Graphs

- **Types**
  - undirected graphs
  - directed graphs or digraphs
- **Examples**
  - networks and wiring diagrams
  - finite state machines
  - dependencies
    - PERT chart
    - evolution or derivation
  - procedure call graphs
  - data structures involving pointers
  - entity-relationship diagrams

Resources

"Algorithms for drawing graphs: an annotated bibliography,"
G. Di Battista, P. Eades, R. Tamassia, I.G. Tolis
`http://www.cs.brown.edu:80/people/rt/gd-biblio.html`

"How to visualize a graph: specification and algorithms,"
I. F. Cruz and R. Tamassia, Course Notes
`http://www.cs.brown.edu:80/people/rt/gd-tutorial.html`

"Applications of graph visualization,"
S.C. North, E. Koutsofios, GI '94
`ftp://ftp.research.att.com/dist/drawdag`

"Drawing graphs with *dot*," E. Koutsofios, S.C. North
`http://www.research.att.com:80/sw/tools/reuse`

Graph Drawing '93, '94, '95, '96

See course web page for additional pointers ...

Aesthetics

- **Favor symmetry and balance**
- **Expose hierarchical structures**
  - Aim edges in the same direction
  - Identify sources and sinks
  - Use tree drawing algorithms
- **Avoid clutter**
  - Minimize edge crossings
  - Straighten edges, minimize bends
  - Keep edges short
  - Maximize smallest angle between edges
  - Compact (minimize area)
- **Cluster subgraphs**
  - Group related nodes
  - Avoid long edges

Techniques

- **Positions determined by attributes**
  - geographic layout
  - dominant variable (time)
- **Multidimensional scaling**
  - symmetric interconnection matrix
  - compute scaling matrix
- **Planar graphs**
- **Hierarchical layout by layers**
- **Spring system**
- **3D**

Planar Graphs

- **Planarity can be detected in *O(n)* time**
  - **Hopcroft, Tarjan 74**
- **All PG may be drawn with straight lines**
  - **Fary 48, Stein 51, Steinitz 34, Wagner 36**
- **Planarization is hard**
  - **Maximal planar subgraph is NP-hard**
  - **Minimim number of edges crossings is NP-hard**

Layering

---

4 step procedure:

1. **Assign vertices to layers so that arcs are directed downward and vertices are evenly distributed**
2. **Select permutation of vertices at each layer to minimize edge crossings**
3. **Route edges to minimize bends**
4. **Tune the final drawing (create splines)**

Carpano 80

Sugiyama Tagawa Toda 81

North 88

Springs

---

3 step procedure:

1. Replace the edges by springs with length and spring constant
2. Assign vertices to an initial position
3. Iterate the system according to the laws of motion

Enhancements:

– vertex distribution
– edge crossings
– walls and obstacles
– contrain vertices, edges
– simulated anealing (add randomness)

Original reference: Eades 84

Statistical Graphics

A Short Picture Tour ...

- **Scatterplot**
  - Reeds, Shields, Scullin, Tavera, Elford, Fig. 2
- **Parallel coordinates**
  - Keim p. 15
- **Worlds within worlds**
  - Feiner Beshers 91
- **Dot plots**
  - Cleveland p. 150
- **Time series**
  - Cleveland p. 182, 191
- **Trellis**
  - Becker, Cleveland, Shyu p. 129

See *The Elements of Graphic Data,* Cleveland, 1993