'_{sp} \cdot \tan^{-1}\left({\varepsilon'_{sp}}^{4}\right) \ \ \forall e = (n_1, n_2) \in E$$

- Spring embedder converges to local minimum

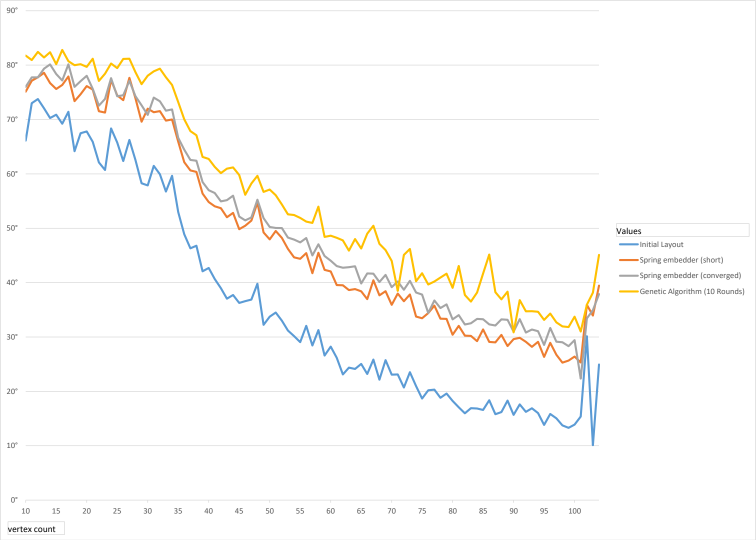
$\implies$ Change embedding locally, keep good changes

$\implies$ Genetic algorithm

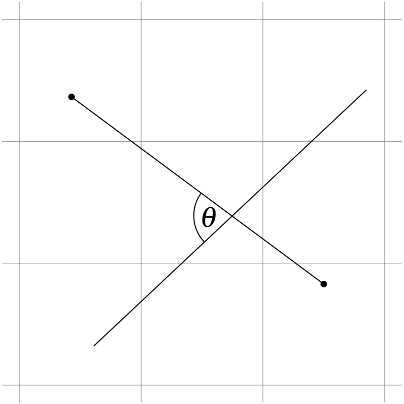
Mutation strategies

- Change parameters of forces
- Exchange force variants
- Remove crossing with worst crossing angle by swapping node positions

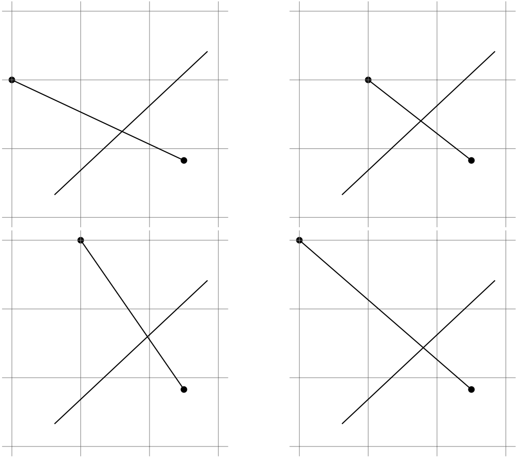
Some data


90°
80°
70°
60°
50°
40°
30°
20°
10°
0°
10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100
vertex count
Values
Initial Layout
Spring embedder (short)
Spring embedder (converged)
Genetic Algorithm (10 Rounds)

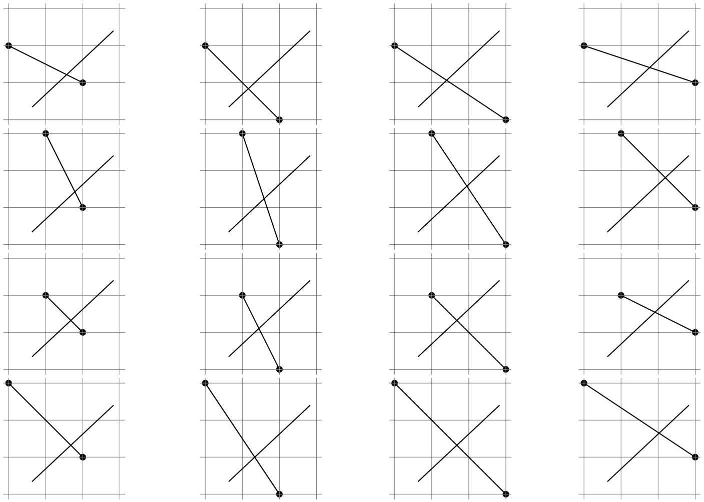
Gridding the graph



Four possibilities

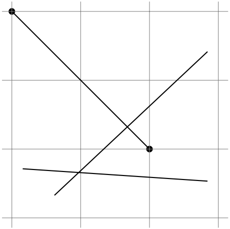


Sixteen possibilities...

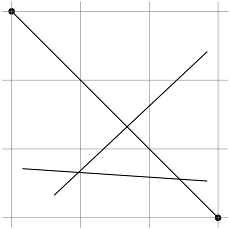


Implementation of two algorithms

- Simple Gridding: Check for each vertex v if creates new crossing with grid coordinates?

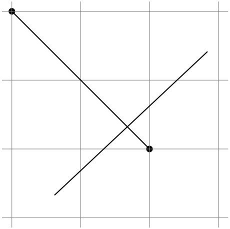


$\surd$

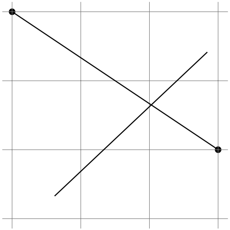


$\lightning$

- Non-Simple Gridding: Check for each vertex v that minimum angle not compromised!



$\surd$



$\lightning$

Trade-Off

Simple Gridding:

- Fast
- Creates still some new crossings (sometime inevitable) $\Rightarrow$ Worse angles

Non-Simple Gridding:

- Slow
- Creates no new crossings