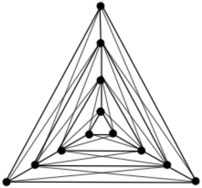
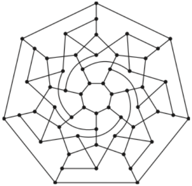
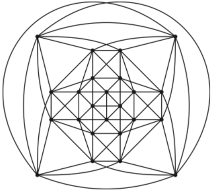
Force-Directed Layout of Node-Link Diagrams

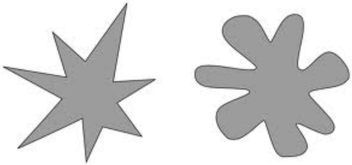
Stephen Kobourov
University of Arizona

University of Saskatchewan
June 17-19, 2019

Motivation: Visualization

- Pictures are powerful

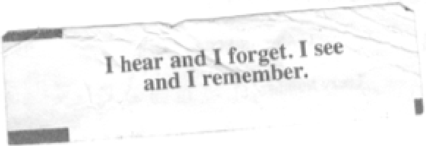


Motivation: Visualization

- Pictures are powerful
- People are good at pictures

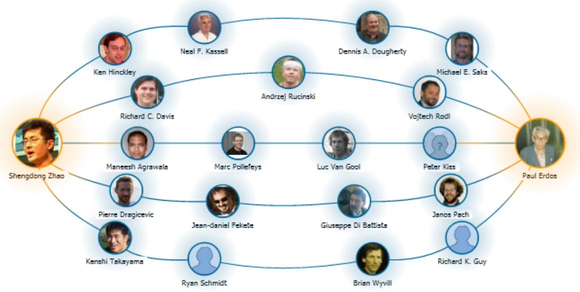
Motivation: Visualization

- Pictures are powerful
- People are good at pictures
- A picture is worth a 1000 words



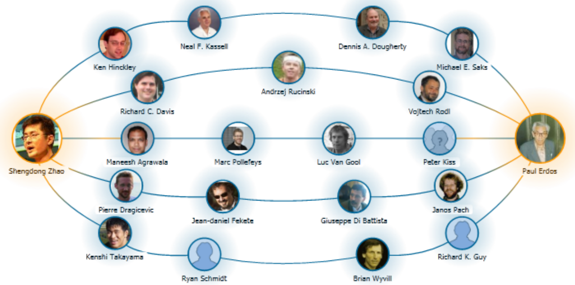
Graphs

- *Graphs* model relational data
  - objects → nodes
  - relationships → links



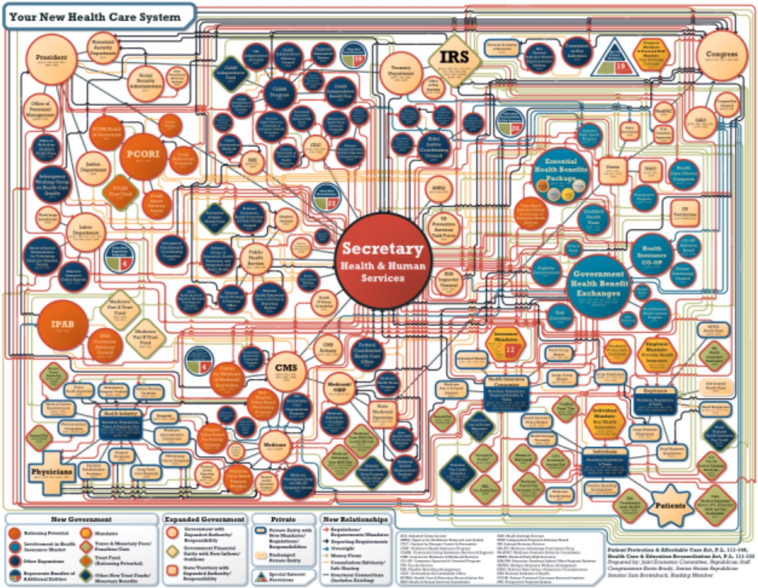
Graphs

- *Graphs* model relational data
  - objects → nodes
  - relationships → links

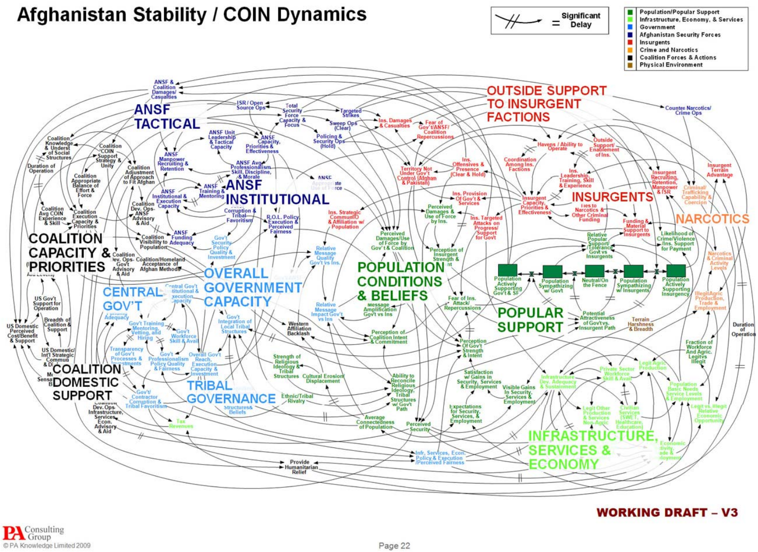


- *Graphs* model metric data
  - points → nodes
  - distances → weighted links

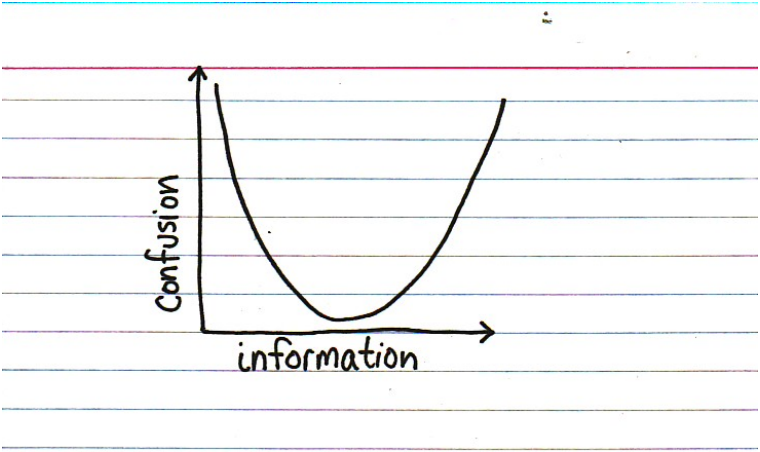
Confusing Graphs



Confusing Graphs

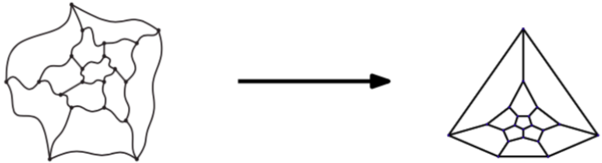


Less is More



What is Graph Drawing (ELI5)?

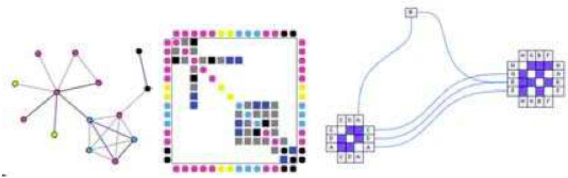
Given an abstract graph $G = (V, E)$ with a set of vertices V and edges E produce a visualization.



