Networks → any collection of objects connected by links

• Visual clutter makes visualization useless.

• Arrange Tables → • Express Values • Separate, order, Align regions.
• Axis rotation. (rectilinear, parallel, radial)

• Networks → • Online Social network → fb, twitter

• Information network → Connects various information through hyperlinks

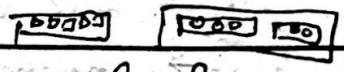
• Computer Networks → Internet & wireless connections, enabling communication

• Visualizing Relations → • Visualizing data
• Visualizing relations
• overall structure of data.

• Ways to represent relations? → • Node-Link diagrams (networks, trees)
• Adjacency Matrix (network, trees)
• Enclosure (trees)

• limitation is edge-crossing

can co-exist



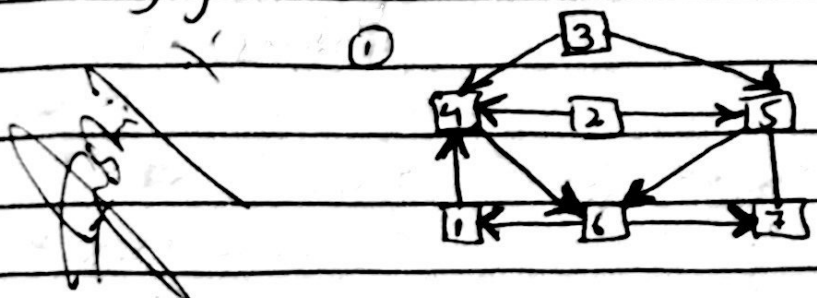
• Graphs → • well suited for topology-related problem.

• Position nodes & edges → • avoid clutter
• maintain appropriate relations

• Major graph visualization techniques
  • Sugiyama-Style layout
  • Force-directed Layout
  • Matrix diagram.

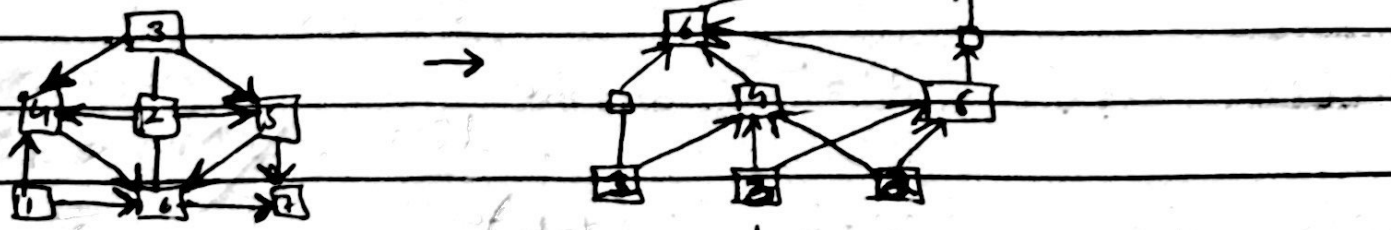
CLCVE

• Sugiyama.



