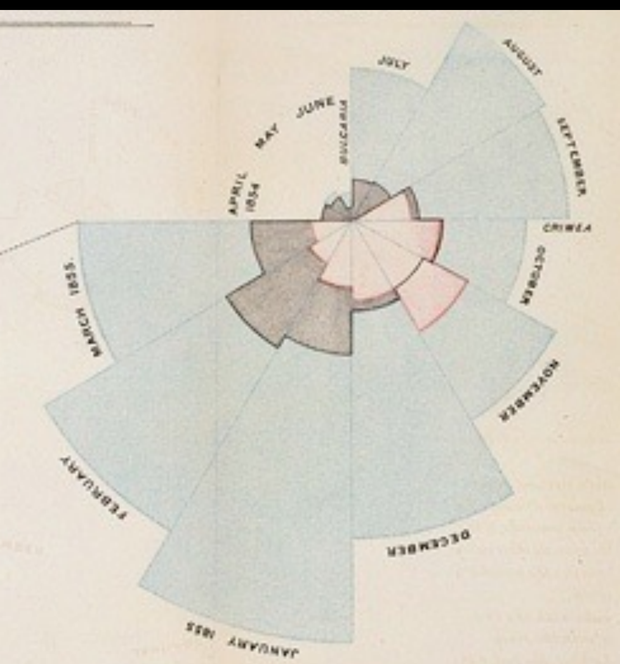
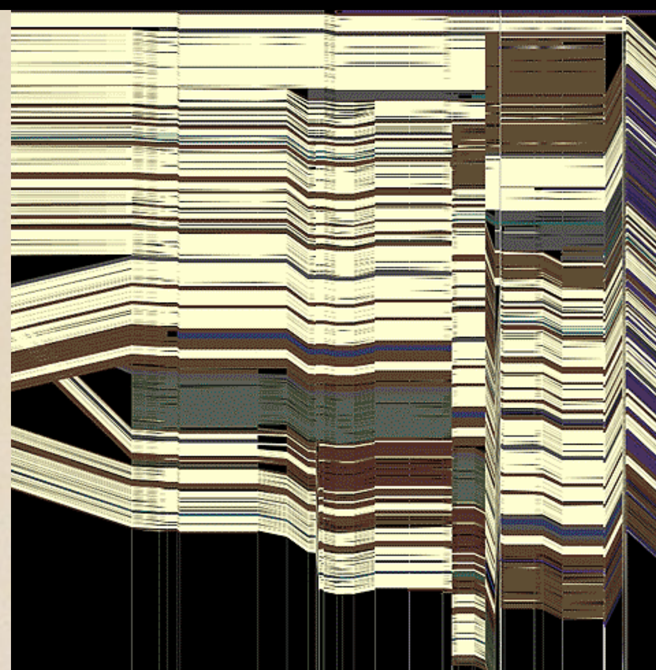
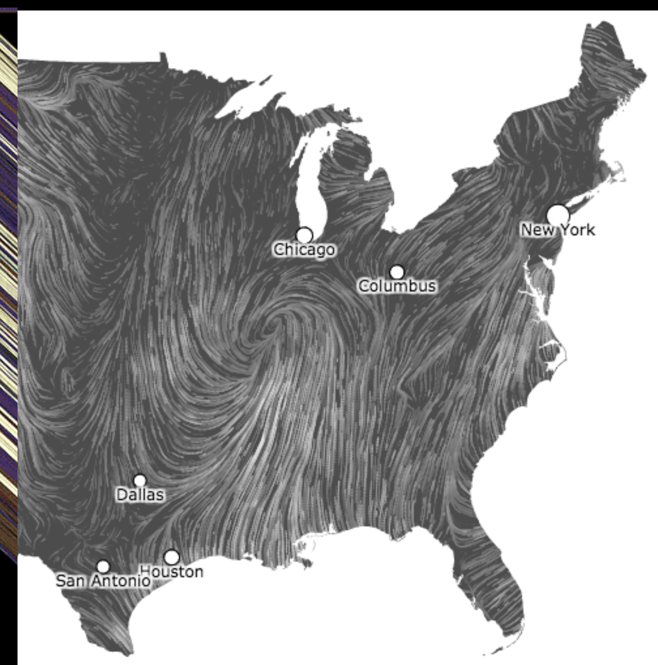
CSE 442 - Data Visualization

Networks



Jeffrey Heer University of Washington

Graphs and Trees

Last Time: Visualizing Hierarchical Data
Today: Visualizing Network Data

Goals
Overview of layout approaches
Assess strengths and weaknesses
Insight into implementation techniques

Applications

Tournaments
Organization Charts
Genealogy
Diagramming (e.g., Visio)
Biological Interactions (Genes, Proteins)
Computer Networks
Social Networks
Simulation and Modeling
Integrated Circuit Design

Topics

- Node-Link Layout Techniques
  - Force-Directed Layout
  - Constraint-Based Layout
- Hierarchical Edge Bundling
- Alternatives to Node-Link Diagrams
  - Matrix Diagrams
  - Attribute-Driven Layout & Hive Plots

Node-Link Layout

Spatial Layout

A primary concern of tree/graph drawing is the spatial arrangement of nodes and edges.

Often (but not always) the goal is to effectively depict the graph structure:

- Connectivity, path-following
- Topological distance
- Clustering / grouping
- Ordering (e.g., hierarchy level)

Spanning Tree Layout

Spanning Tree Layout

Many graphs have useful spanning trees

Websites, Social Networks

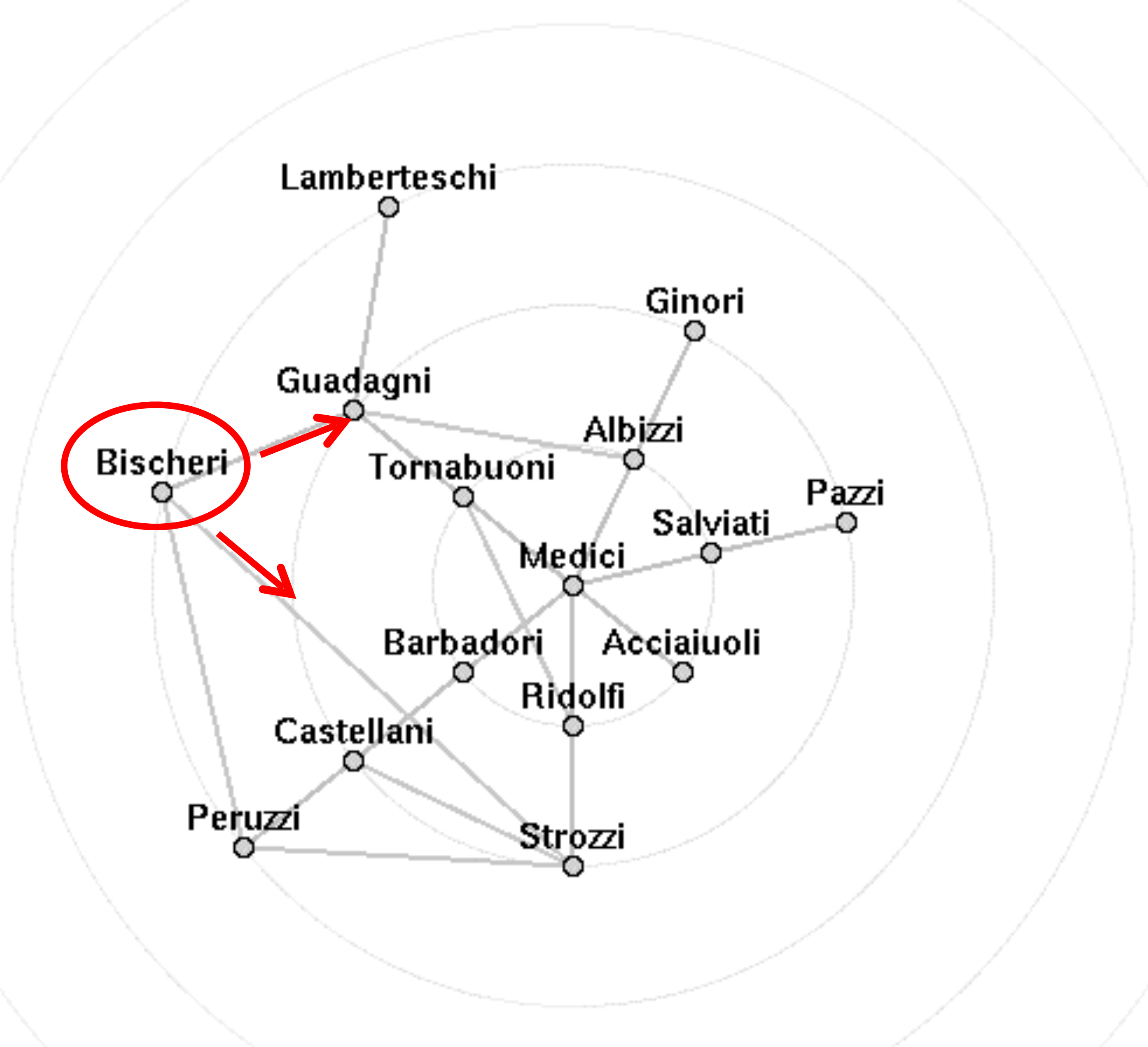
Use tree layout on spanning tree of graph