Goal: the visualization (drawing) should provide some guarantees, e.g., if the graph is planar, the drawing will be crossings-free.

What Graphs are We Drawing?

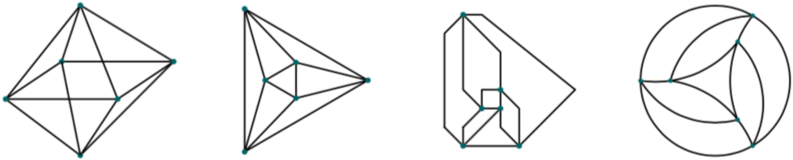
- Graphs ("plain vanilla")
  - vertices (nodes)
  - edges (links)
  - vertices are unweighted and unlabeled
  - edges are unweighted, unlabeled and undirected
- Graph Representation
  - node link diagram
  - adjacency matrix
  - mixed representations
  - other



(a) Node-link diagram (b) Adjacency matrix diagram (c) Combination

Node Link Diagrams

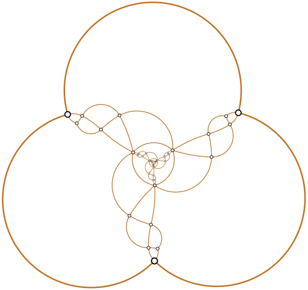
- Node Link Diagram Variants
  - straight-line, or not (curved, polyline, ...)
  - vertices on integer grid, or not
  - 2D Euclidean plane, or not (3D, sphere, hyperbolic space,...)
  - single static layout, or not (multi-level, dynamic, ...)

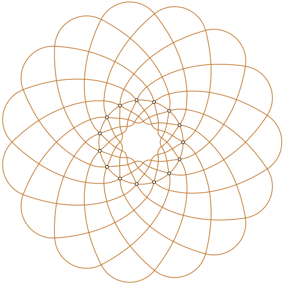


Looking at Graphs

The Pursuit of Beauty

PLANAR & POLY-ARC





LOMBARDI DRAWINGS

Tutte's Algorithm (1960)

Given graph $G = (V, E)$ compute coordinates for each vertex:

$$p(u) = \frac{1}{deg(u)} \sum_{v \in N(u)} p(v)$$

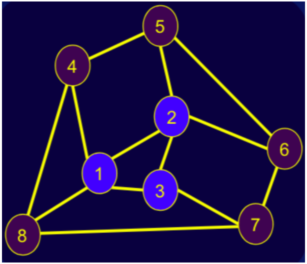
- $p(u)$ is the point in the 2D plane for vertex u
- $deg(u)$ is the degree of vertex u
- $N(u)$ is the set of neighbors of u in G

Of course, we have to solve $2n$ equations:

$$x(u) = \frac{1}{deg(u)} \sum_{v \in N(u)} x(v) \qquad y(u) = \frac{1}{deg(u)} \sum_{v \in N(u)} y(v)$$

Tutte's Algorithm (1960), cont.

- Place 3 or more vertices in general positions (e.g., 4,5,6,7,8)
- Solve the system of equations for the other vertices (e.g., 1,2,3)



$$x(u) = \frac{1}{deg(u)} \sum_{v \in N(u)} x(v) \qquad y(u) = \frac{1}{deg(u)} \sum_{v \in N(u)} y(v)$$

Tutte's Algorithm (1960), cont.

$$x_1 = \frac{1}{4}(x_2 + x_3 + x'_4 + x'_8)$$

$$x_2 = \frac{1}{4}(x_1 + x_3 + x'_5 + x'_6)$$

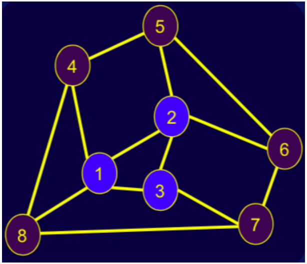
$$x_2 = \frac{1}{3}(x_1 + x_2 + x'_7)$$

and

$$y_1 = \frac{1}{4}(y_2 + y_3 + y'_4 + y'_8)$$

$$y_2 = \frac{1}{4}(y_1 + y_3 + y'_5 + y'_6)$$

$$y_2 = \frac{1}{3}(y_1 + y_2 + y'_7)$$

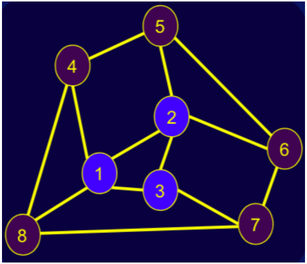


Tutte's Algorithm (1960), cont.

$$4x_1 - x_2 - x_3 = x'_4 + x'_8 = c_1$$
$$-x_1 + 4x_2 - x_3 = x'_5 + x'_6 = c_2$$
$$-x_1 - x_2 + 3x_3 = x'_5 = c_3$$

and

$$4y_1 - y_2 - y_3 = y'_4 + y'_8 = d_1$$
$$-y_1 + 4y_2 - y_3 = y'_5 + y'_6 = d_2$$
$$-y_1 - y_2 + 3y_3 = y'_5 = d_3$$



c_1, c_2, c_3 and d_1, d_2, d_3 are constants from the placed vertices

Tutte's Algorithm (1960), cont.

Step 3. Find vectors $x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$ and $y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$

such that

$$Mx = c$$

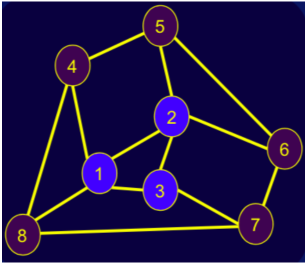
and

$$My = d$$

where

$$M = \begin{bmatrix} 4 & -1 & -1 \\ -1 & 4 & -1 \\ -1 & -1 & 3 \end{bmatrix}$$

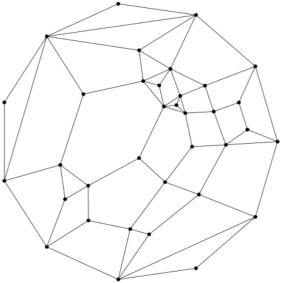
$$M_{u,v} = \begin{cases} deg(u), & if \;\; u = v \\ -1, & if (u, v) \in E \\ 0, & otherwise \end{cases}$$



Tutte's Algorithm (1960), cont.

$$M_{u,v} = \begin{cases} deg(u), \ \ if \ \ u = v \\ -1, \ \ if(u, v) \in E \\ 0, \ \ otherwise \end{cases}$$

- Note that this matrix is the Laplacian of the graph
- Tutte's barycentric algorithm boils down to inverting 2 matrices
- This is similar to the more recently famous PageRank algorithm



Tutte's Algorithm (1960), cont.

Theorem

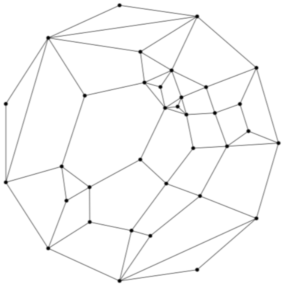
If G is 3-connected and planar then Tutte's algorithm produces a straight-line, crossings-free drawing with convex faces.

Proof.

Pick a face f of G.

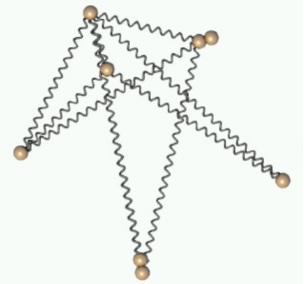
Place the vertices of f on the corners of a polygon of size $|f|$.