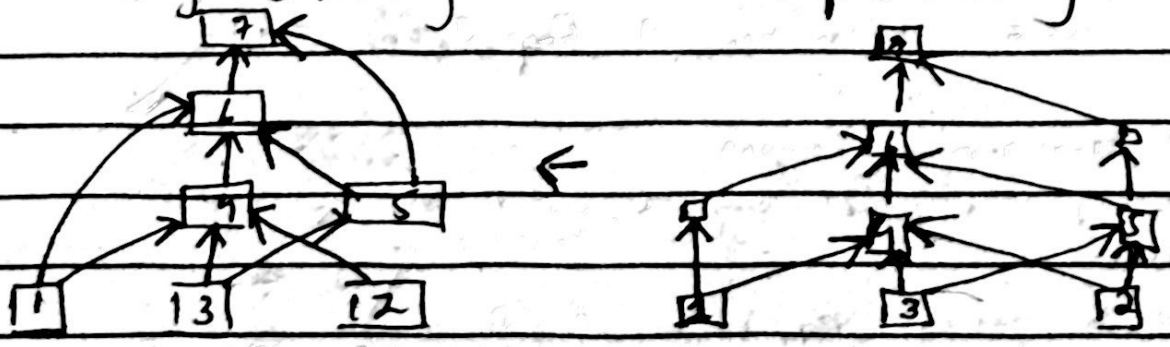
Break cycle

Leveling / Crossing minimization



Edge drawing

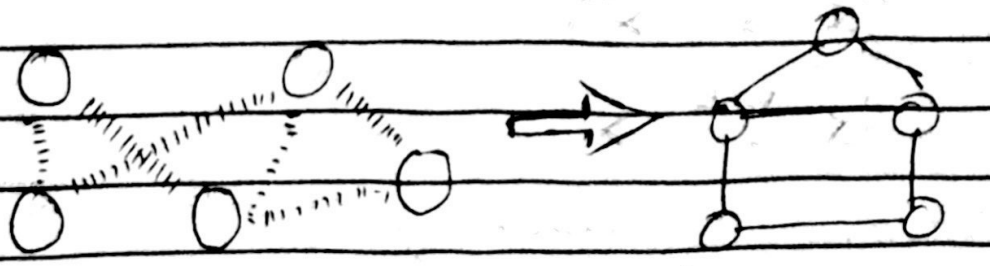
Vertex positioning



- Very flexible, aesthetic
- Can add custom forces
- easy to implement

- Force directed layout

Edges = Springs
Nodes = Repulsive particles.



Repulsion force: $f_r(d) = C_r \cdot m_1 \cdot m_2 / d^2$ (constant, masses)

Attraction force: $f_a(d) = C_a (d-L)$

Total force = Attraction + Repulsion

- Algo:

① Start from random layout
② For every node pair compute attractive & repulsion forces.
③ Total forces per node. (A, B, C)

loop

A C
A B → Repulsion
B C

A B → Attraction
B C

A & B, C say if connected and add all.

④ To answer above ③ step main aye just move every node by it. and it will be its new position.

Stops when layout is good

Problem arises when computing repulsive forces b/w all pairs of nodes

N-Body force Challenge

- Naive approach requires $O(V^2)$ operations (expensive for large graph) — V = no. of nodes

We can approx force calculations using index to achieve $O(V \log V)$

- Barnes-Hut algorithm → • approx force calculations to reduce complexity to $O(V \log V)$ by
  - grouping distant nodes
  - Approximation Parameter (θ)
  - Quadtrees (or octrees)

  - whether a group of distant objects can be treated as a single unit.
  - Large θ → treat more groups as single units

Matrix Diagram

- Adjacency Matrix (alternate to node-link diagram)
  - Change network to tabular data.
  - nodes are keys, edges are values.

- Clique → • graph where every pair of node is connected by an edge.

Example:- A, B, C, D knows each other.

Matrix can be

0	1	1	1
1	0	1	1
1	1	0	1
1	1	1	0

⇒ • 1 means there is an edge (connection) b/w 2 nodes
• 0 means no self loop

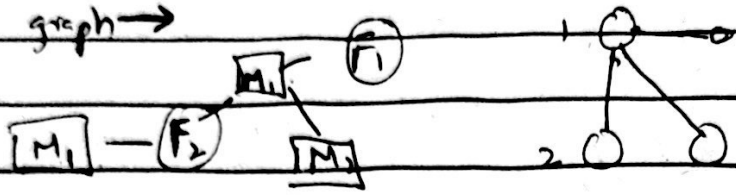
⇒ Adjacency Matrix of clique. Diagonals are 0.

Challenge with large Node-Link diagram

↑

- Attribute-driven layout (values are determined by attributes values)

Pivot graph →



Limitations → • Only 2 variables
• doesn't support continuous variable

- Other node-link layouts.

Orthogonal	Circular	Nested
• great for UML diagram	• Social network diagram	• for hierarchical structure.