Trees created by BFS / DFS

Min/max spanning trees

Fast tree layouts allow graph layouts to be recalculated at interactive rates

Heuristics may further improve layout



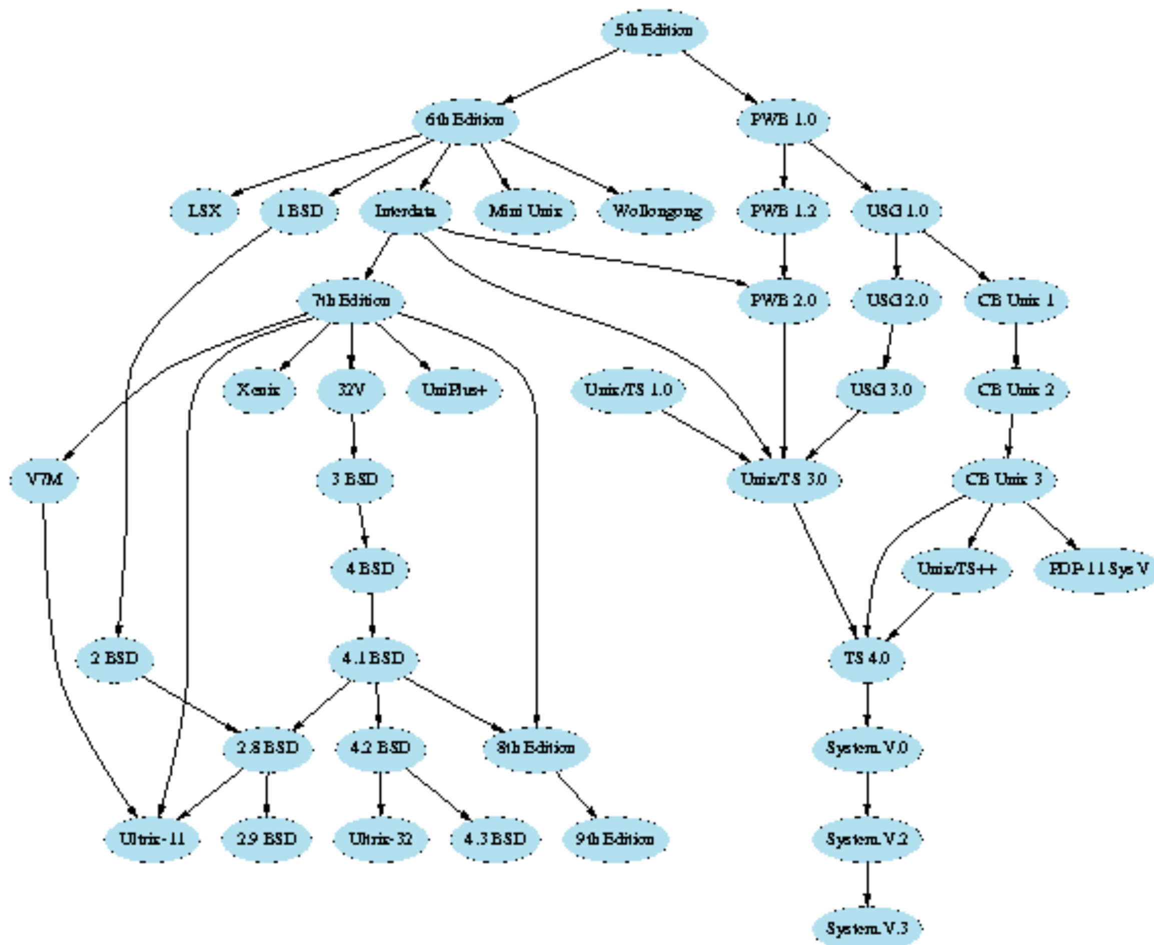
Spanning tree layout may result in arbitrary parent node!

Sugiyama-Style Layout

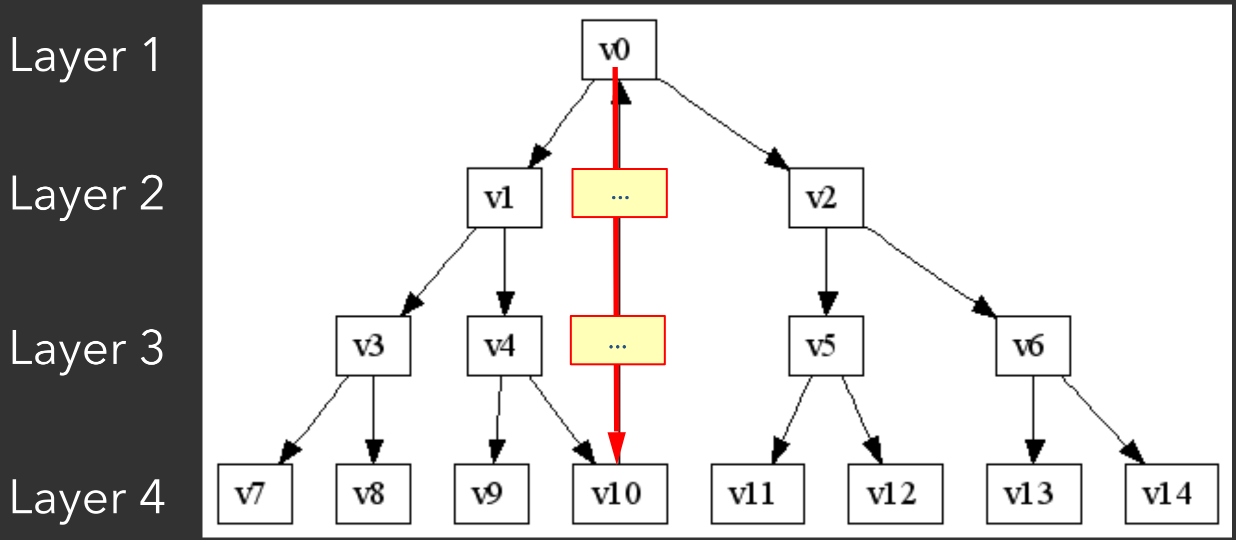
Sugiyama-Style Layout

Evolution of the UNIX operating system

Hierarchical layering based on descent

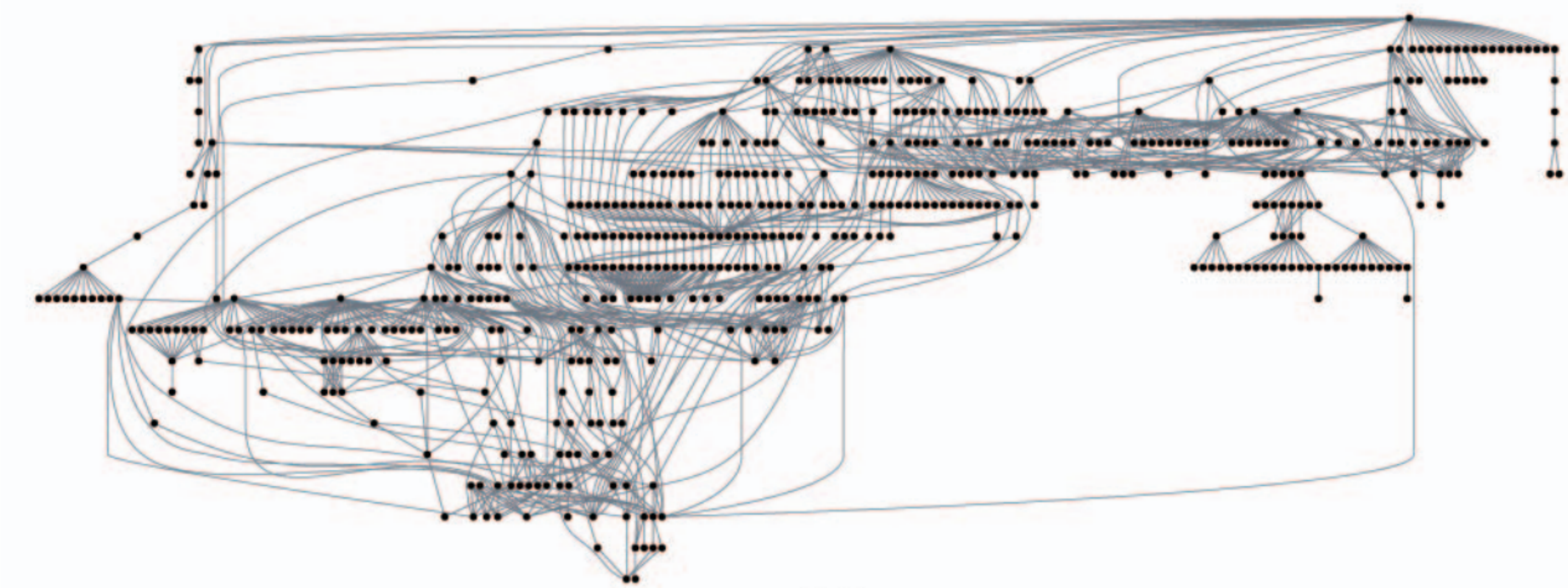


Sugiyama-Style Layout



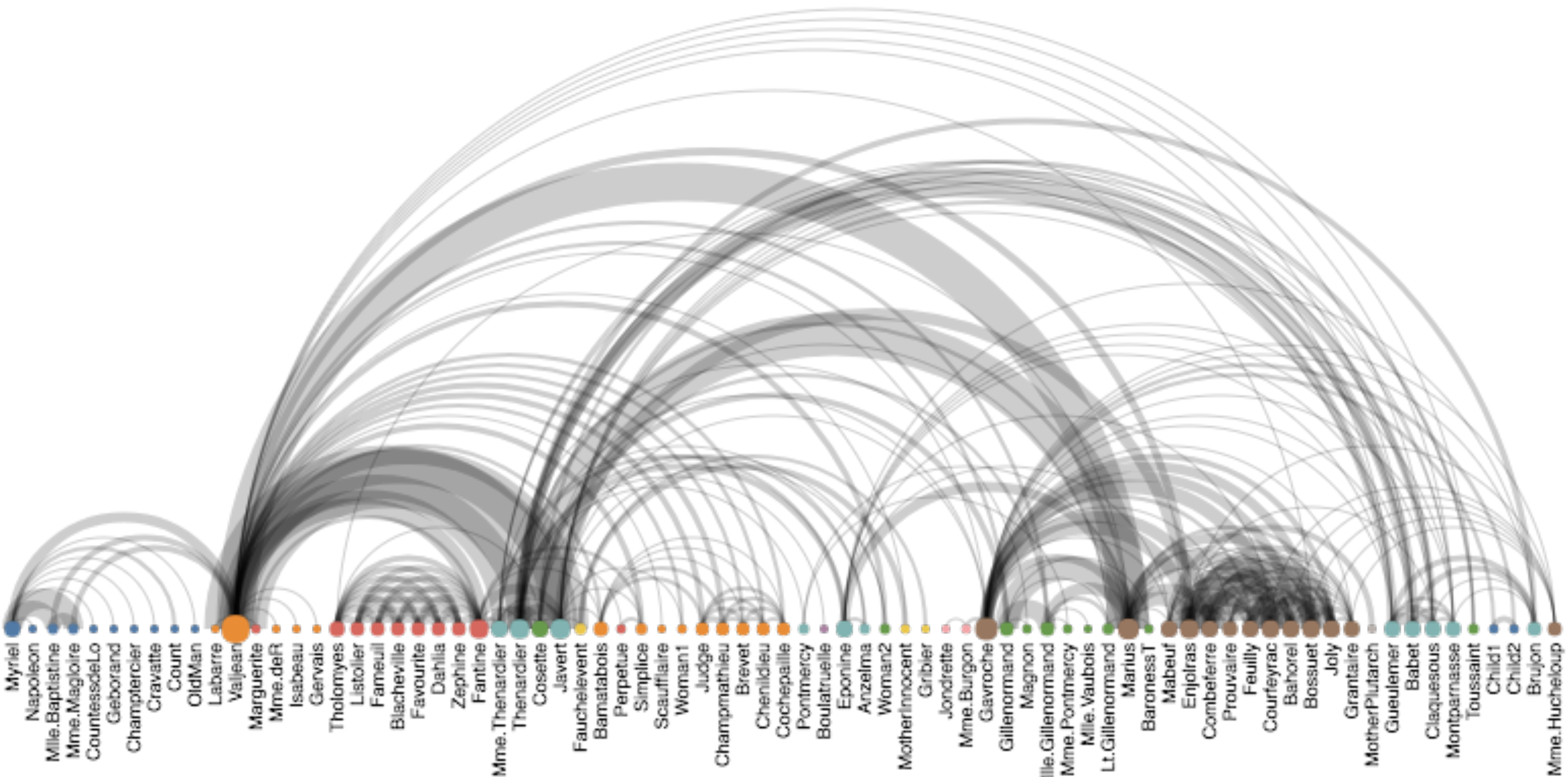
Reverse edges to remove cycles
Assign nodes to hierarchy layers
Create dummy nodes to “fill in” missing layers
Arrange nodes within layer, minimize edge crossings
Route edges – layout splines if needed