Apply Tutte's barycentric algorithm. $\square$



Energy View of Tutte's Algorithm

- The Euclidean distance between vertices u and v in a drawing D is $d(u,v) = \sqrt{(x_u - x_v)^2 + (y_u - y_v)^2}$
- The force on edge (u,v) is $d(u,v)^2 = (x_u - x_v)^2 + (y_u - y_v)^2$
- The energy of the entire drawing D is the sum of the energy of the edges:



$$E(D) = \sum_{(u,v)\in E} d(u,v)^2 = \sum_{(u,v)\in E} (x_u - x_v)^2 + (y_u - y_v)^2$$

- Minimize the energy

Energy View of Tutte's Algorithm (cont.)

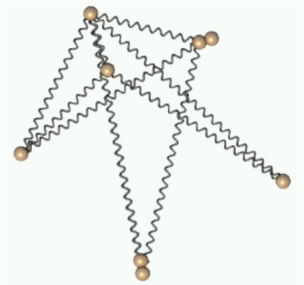
- Compute vertex positions to minimize:

$$E(D) = \sum_{(u,v)\in E} (x_u - x_v)^2 + (y_u - y_v)^2$$

- Minimum is unique and occurs at:

$$\frac{\partial E(D)}{\partial x_u} = 0 \qquad \frac{\partial E(D)}{\partial y_u} = 0$$

- This yields the barycentric equations



Energy View of Tutte's Algorithm (cont.)

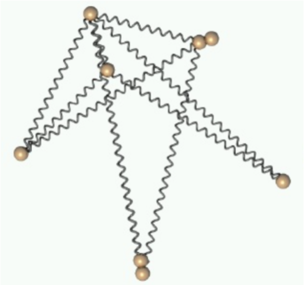
Solving the equations for x and y:

$$\frac{\partial E(D)}{\partial x_u} = 0$$

$$\frac{\partial}{\partial x_u}\left(\sum_{(u,v)\in E}(x_u - x_v)^2 + (y_u - y_v)^2\right) = 0$$

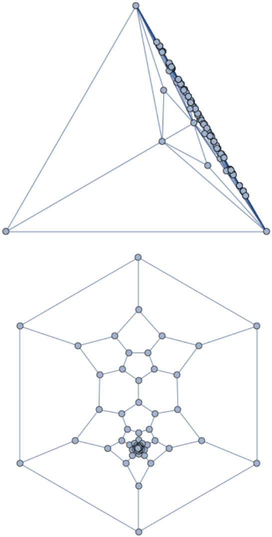
$$\sum_{v\in V,(u,v)\in E} 2(x_u - x_v) = 0$$

$$x_u = \frac{1}{deg(u)} \sum_{v\in V,(u,v)\in E} x_v$$

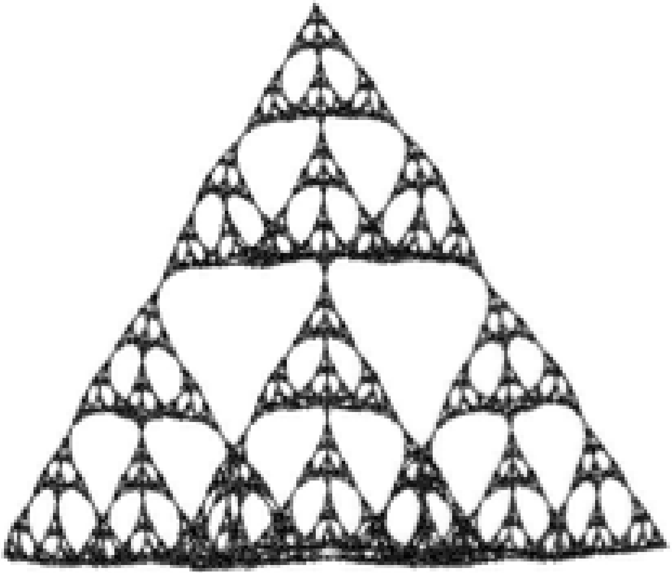


Are We Done Now?

- \+ Beautiful algorithm
- \+ Simple and elegant
- \+ Reasonably efficient
- \+ Strong planar guarantees
- – Poor vertex resolution
- – Requires exponential drawing area



Force-Directed Methods



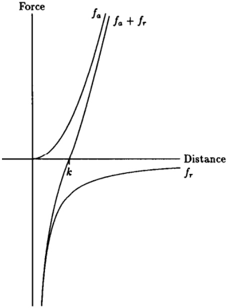
Fruchterman-Reingold:

$$F(v) = F_r(v) + F_a(v)$$

$$F_r(v) = \sum_{\forall u \in V} \frac{-\kappa^2}{\|pos[u] - pos[v]\|}(pos[u] - pos[v])$$

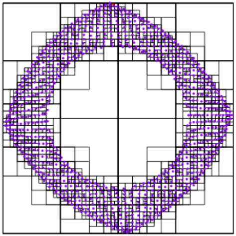
$$F_a(v) = \sum_{u \in Adj(v)} \frac{\|pos[u] - pos[v]\|^2}{\kappa}(pos[u] - pos[v])$$

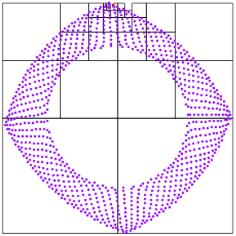
- $\kappa = c\sqrt{Area/|V|}$, ideal distance between vertices
- Start from random positions and iteratively improve



Faster Force Calculations

- The $O(n^2)$ repulsive calculation can be approximated
- Barnes-Hut n-body simulation can be done in $O(n \log n)$ time
- Quigley and Eadges (FADE) 2000
- Hachul and Jünger (FM3) 2004
- Hu (sfdp) 2008

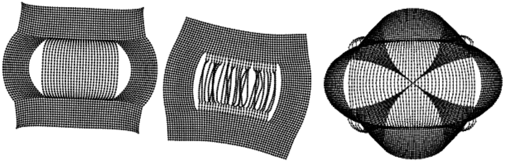




Multi-Level Methods

First consider an abstraction, disregarding some of the graph's fine details. This abstraction is then drawn, yielding a "rough" layout in which only the general structure is revealed. Then the details are added and the layout is corrected.

- Hadany and Harel 2000: edge contractions
- Harel and Koren 2000: k-centers
- Walshaw 2000: maximal independent set
- Gajer et al. 2000 (GRIP): vertex filtrations, high dimensional embedding

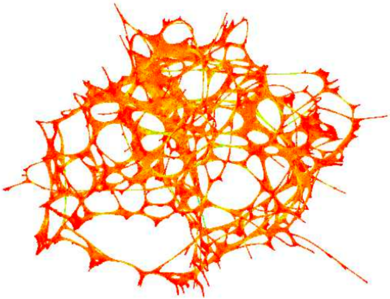


(a) (b) (c)

Fig. 9. (a,b) 55x55 (3025-vertex) sparse grid: (a) beautification considered larger than normal neighborhoods, of radius 35 (b) beautification considered usual neighborhoods of radius 7; (c) 80x80 (6400-vertex) sparse grid with each two opposite corners connected

AT&T: GraphViz and sfdp

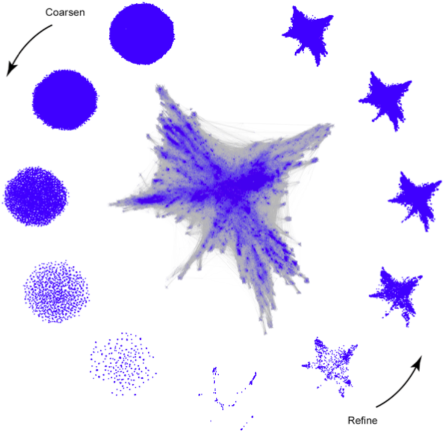
- Multi-scale: edge contractions or maximal independent set
- Barnes-Hut n-body simulation with quad-trees



boneS10. $|V| = 914898, |E| = 27276762$

Sandia: VxOrd and OpenOrd

- scalable version of Fruchterman-Reingold with simulated annealing
- recently multi-level and parallelized



MDS for Graph Drawing

- Kamada-Kawai use multi-dimensional scaling (MDS)
- Idea: match Euclidean distances to graph distances

$$F(v) = \sum_{u \in V} \left(\frac{\|pos[u] - pos[v]\|^2}{(\kappa \times dist_G(u,v))^2} - 1 \right) (pos[u] - pos[v])$$

- $\kappa = \sqrt{A_{frame}/|E|}$, ideal edge length

Multi-Dimensional Scaling (MDS)

- Morse code confusion matrix: (probability send i, receive j)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	1	2	3	4	5	6	7	8	9	0
A	92	04	06	13	03	14	10	13	46	05	22	03	25	34	06	06	09	35	23	06	37	13	17	12	07	03	02	07	05	05	08	06	05	06	02	03
B	05	84	37	31	05	28	17	21	05	19	34	40	06	10	12	22	25	16	18	02	18	34	08	84	30	42	12	17	14	40	32	74	43	17	04	04
C	04	38	87	17	04	29	13	07	11	19	24	35	14	03	09	51	34	24	14	06	06	11	14	32	82	38	13	15	31	14	10	30	28	24	18	12
D	08	62	17	88	07	23	40	36	09	13	81	56	08	07	09	27	09	45	29	06	17	20	27	40	15	33	03	09	06	11	09	19	08	10	05	06
E	06	13	14	06	97	02	04	04	17	01	05	06	04	04	05	01	05	10	07	67	03	03	02	05	06	05	04	03	05	03	05	02	04	02	03	03
F	04	51	33	19	02	90	10	29	05	33	16	50	07	06	10	42	12	35	14	02	21	27	25	19	27	13	08	16	47	25	26	24	21	05	05	05
G	09	18	27	38	01	14	90	06	05	22	33	16	14	13	82	52	23	21	05	03	15	14	32	21	23	39	15	14	05	10	04	10	17	23	20	11
H	03	45	23	25	09	32	08	87	10	10	09	29	05	08	08	14	08	17	37	04	36	59	09	33	14	11	03	09	15	43	70	35	17	04	03	03
I	64	07	07	13	10	08	06	12	93	03	05	16	13	30	07	03	05	19	35	16	10	05	08	02	05	07	02	05	08	09	06	08	05	02	04	05
J	07	09	38	09	02	24	18	05	04	85	22	31	08	03	21	63	47	11	02	07	09	09	09	22	32	28	67	66	33	15	07	11	28	29	26	23
K	05	24	38	73	01	17	25	11	05	27	91	33	10	12	31	14	31	22	02	02	23	17	33	63	16	18	05	09	17	08	08	18	14	13	05	06
L	02	69	43	45	10	24	12	26	09	30	27	86	06	02	09	37	36	28	12	05	16	19	20	31	25	59	12	13	17	15	26	29	36	16	07	03
M	24	12	05	14	07	17	29	08	08	11	23	08	96	62	11	19	15	20	07	09	13	04	21	09	18	08	05	07	06	06	05	07	11	07	10	04
N	31	04	13	30	08	12	10	16	13	03	16	08	59	93	05	09	05	28	12	10	16	04	12	04	06	11	05	02	03	04	04	06	02	02	10	02
O	07	07	20	06	05	09	76	07	02	39	26	10	04	08	86	37	35	10	03	04	11	14	25	35	27	27	19	17	07	07	06	18	14	11	20	12
P	05	22	33	12	05	36	22	12	03	78	14	46	05	06	21	83	43	23	09	04	12	19	19	19	41	30	34	44	24	11	15	17	24	23	25	13
Q	08	20	38	11	04	15	10	05	02	27	23	26	07	06	22	51	91	11	02	03	06	14	12	37	50	63	34	32	17	12	09	27	40	58	37	24
R	13	14	16	23	05	34	26	15	07	12	21	37	14	12	12	29	08	87	16	02	23	23	62	14	12	13	07	10	13	04	07	12	07	09	01	02
S	17	24	05	30	11	26	05	59	16	03	13	10	05	17	06	06	03	18	96	09	56	24	12	10	06	07	08	02	02	15	28	09	05	05	05	02
T	13	10	01	05	46	03	06	06	14	06	14	07	06	05	06	11	04	04	07	96	08	05	04	02	02	06	05	05	03	03	03	08	07	06	14	06
U	14	29	12	32	04	32	11	34	21	07	44	32	11	13	06	20	12	40	51	06	93	57	34	17	09	11	06	06	16	34	10	09	09	07	04	03
V	05	17	24	16	09	29	06	39	05	11	26	43	04	01	09	17	10	17	11	06	32	92	17	57	35	10	10	14	28	79	44	36	25	10	01	05
W	09	21	30	22	09	36	25	15	04	25	29	18	15	06	26	20	25	61	12	04	19	20	86	22	25	22	10	22	19	16	05	09	11	06	03	07
X	07	64	45	19	03	28	11	06	01	35	50	42	10	08	24	32	61	10	12	03	12	17	21	91	48	26	12	20	24	27	16	57	29	16	17	06
Y	09	23	62	15	04	26	22	09	01	30	12	14	05	06	14	30	52	05	07	04	06	13	21	44	86	23	26	44	40	15	11	26	22	33	23	16
Z	03	46	45	18	02	22	17	10	07	23	21	51	11	02	15	59	72	14	04	03	09	11	12	36	42	87	16	21	27	09	10	25	66	47	15	15
1	02	05	10	03	03	05	13	04	02	29	05	14	09	07	14	30	28	09	04	02	03	12	14	17	19	22	84	63	13	08	10	08	19	32	57	55
2	07	14	22	05	04	20	13	03	25	26	09	14	02	03	17	37	28	06	05	03	06	10	11	17	30	13	62	89	54	20	05	14	20	21	16	11
3	03	08	21	05	04	32	06	12	02	23	06	13	05	02	05	37	19	09	07	06	04	16	06	22	25	12	18	64	86	31	23	41	16	17	08	10
4	06	19	19	12	06	25	14	16	07	21	13	19	03	03	02	17	29	11	09	03	17	55	08	37	24	03	05	26	44	89	42	44	32	10	03	03
5	08	45	15	14	02	45	04	67	07	14	04	41	02	00	04	13	07	09	27	02	14	45	07	45	10	10	14	10	30	69	90	42	24	10	06	05
6	07	80	30	17	04	23	04	14	02	11	11	27	06	02	07	16	30	11	14	03	12	30	09	58	38	39	15	14	26	24	17	86	69	14	05	14
7	06	33	22	14	05	25	06	04	06	24	13	32	07	06	07	36	39	12	06	02	03	13	09	30	30	50	22	29	18	15	12	61	85	70	20	13
8	03	23	40	06	03	15	15	06	02	33	10	14	03	06	14	12	45	02	06	04	06	07	05	24	35	50	42	29	16	16	09	30	60	89	61	26
9	03	14	23	03	01	06	14	05	02	30	06	07	16	11	10	31	32	05	06	07	06	03	08	11	21	24	57	39	09	12	04	11	42	56	91	78
0	09	03	11	02	05	07	14	04	05	30	08	03	02	03	25	21	29	02	03	04	05	03	02	12	15	20	50	26	09	11	05	22	17	52	81	94