Produces Hierarchical Layouts



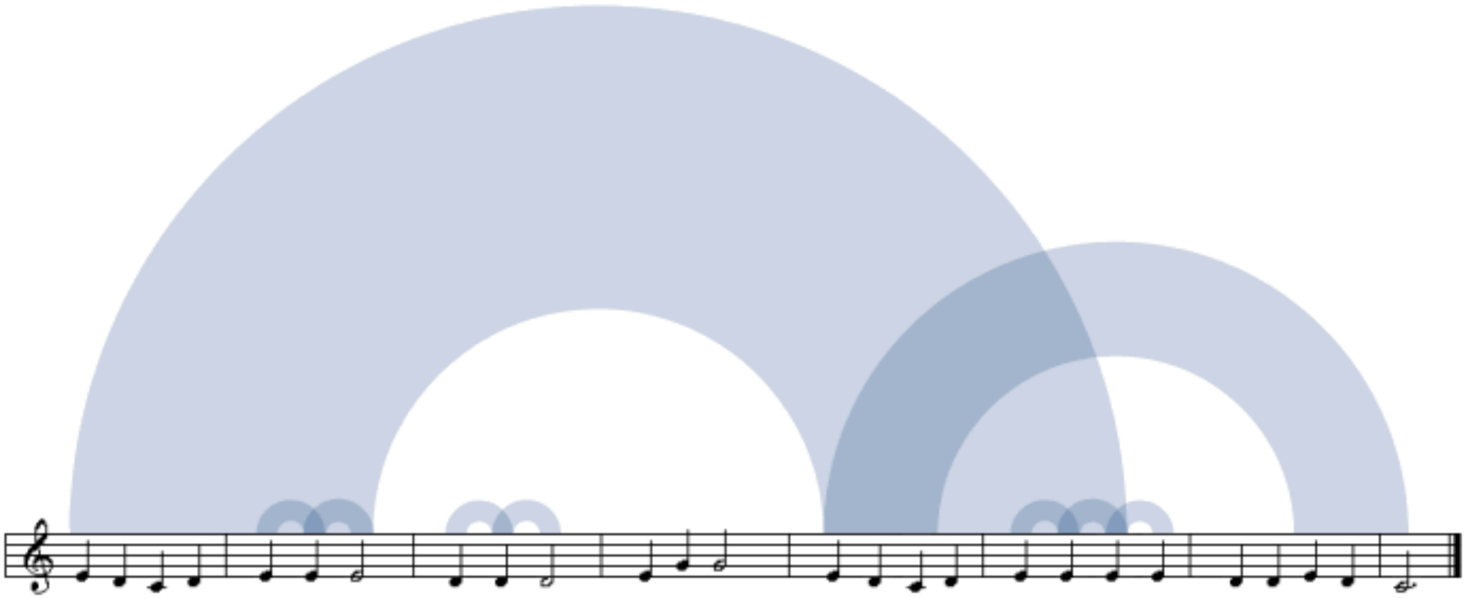
Sugiyama-style layout emphasizes hierarchy.
However, cycles in the graph may mislead.
Long edges can impede perception of proximity.

Arc Diagrams



Linear node layout, circular arcs show connections.

Layout quality sensitive to node ordering!



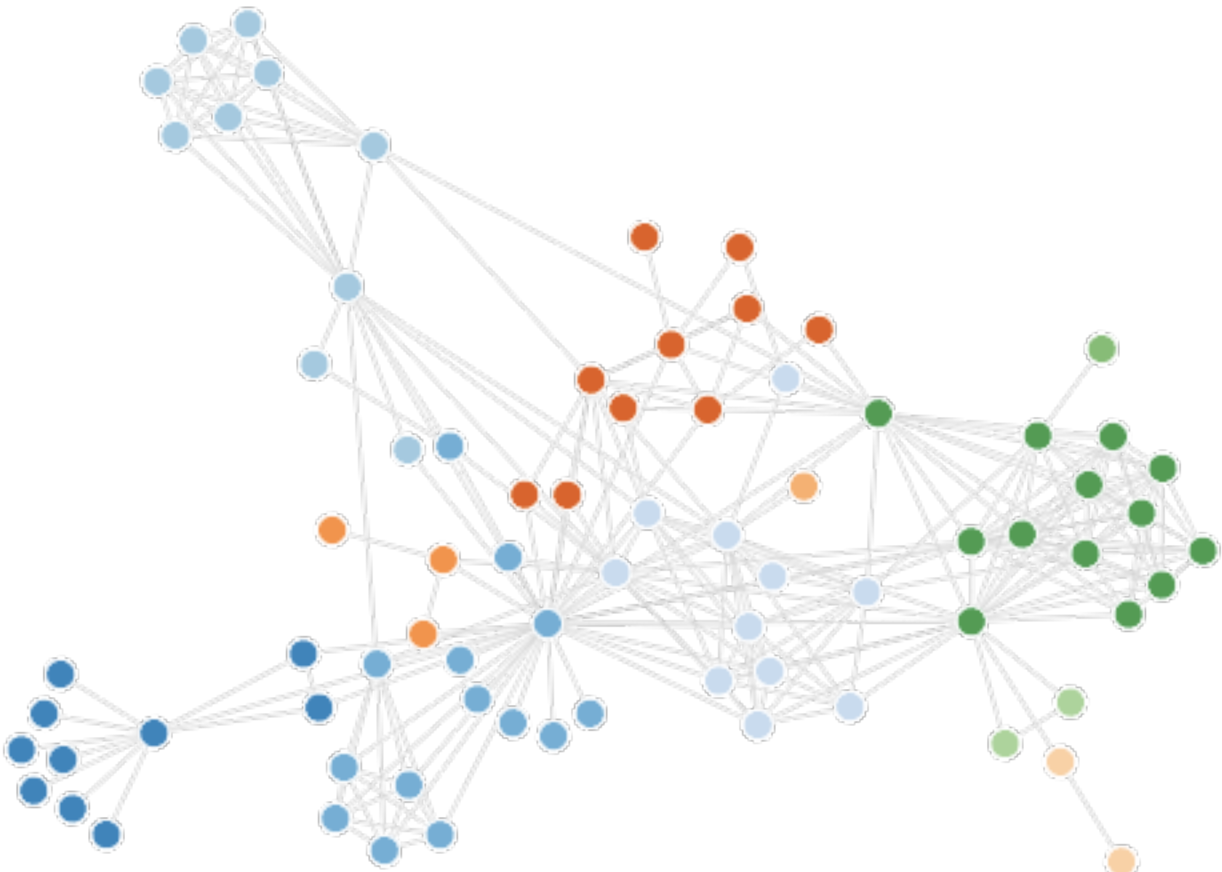
For example, the picture above was built from the first line of a very simple piece: *Mary Had a Little Lamb*. Each arch connects two identical passages. To clarify the connection between the visualization and the song, in this diagram the score is displayed beneath the arches.

The Shape of Song

[Wattenberg '01]

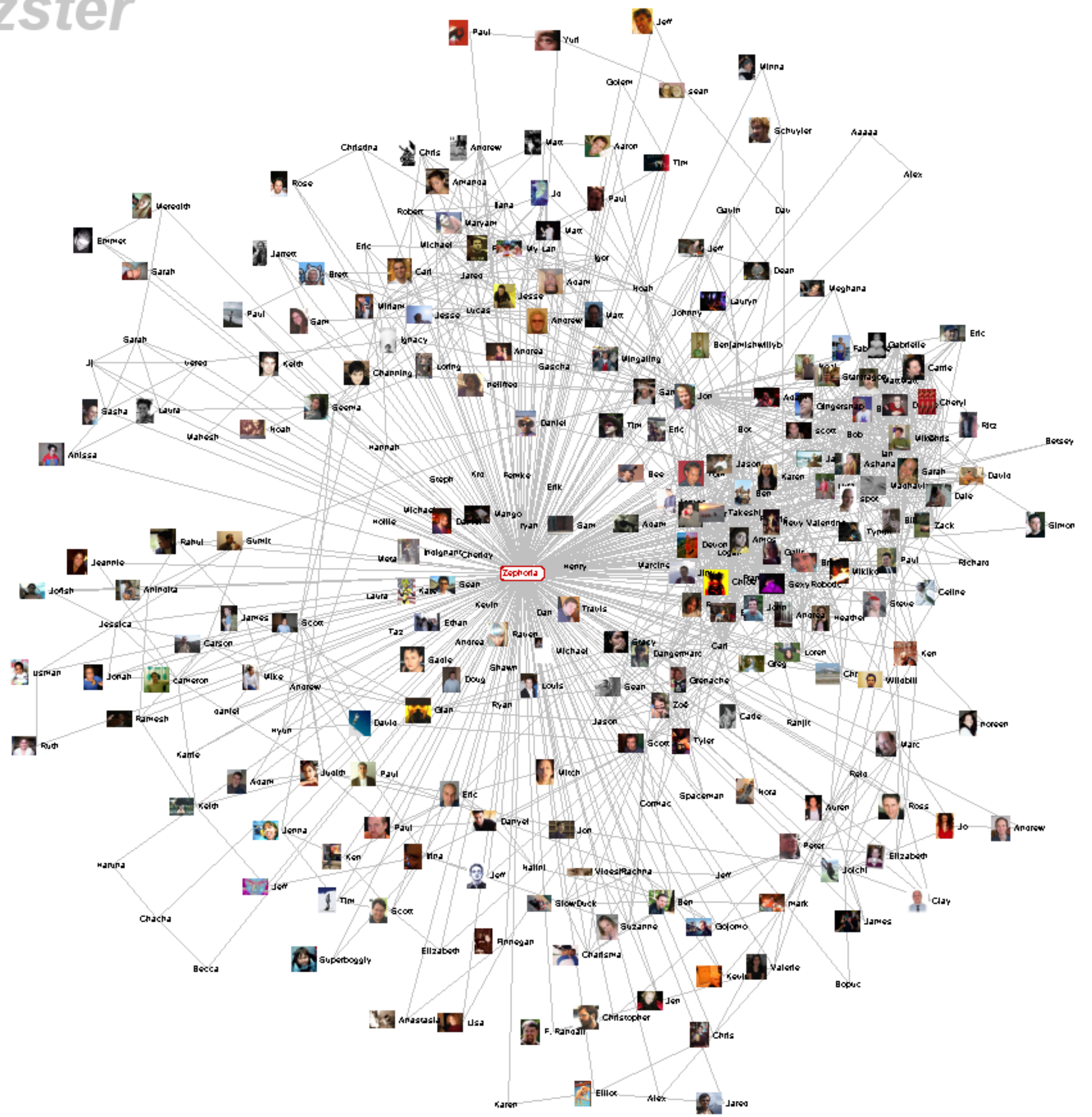
This diagram visualizes the refrain from the folk song *Clementine*. As you would expect, the refrain consists of multiple repetitions of the same passage--and that is exactly what the diagram shows. The score isn't shown in this diagram since the notes would be too small to read.