Multi-Dimensional Scaling (MDS)

- Morse code confusion matrix: (probability send i, receive j)
- Visualize it "nicely": see patterns and trends

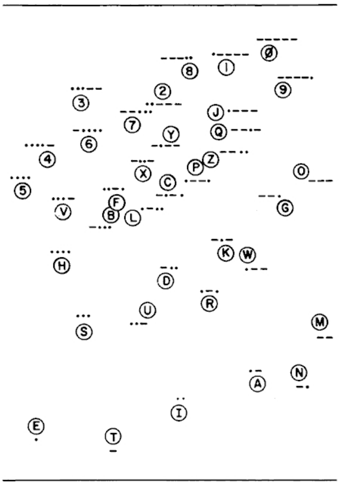
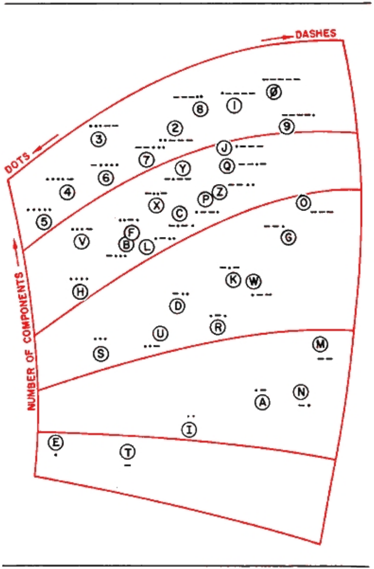


Figure 3A: Configuration Resulting from Morse Code Similarities

Multi-Dimensional Scaling (MDS)

- Morse code confusion matrix: (probability send i, receive j)
- Visualize it “nicely”: see patterns and trends
- Why is this useful? tricky TLAs: BPK, FZW



Stress Model

The stress model assumes that there are springs connecting all pairs of vertices of the graph, with the ideal spring length equal to the predefined edge length. The energy of this spring system is

$$E = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{1}{2} k_{i,j} (|p_i - p_j| - l_{i,j})^2,$$

- Kamada-Kawai is an example
- neato in GraphViz
- optimize in force-directed fashion
- optimize using stress minimization/majorization

MDS Strain Model

Classical Multidimensional Scaling (MDS) tries to fit the inner product of positions, instead of the distance between points.

- LandmarkMDS places subset of nodes by classical MDS; other node positions based on distances to already placed nodes
- PivotMDS uses Singular Value Decomposition
- neato in GraphViz

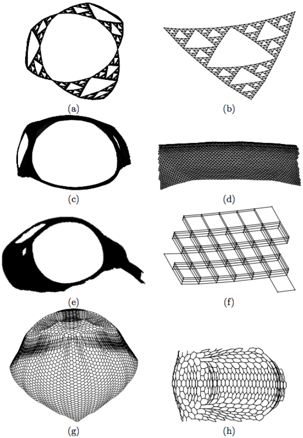
(a) Pivot MDS (b) sparse stress minim. (c) original

Fig. 5. Drawings for a large graph representing the street network in Germany (4 044 153 nodes, 9 564 235 edges, diameter 1 059)

High Dimensional Embedding

The high-dimensional embedding (HDE) algorithm computes vertex positions in k-dimensional space and projects them to a suitably chosen 2D or 3D space.

- k-centers are chosen as in LandmarkMDS
- the graph distances from each vertex to the k-centers form a k-dimensional coordinate system
- find projection to 2D space that minimizes correlations
- very fast $O(n)$ time



Comparative Analysis: Brandenburg et al. 1996

- Algorithms: KK, FR, DH, GEM, TU
- Measures: running time, ratio of longest to shortest edge, number of crossings
- Dataset: 59 graphs from the papers describing the algorithms + K_4 to K_{24}
- Graph sizes: under 100 vertices
- Conclusions: similar results, no clear winner

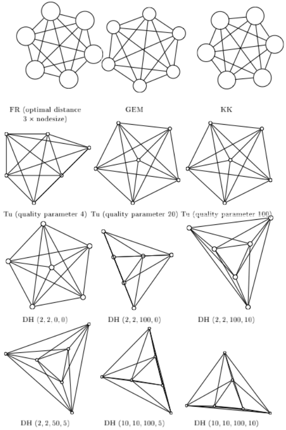
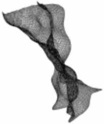


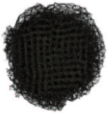
Fig. 5. Several drawings of the K_5

Comparative Analysis: Hachul and Jünger 2005

- Algorithms: FR*, KK*, GRIP, HDE, FM3, ACE
- Measures: running time, aesthetics of layouts
- Dataset: 30 graphs from earlier papers and synthetic ones
- Graph sizes: 930-143,437 vertices and 970-184,532 edges
- Conclusions:
  - multi-scale algorithms work well
  - algebraic methods work fast
  - FM3 slower but better



(c) GRIP

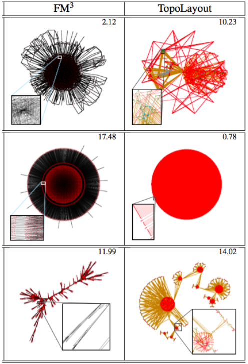


(f) GVA

(i) GVA

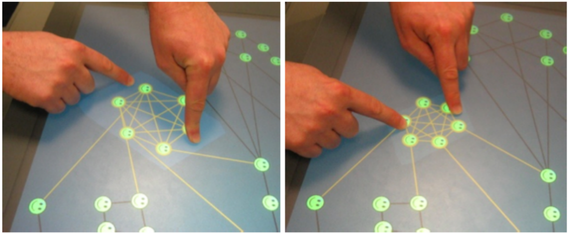
Comparative Analysis: Archambault et al. 2007

- Algorithms: GRIP, HDE, FM3, ACE, TopoLayout
- Measures: running time, structure of layouts
- Dataset: 11 graphs (from earlier papers + tough real world graphs)
- Graph sizes: 1,104-77,251 vertices and 3,232-191,659 edges
- Conclusions:
  - ACE, HDE: only for mesh-like graphs
  - FM3 and TopoLayout are better

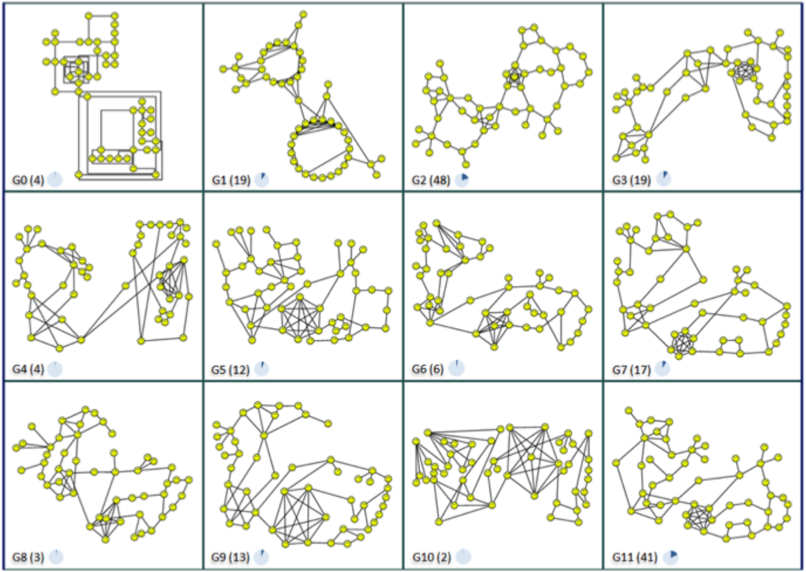


Comparative Analysis: Dwyer et al. 2009

- User-generated layouts
- Automatic layout (orthogonal, circular, force-directed)
- 32/194 users: draw a graph with 50 nodes and 74/77 links
- Tasks: find a K_6, P_4, a cut node, and leaf nodes
- Measures: task accuracy and time, stress, edge crossings
- Conclusions
  - user-generated better than orthogonal/circular aut. layout
  - the most popular layout overall: aut. force-directed layout