Force-Directed Layout



Interactive Example: Configurable Force Layout

vizster



community >> Enable

search >>

Zephoria

User ID	21721
Friends	266
Age	??
Gender	Female
Status	Single
Location	San Francisco, CA
Hometown	Lancaster, PA
Occupation	researcher: social networks, identity, context
Interests	apophenia, observing people, culture, questioning power, reading, buddhism, ipseity, computer-mediated communication, social networks, technology, anthropology, stomping
Music	psytrance/goa/trance [Infected Mushroom, Son Kite... Iboga/Digital Structures], Ani Difranco, downtempo, Thievery Corporation, Beth Orton, Morcheeba, Ween, White Stripes
Books	Authors: Erving Goffman, Stanley Milgram, Jeanette Winterson, Eric Schlosser, Leslie Feinberg, Dorothy Allison, Italo Calvino, Hermann Hesse
TV Shows	??
Movies	Koyaanisqatsi, Amelie, Waking Life, Tank Girl, The Matrix, Clockwork Orange, American Beauty, Fight Club, Boys Don't Cry
Member Since	??
Last Login	2003-10-21
Last Updated	2003-10-21
About	[Some know me as danah...] I'm a geek, an activist and an academic, fascinated by people and society. I see life as a very large playground and enjoy exploring its intricacies. I revel in life's chaos, while simultaneously providing my own insane element. My musings: http://www.zephoria.org/thoughts/
Want to Meet	Someone who makes life's complexities seem simply elegant

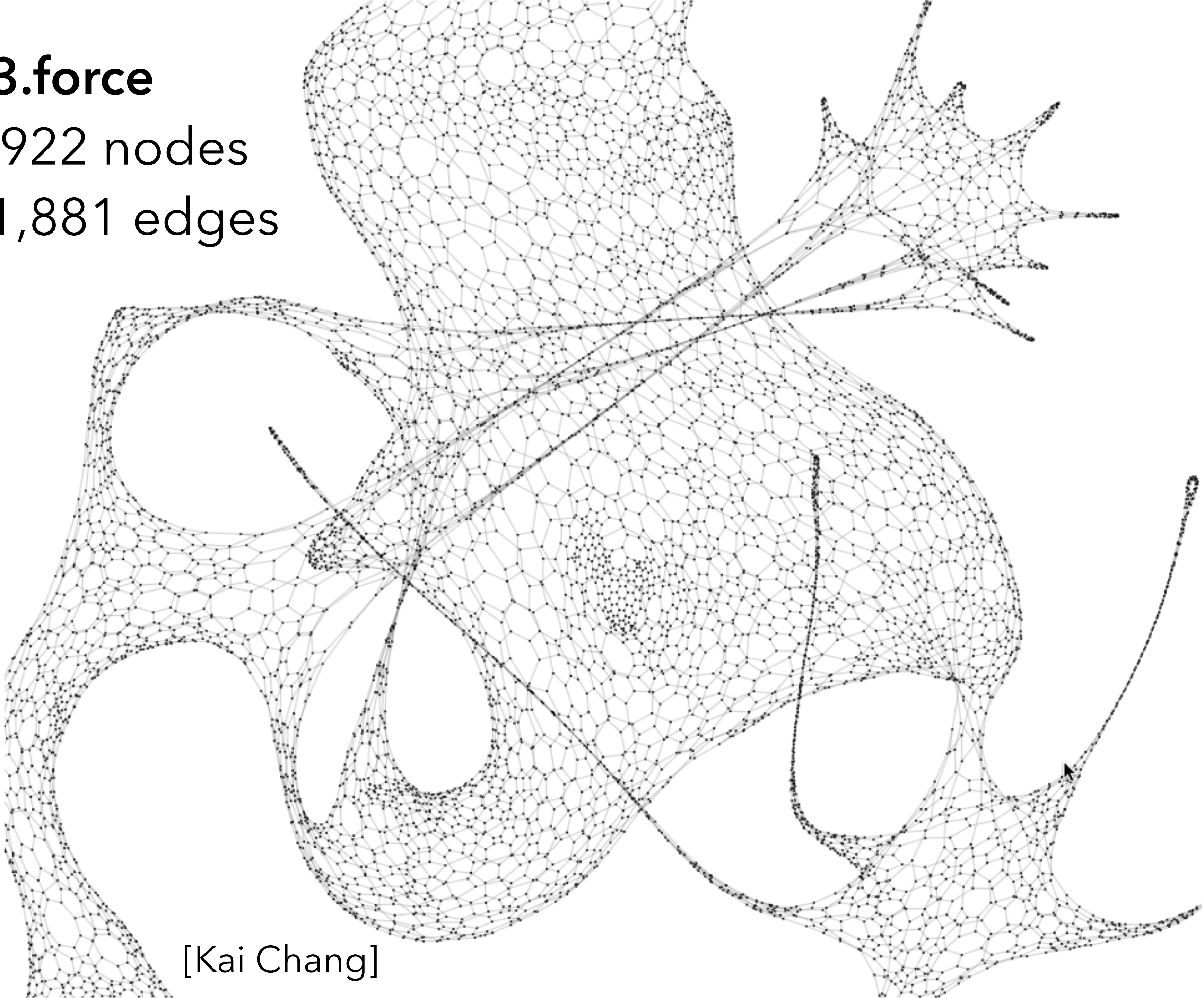
Use the Force!

http://mbostock.github.io/d3/talk/20110921/

d3.force

7,922 nodes

11,881 edges



[Kai Chang]

Force-Directed Layout

Nodes = charged particles $F = q_i * q_j / d_{ij}^2$

with air resistance $F = -b * v_i$

Edges = springs $F = k * (L - d_{ij})$

At each timestep, calculate forces acting on nodes.
Integrate for updated velocities and positions.

D3's force layout uses **velocity Verlet** integration.
Assume uniform mass ***m*** and timestep ***Δt***:
$F = ma \rightarrow F = a \rightarrow F = \Delta v / \Delta t \rightarrow F = \Delta v$
Forces simplify to velocity offsets!

N-Body Force

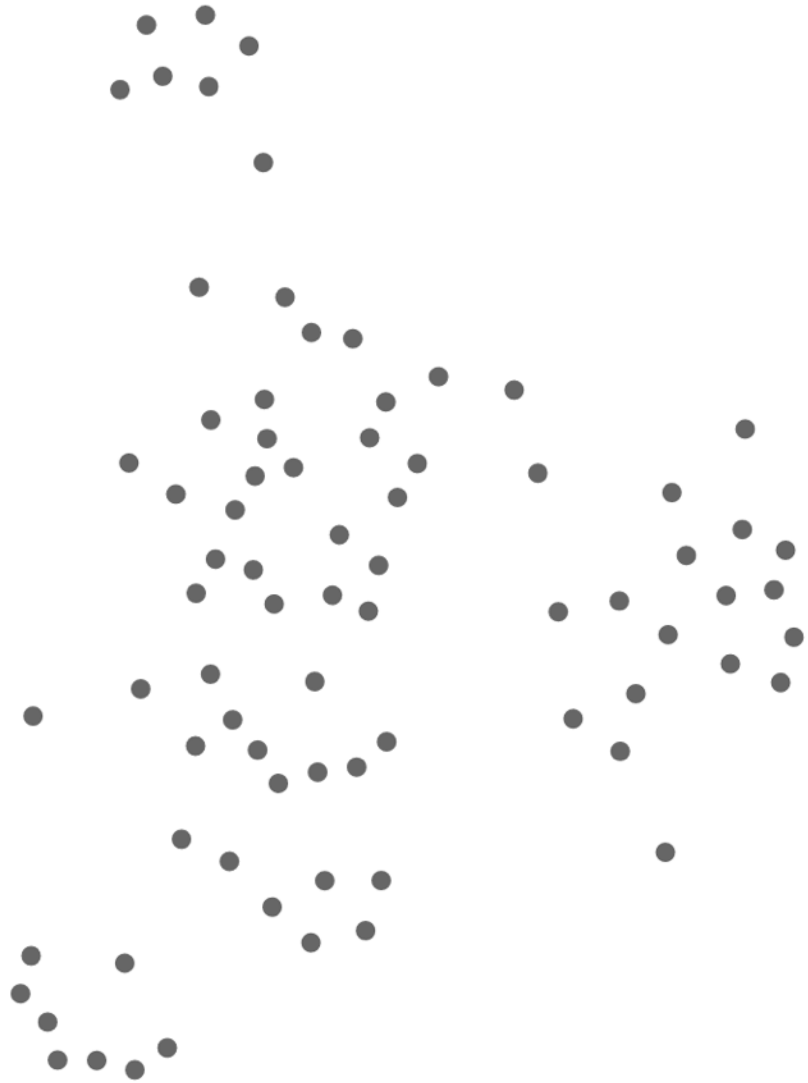
Naïve calculation of repulsive force doesn’t scale!

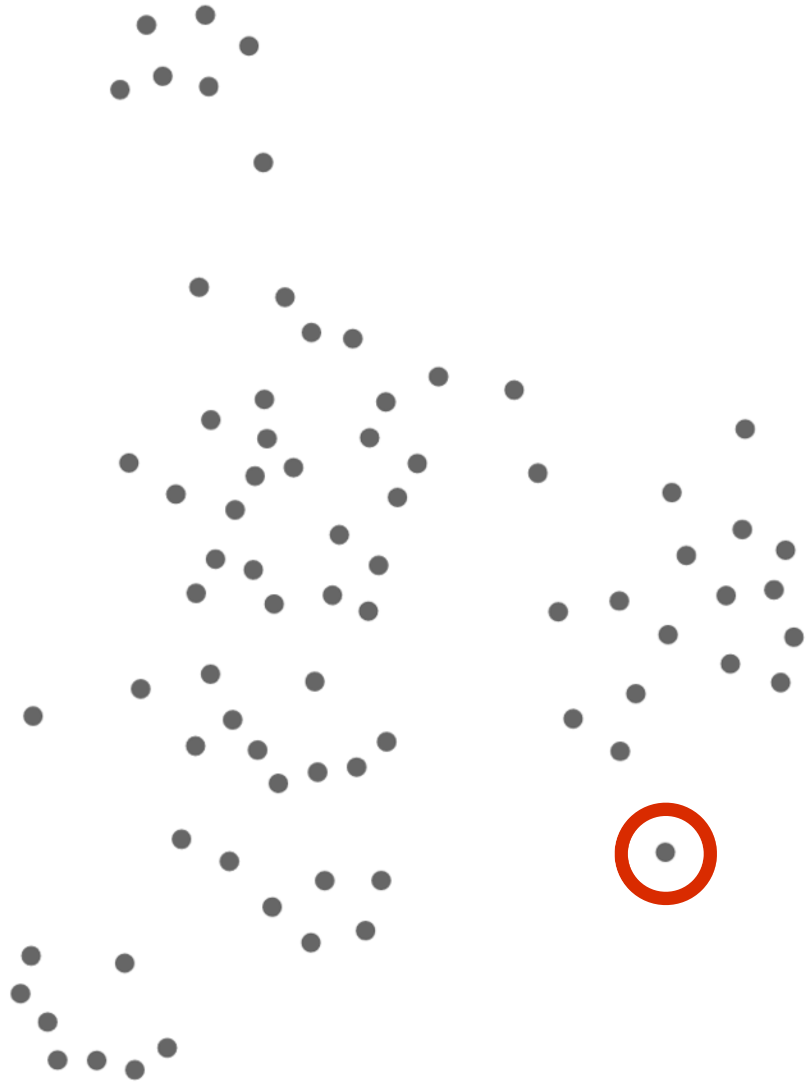
Comparing all pairs of nodes is $O(V^2)$

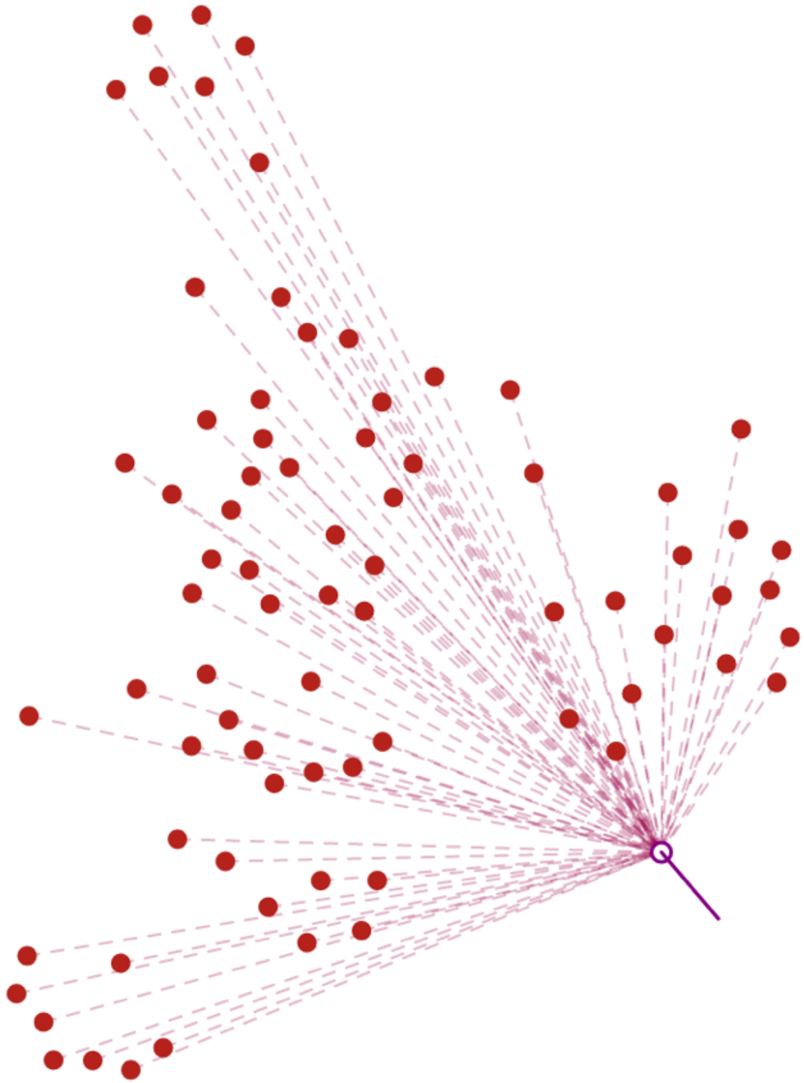
We can approximate force calculations using a spatial index (e.g., quadtree) to achieve O(V log V)

One such approach is the **Barnes-Hut algorithm**, originally created for astronomical simulations.

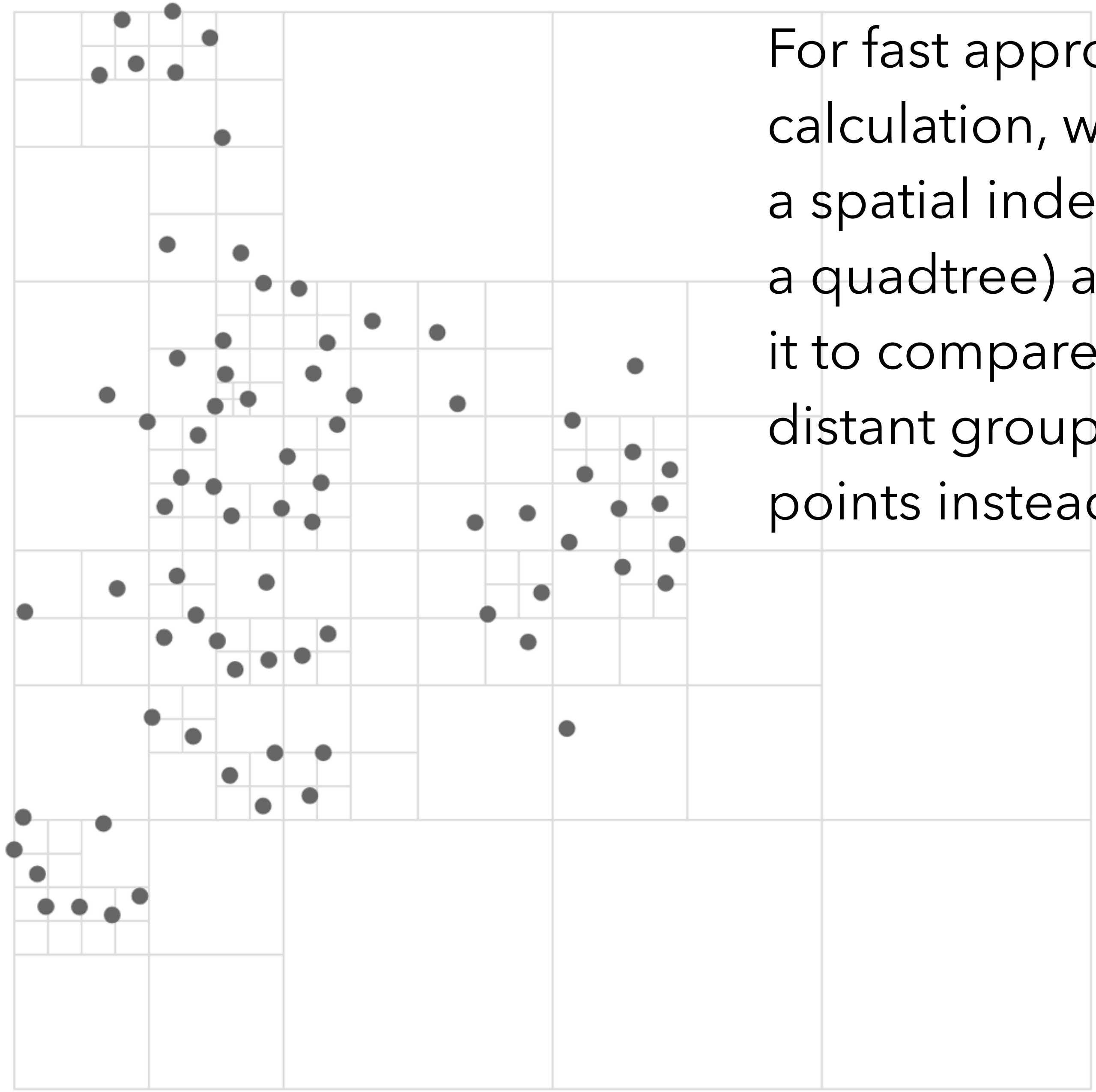
The key idea is to approximate forces from distant nodes by comparing to aggregate centers of charge rather than individual nodes.