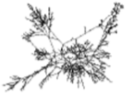


Comparative Analysis: Dwyer et al. 2009



Comparative Analysis: Brandes and Pich 2009

- Algorithms: FM3, HDE, GRIP, MDS (strain, stress)
- Dataset: 10 graphs with 500-3,000 vertices and 700-15,000 edges
- Measures: stress
- Conclusions
  - graph-theoretic distance models work well
  - the very fast MDS methods can hide structure
  - fast MDS as initial layout improves the results



Comparative Analysis: Brandes and Pich 2009

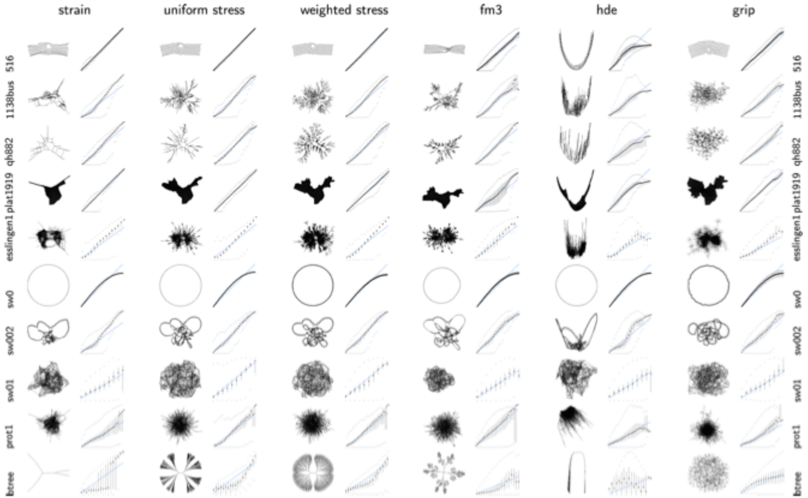
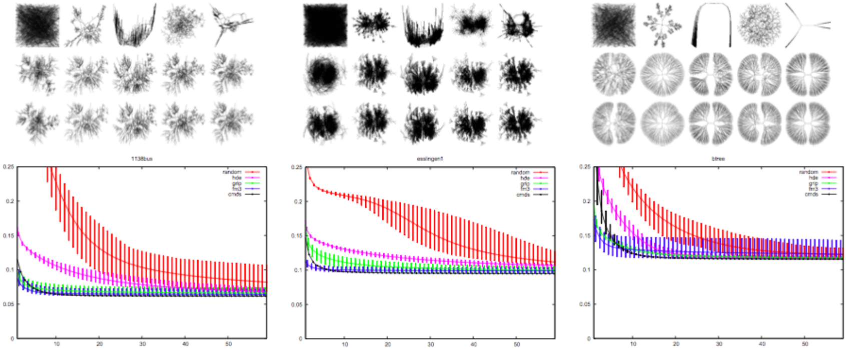


Fig. 2. Drawings the test graphs, and quartile plots of d_{ij} (abscissa) vs. $\|x_i - x_j\|$. Large dots indicate the median, small dots minimum and maximum, and black lines the range of the two middle quartiles (25–75 per cent). The thin blue line with slope 1 is a visual aid.

Comparative Analysis: Brandes and Pich 2009



(a) 1138bus (b) esslingen1 (c) btree

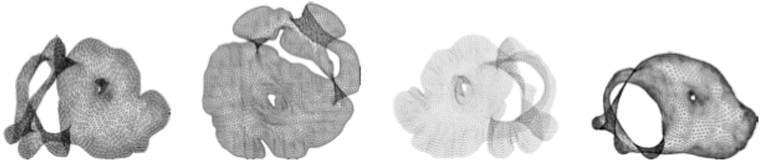
Fig. 3. Upper row: The majorization process with different initializations random, fm3, hde, grip, cmds after 0, 30, 60 iterations. Lower row: Number of iterations vs. stress. The bars indicate the range of values, the dots the median value, in 25 runs.

Comparative Analysis: Bartel et al. 2010

- Algorithms: multi-scale methods, different coarsening/refining
- Layout: EAD, FR, KK, GEM, NMM, FME, DH
- Dataset: 43 graphs, 34-16,000 vertices and 78-48,000 edges
- Measures: running time, edge lengths, crossings
- Conclusions
  - no clear winner
  - EAD, NMM, FME consistently good
  - KK is best for edge lengths and structure, but too costly
  - some coarsening steps consistently good
  - graph-theoretic distance models work well
  - the very fast MDS methods can hide structure
  - fast MDS as initial layout improves the results

(a) KK (b) NMM (c) FME

Comparative Analysis: Bartel et al. 2010

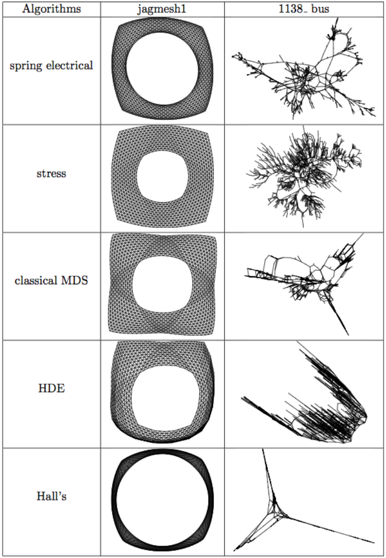


(a) BP,MM,KKFR (b) MP,SM,FME (c) KKS (d) EADS

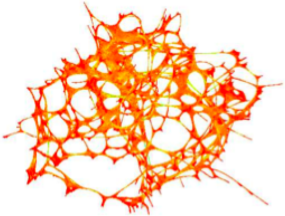
Fig. 2: Layout of the planar graph *3elt* computed with slowest(BP,MM,KKFR, 15 levels, 91s) and fastest overall combination (MP,SM,FME, 5 levels, 2,27s) compared to results of fastest (EADS, 2,77s) and slowest (KKS, 158s) single level methods .

Survey: Hu 2011

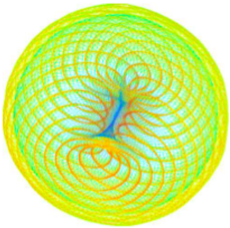
- Algorithms: FR, HDE, MDS (strain, stress)
- Dataset: 2,272 graphs (UF sparse matrices) with up to 27M nodes
- Measures: stress and appearance
- Conclusions: HDE, PivotMDS, LandmarkMDS have trouble with sparse graphs



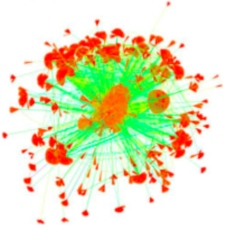
Survey: Hu 2011



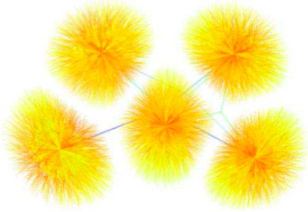
boneS10. $|V| = 914898, |E| = 27276762$



cvxbqp1. $|V| = 40000, |E| = 120000.$



connectus. $|V| = 392366, |E| = 1124842$

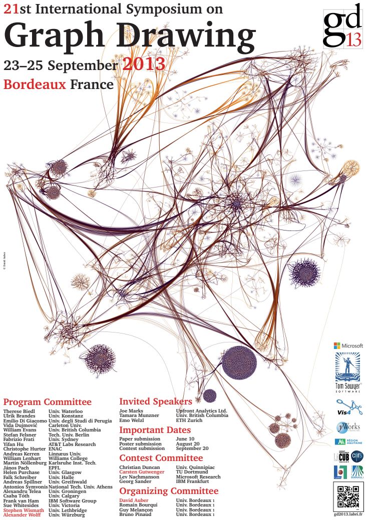


aircraft. $|V| = 11271, |E| = 20267.$

Figure 9: Drawing of some graphs from the University of Florida Sparse Matrix Collection [10]. More drawings can be found at [33].