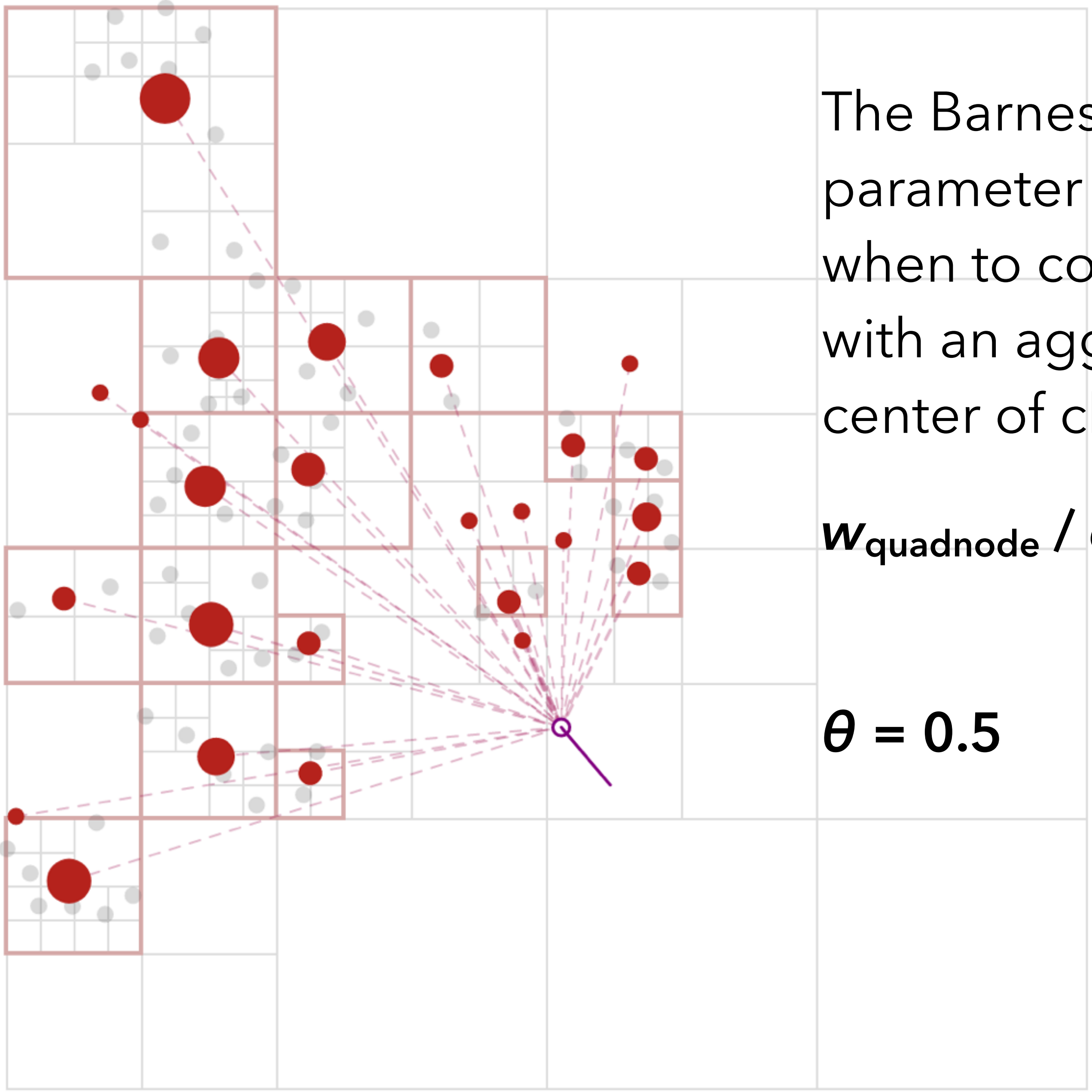




Naive calculation of forces at a point uses sum of forces from all other n-1 points.



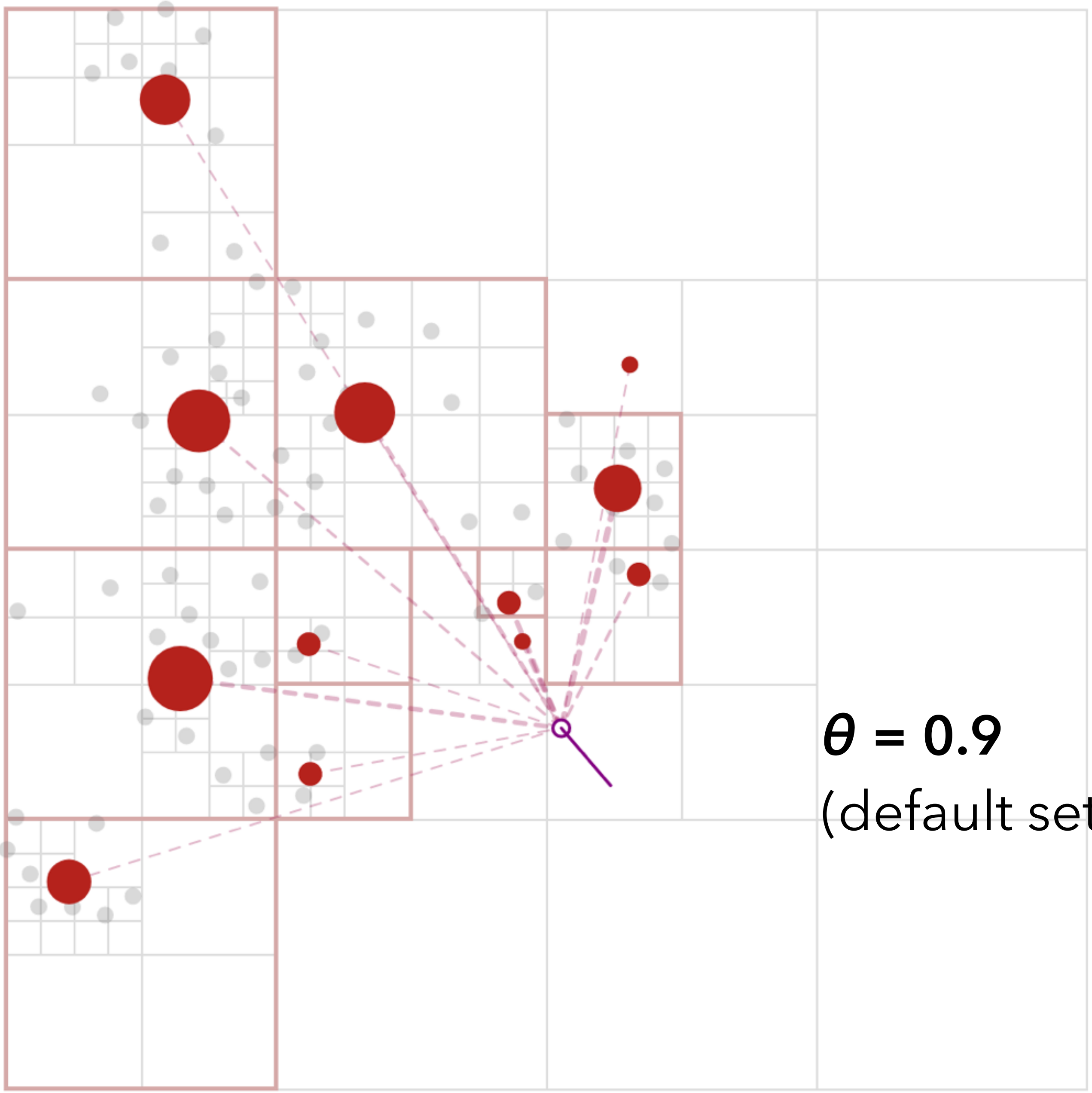
For fast approximate calculation, we build a spatial index (here, a quadtree) and use it to compare with distant groups of points instead.



The Barnes-Hut θ parameter controls when to compare with an aggregate center of charge.

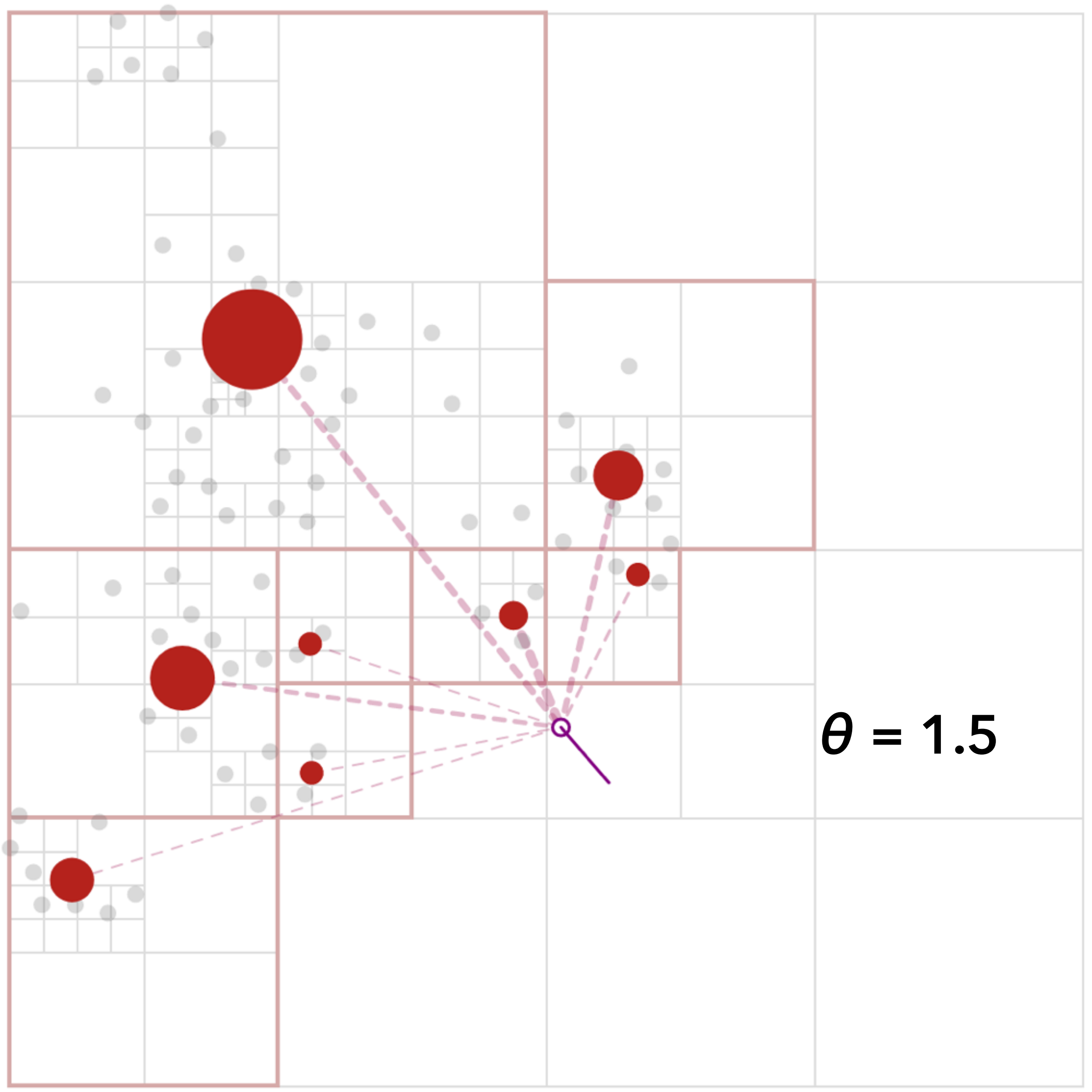
$w_{quadnode} / d_{ij} < \theta$?