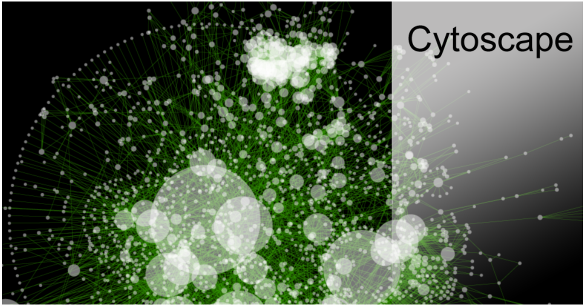
GD Libraries

- GraphViz
- OGDF
- MSAGL
- VTK
- Prefuse
- Tulip
- ...



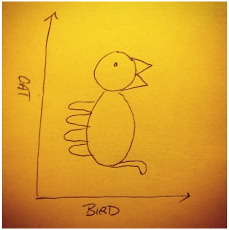
Drawing Software

- TouchGraph
- NodeXL
- Pajek
- Cytoscape
- TopFish
- Gephi
- yEd
- ...



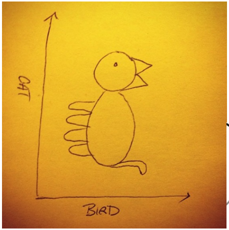
When is a Drawing Good?

- “you will know it when you see it”



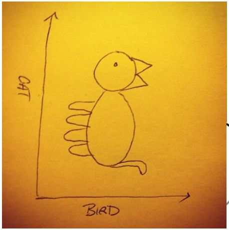
When is a Drawing Good?

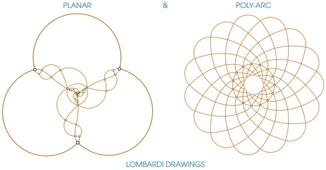
- "you will know it when you see it"
- quantitative measures (e.g., stress, precision/recall)



When is a Drawing Good?

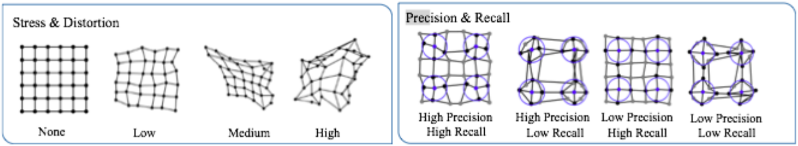
- "you will know it when you see it"
- quantitative measures (e.g., stress, precision/recall)
- qualitative measures (e.g., "aesthetically pleasing")





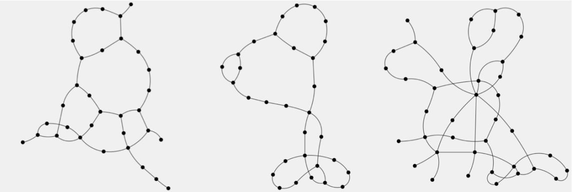
Quantitative measures

- Stress: the classic evaluation function used for MDS
- Precision/Recall measure:
  - false positives (low similarity but small Euclidean distance)
  - false negatives (high similarity but large Euclidean distance)
- Distortion: distances b/n pairs of nodes compared to desired distances



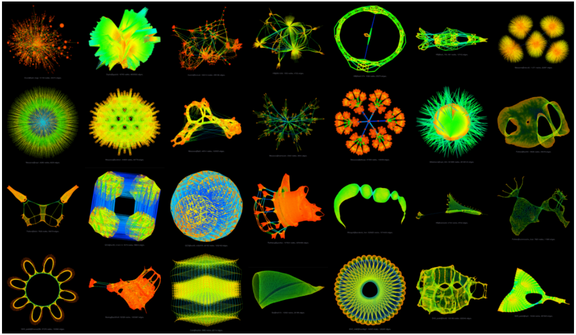
Qualitative measures

- Legibility/readability of the visualization: can one find certain nodes, edges, paths, clusters?
- Utility: which visualizations provide actionable insights and/or prompt more meaningful questions, or are otherwise useful for human decision-making?
- Engagement and enjoyment: beyond "do you like this drawing?"
- Memorability: how well is the shown data remembered?



Datasets

- Rome library: about 11K graphs with 10-100 vertices; 8K are non-planar
- Stanford library (2004): contains MS IM network from 2006 with 240 million nodes and 1.3 billion edges
- Florida Sparse Matrix library: 2.5K graphs with up to 118M nodes and 2B edges



References I

D. Archambault, T. Munzner, and D. Auber.
Topolayout: Multilevel graph layout by topological features.
Visualization and Computer Graphics, IEEE Transactions on, 13(2):305–317, March 2007.

D. Auber, D. Archambault, R. Bourqui, A. Lambert, M. Mathiaut, P. Mary, M. Delest, J. Dubois, and G. Mélançon.
The Tulip 3 framework: A scalable software library for information visualization applications based on relational data.
2012.

G. Bartel, C. Gutwenger, K. Klein, and P. Mutzel.
An experimental evaluation of multilevel layout methods.
In *Proceedings of the 18th International Conference on Graph Drawing*, GD'10, pages 80–91, Berlin, Heidelberg, 2011. Springer-Verlag.

F. Brandenburg, M. Himsolt, and C. Rohrer.
An experimental comparison of force-directed and randomized graph drawing algorithms.
In *Graph Drawing*, pages 76–87, 1996.

U. Brandes and C. Pich.
Eigensolver methods for progressive multidimensional scaling of large data.
In *Proceedings 14th Symposium on Graph Drawing (GD)*, pages 42–53, 2006.

U. Brandes and C. Pich.
An experimental study on distance-based graph drawing.
In *Graph Drawing*, pages 218–229. Springer, 2009.

R. Chernobelskiy, K. I. Cunningham, M. T. Goodrich, S. G. Kobourov, and L. Trott.
Force-directed Lombardi-style graph drawing.
In *Graph Drawing*, pages 320–331, 2011.

References II

M. Chimani, C. Gutwenger, M. Jünger, G. W. Klau, K. Klein, and P. Mutzel.
The open graph drawing framework (OGDF).
Handbook of Graph Drawing and Visualization, pages 543–569, 2011.

R. Davidson and D. Harel.
Drawing graphs nicely using simulated annealing.
ACM Transactions on Graphics, 15(4):301–331, Oct. 1996.

W. De Nooy, A. Mrvar, and V. Batagelj.
Exploratory social network analysis with Pajek, volume 27.
Cambridge University Press, 2011.

T. Dwyer, Y. Koren, and K. Marriott.
Constrained graph layout by stress majorization and gradient projection.
Discrete Mathematics, 309(7):1895–1908, 2009.

T. Dwyer, B. Lee, D. Fisher, K. I. Quinn, P. Isenberg, G. G. Robertson, and C. North.
A comparison of user-generated and automatic graph layouts.
IEEE Trans. Vis. Comput. Graph., 15(6):961–968, 2009.

T. Dwyer, K. Marriott, and M. Wybrow.
Topology preserving constrained graph layout.
In *Graph Drawing*, pages 230–241, 2008.

P. Eades.
A heuristic for graph drawing.
Congressus Numerantium, 42:149–160, 1984.

J. Ellson, E. R. Gansner, E. Koutsofios, S. C. North, and G. Woodhull.
Graphviz - open source graph drawing tools.
In *Graph Drawing*, pages 483–484, 2001.

References III

A. Frick, A. Ludwig, and H. Mehldau.
A fast adaptive layout algorithm for undirected graphs.
In R. Tamassia and I. G. Tollis, editors, *Proceedings of the 2nd Symposium on Graph Drawing (GD)*, volume 894 of *Lecture Notes in Computer Science*, pages 388–403. Springer-Verlag, 1995.

Y. Frishman and A. Tal.
Dynamic drawing of clustered graphs.
pages 191–198.

T. Fruchterman and E. Reingold.
Graph drawing by force-directed placement.
Softw. – Pract. Exp., 21(11):1129–1164, 1991.

P. Gajer, M. T. Goodrich, and S. G. Kobourov.
A fast multi-dimensional algorithm for drawing large graphs.
Computational Geometry: Theory and Applications, 29(1):3–18, 2004.

P. Gajer and S. G. Kobourov.
GRIP: Graph dRawing with Intelligent Placement.
Journal of Graph Algorithms and Applications, 6(3):203–224, 2002.

E. Gansner, Y. Koren, and S. North.
Graph drawing by stress minimization.
In *Proceedings 12th Symposium on Graph Drawing (GD)*, pages 239–250, 2004.

S. Hachul and M. Jünger.
Drawing large graphs woth a potential-field-based multilevel algorithm.
In *Proceedings of the 12th Symposium on Graph Drawing (GD)*, volume 3383 of *Lecture Notes in Computer Science*, pages 285–295. Springer-Verlag, 2004.

References IV

S. Hachul and M. Jünger.
An experimental comparison of fast algorithms for drawing general large graphs.
In *Proceedings of the 13th International Conference on Graph Drawing*, GD'05, pages 235–250, Berlin, Heidelberg, 2006. Springer-Verlag.

R. Hadany and D. Harel.
A multi-scale algorithm for drawing graphs nicely.
Discrete Applied Mathematics, 113(1):3–21, 2001.

D. Hansen, B. Shneiderman, and M. A. Smith.
Analyzing social media networks with NodeXL: Insights from a connected world.
Morgan Kaufmann, 2010.

D. Harel and Y. Koren.
A fast multi-scale method for drawing large graphs.
Journal of graph algorithms and applications, 6:179–202, 2002.

D. Harel and Y. Koren.
Graph drawing by high-dimensional embedding.
In *Graph Drawing*, pages 207–219. Springer, 2002.

D. Harel and M. Sardas.
Randomized graph drawing with heavy-duty preprocessing.
J. Visual Lang. Comput., 6(3):233–253, 1995.

D. Harel and M. Sardas.
An algorithm for straight-line drawing of planar graphs.
Algorithmica, 20(2):119–135, 1998.

References V

J. Heer, S. K. Card, and J. A. Landay.
Prefuse: a toolkit for interactive information visualization.
In *Proc. SIGCHI conference on Human factors in computing systems*, pages 421–430. ACM, 2005.

Y. Hu.
Efficient and high quality force-directed graph drawing.
The Mathematica Journal, 10:37–71, 2005.

T. Kamada and S. Kawai.
An algorithm for drawing general undirected graphs.
Inform. Process. Lett., 31:7–15, 1989.

S. G. Kobourov.
Force-directed drawing algorithms.
Handbook of Graph Drawing and Visualization, pages 383–408, 2013.

Y. Koren, L. Carmel, and D. Harel.
Ace: A fast multiscale eigenvectors computation for drawing huge graphs.
In *Information Visualization, 2002. INFOVIS 2002. IEEE Symposium on*, pages 137–144. IEEE, 2002.

J. Kruskal and J. Seery.
Designing network diagrams.
In *Proceedings 1st General Conference on Social Graphics*, pages 22–50, 1980.

J. B. Kruskal.
Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis.
Psychometrika, 29:1–27, 1964.

M. J. M. Bastian, S. Heymann.
Gephi: an open source software for exploring and manipulating networks.
International AAAI Conference on Weblogs and Social Media, 2009.

References VI

K.-L. Ma and C. W. Muelder.
Large-scale graph visualization and analytics.
Computer, 46(7):39–46, 2013.

S. Martin, W. M. Brown, R. Klavans, and K. W. Boyack.
Openord: an open-source toolbox for large graph layout.
volume 7868, pages 786806–786806–11, 2011.

A. Noack.
Energy models for graph clustering.
J. Graph Algorithms Appl., 11(2):453–480, 2007.

A. Nocaj, M. Ortmann, and U. Brandes.
Untangling hairballs.
In *Graph Drawing*, pages 101–112. Springer, 2014.

A. Quigley and P. Eades.
FADE: graph drawing, clustering, and visual abstraction.
In *Proceedings of the 8th Symposium on Graph Drawing (GD)*, volume 1984 of *Lecture Notes in Computer Science*, pages 197–210. Springer-Verlag, 2000.

P. Shannon, A. Markiel, O. Ozier, N. S. Baliga, J. T. Wang, D. Ramage, N. Amin, B. Schwikowski, and T. Ideker.
Cytoscape: a software environment for integrated models of biomolecular interaction networks.
Genome research, 13(11):2498–2504, 2003.

D. Tunkelang.
JIGGLE: Java interactive general graph layout environment.
In *Graph Drawing*, volume 1547 of *Lecture Notes in Computer Science*, pages 413–422. Springer-Verlag, 1998.

References VII

T. Von Landesberger, A. Kuijper, T. Schreck, J. Kohlhammer, J. J. van Wijk, J.-D. Fekete, and D. W. Fellner.
Visual analysis of large graphs: State-of-the-art and future research challenges.
In *Computer graphics forum*, volume 30, pages 1719–1749. Wiley Online Library, 2011.

C. Walshaw.
A multilevel algorithm for force-directed graph drawing.
Journal of Graph Algorithms and Applications, 7(3):253–285, 2003.

R. Wiese, M. Eiglsperger, and M. Kaufmann.
yFiles: Visualization and automatic layout of graphs.
In *Graph Drawing*, pages 453–454, 2001.

Questions

- Quantitative measures beyond time, stress, neighborhood preservation, drawing area

Questions

- Quantitative measures beyond time, stress, neighborhood preservation, drawing area
- Qualitative measures (aesthetics) beyond edge crossings

Questions

- Quantitative measures beyond time, stress, neighborhood preservation, drawing area
- Qualitative measures (aesthetics) beyond edge crossings
- Validated datasets and benchmarks

Questions

- Quantitative measures beyond time, stress, neighborhood preservation, drawing area
- Qualitative measures (aesthetics) beyond edge crossings
- Validated datasets and benchmarks
- Real-world tasks for user studies

Questions

- Quantitative measures beyond time, stress, neighborhood preservation, drawing area
- Qualitative measures (aesthetics) beyond edge crossings
- Validated datasets and benchmarks
- Real-world tasks for user studies
- Real-world graphs, with labels and weights