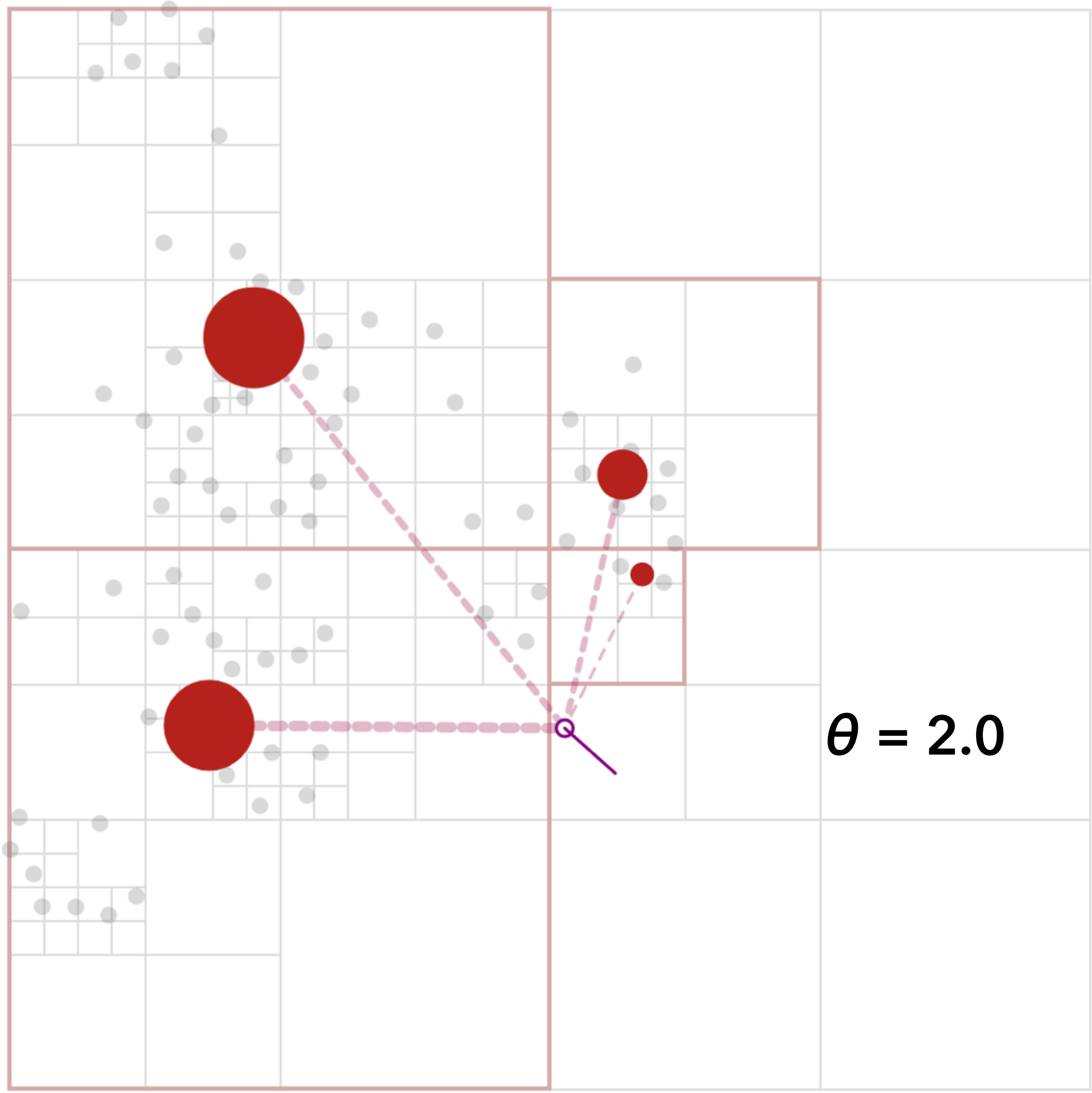
$\theta = 0.5$



θ = 0.9

(default setting)


θ = 1.5


$\theta = 2.0$

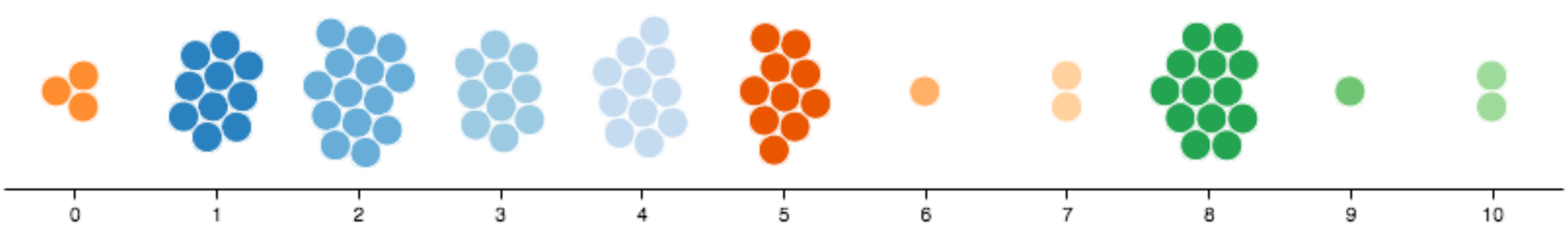
Customized Force Layouts

Different forces can be composed to create an expressive space of custom layouts.

A **beeswarm plot** can be made by combining:

Attractive **X** and **Y** forces to draw nodes of a certain category to a desired point

Collide force to detect collision & remove overlap



Constraint-Based Layout

Optimization Techniques

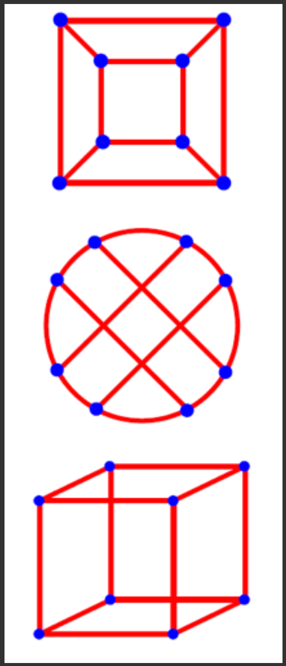
Treat layout as an *optimization problem*
Define layout using an *energy model* along with *constraints*: equations the layout should obey.
Use optimization algorithms to solve

Commonly posed as a physical system
Charged particles, springs, drag force, …

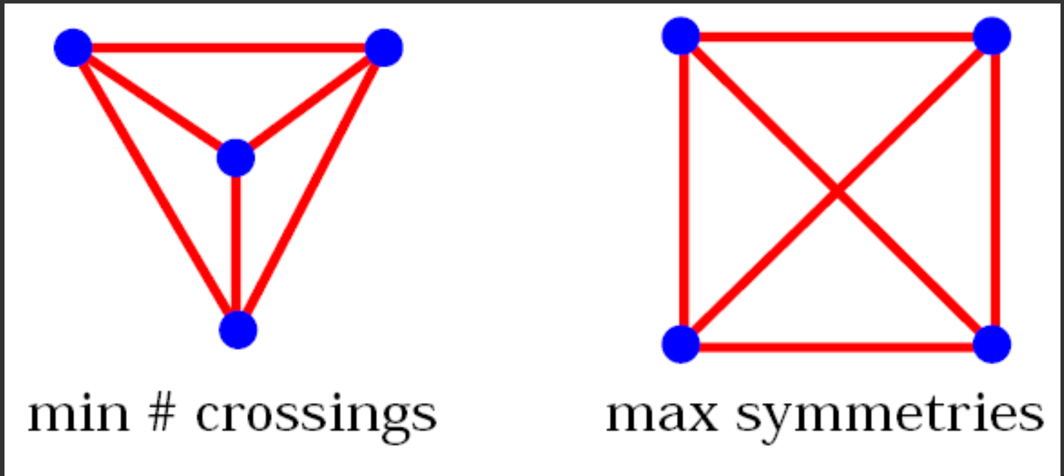
We can introduce directional constraints
DiG-CoLa (Di-Graph Constr Optimization Layout) [Dwyer 05]
Iterative constraint relaxation

Optimizing Aesthetic Constraints

Minimize edge crossings
Minimize area
Minimize line bends
Minimize line slopes
Maximize smallest angle between edges
Maximize symmetry



but, can't do it all.



Optimizing these criteria is often NP-Hard, requiring approximations.

Add Constraints to Force Layouts

Instead of springs, edges can be thought of as **distance constraints**. Constraints can be used for other purposes, such as overlap removal.

Iteratively **relax** each constraint [Dwyer 09]

Given a constraint (e.g., $| x_i - x_j | = 5$)

Simply push nodes apart to satisfy!

Each relaxation may **clobber** prior results, however this typically **converges quickly**.

Enables **expressive constraints**.

Formulate Layout as Optimization

Minimize stress function

$\text{stress}(X) = \Sigma_{i<j} w_{ij} (\|X_i - X_j\| - d_{ij})^2$

X: node positions, d: optimal edge length,

w: normalization constants

Says: Try to place nodes d_{ij} apart

Formulate Layout as Optimization

Minimize stress function

$\text{stress}(X) = \Sigma_{i<j}\, w_{ij}\, (\, \|X_i - X_j\| - d_{ij}\,)^2$

X: node positions, d: optimal edge length,

w: normalization constants

Says: Try to place nodes d_{ij} apart

Add hierarchy ordering constraints

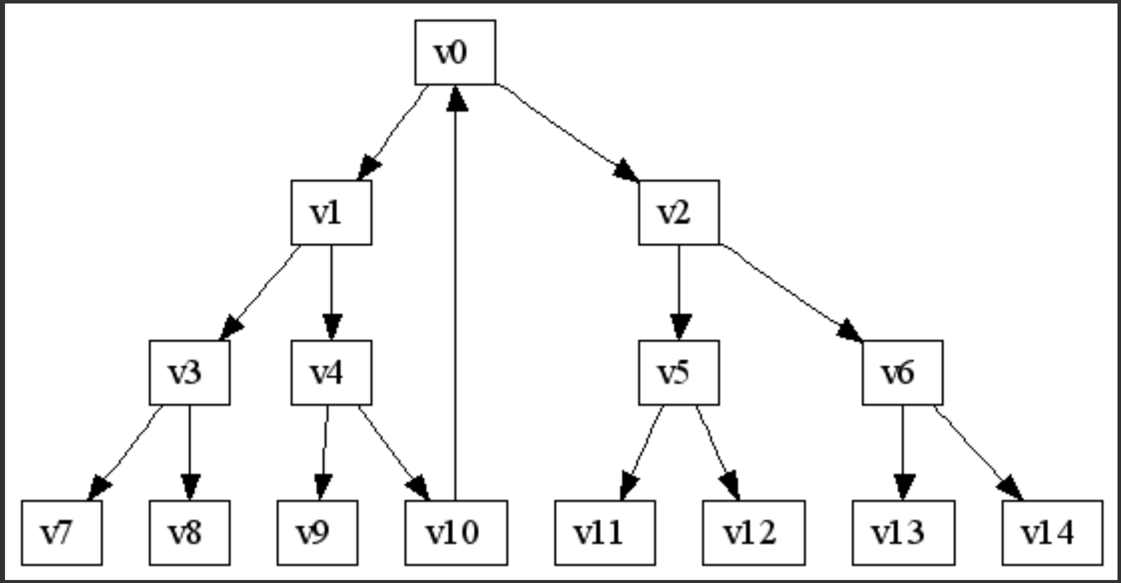
$E_H(y) = \Sigma_{(i,j)\in E}\, (\, y_i - y_j - \delta_{ij}\,)^2$

y: node y-coordinates

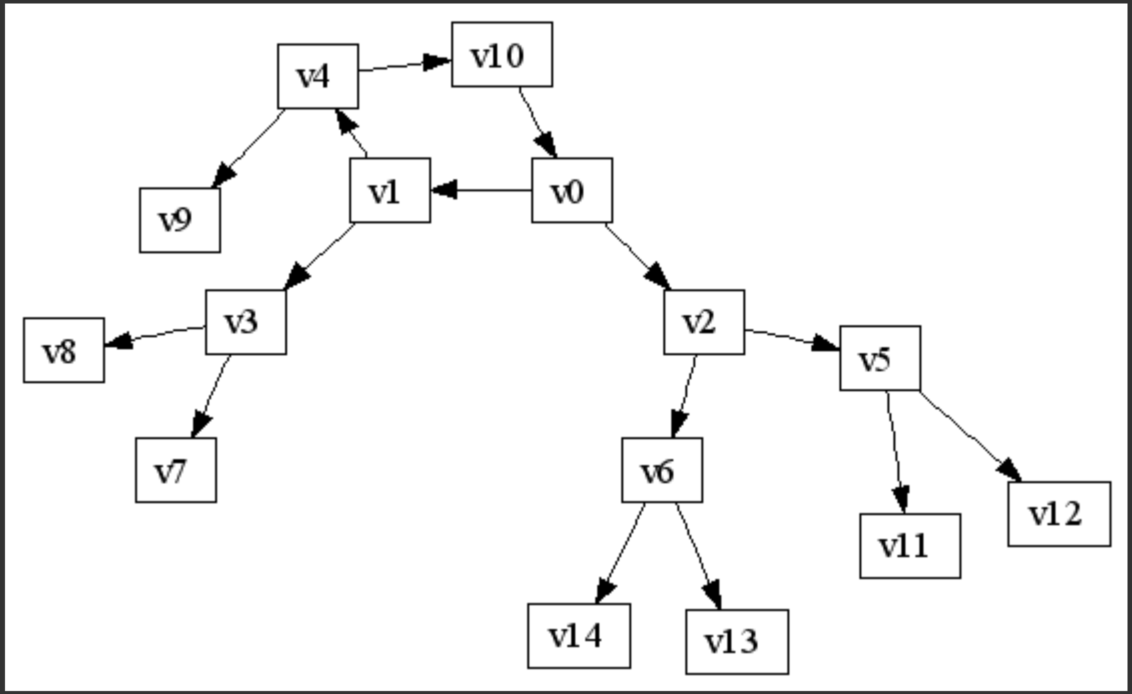
δ: edge direction (e.g., 1 for i->j, 0 for undirected)

Says: If i points to j, it should have a lower y-value

Sugiyama layout (dot)
Preserve tree structure

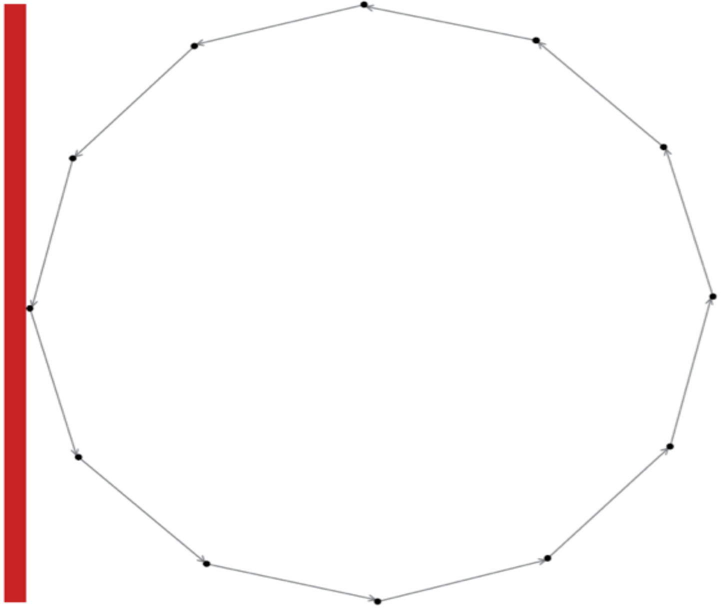


DiG-CoLa method
Preserve edge lengths

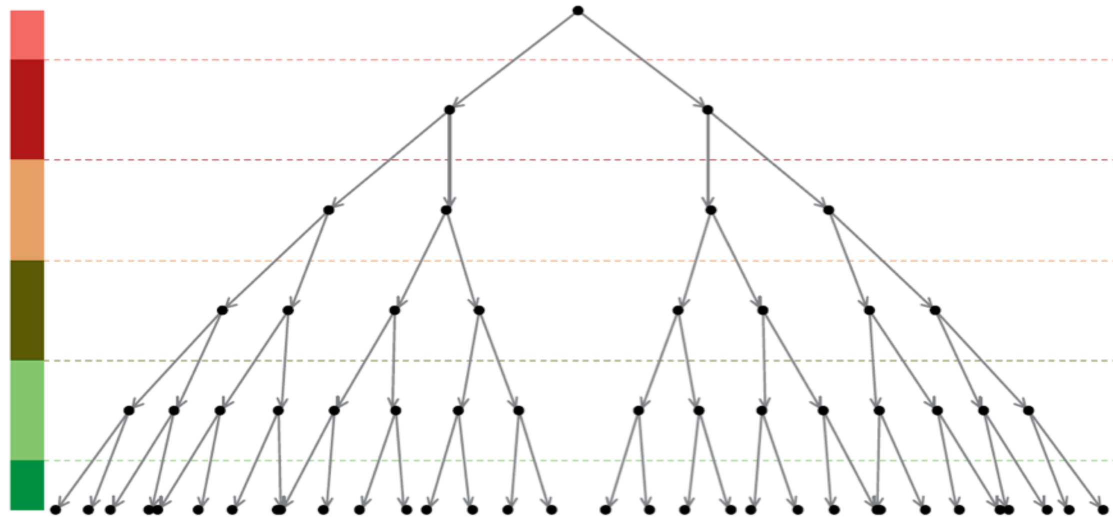


[Slide from Tim Dwyer]

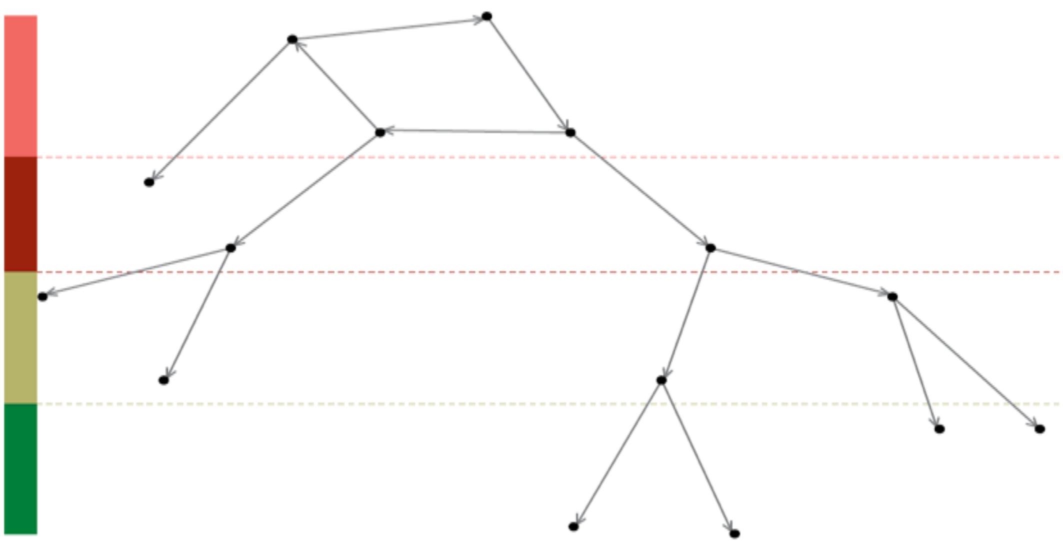
[Slide from Tim Dwyer]



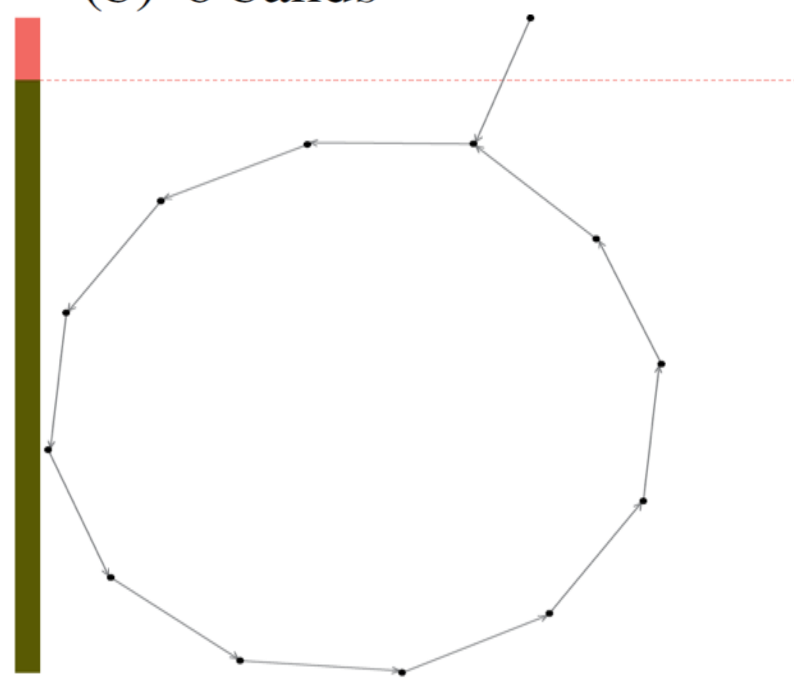
(a) 1 band



(b) 6 bands

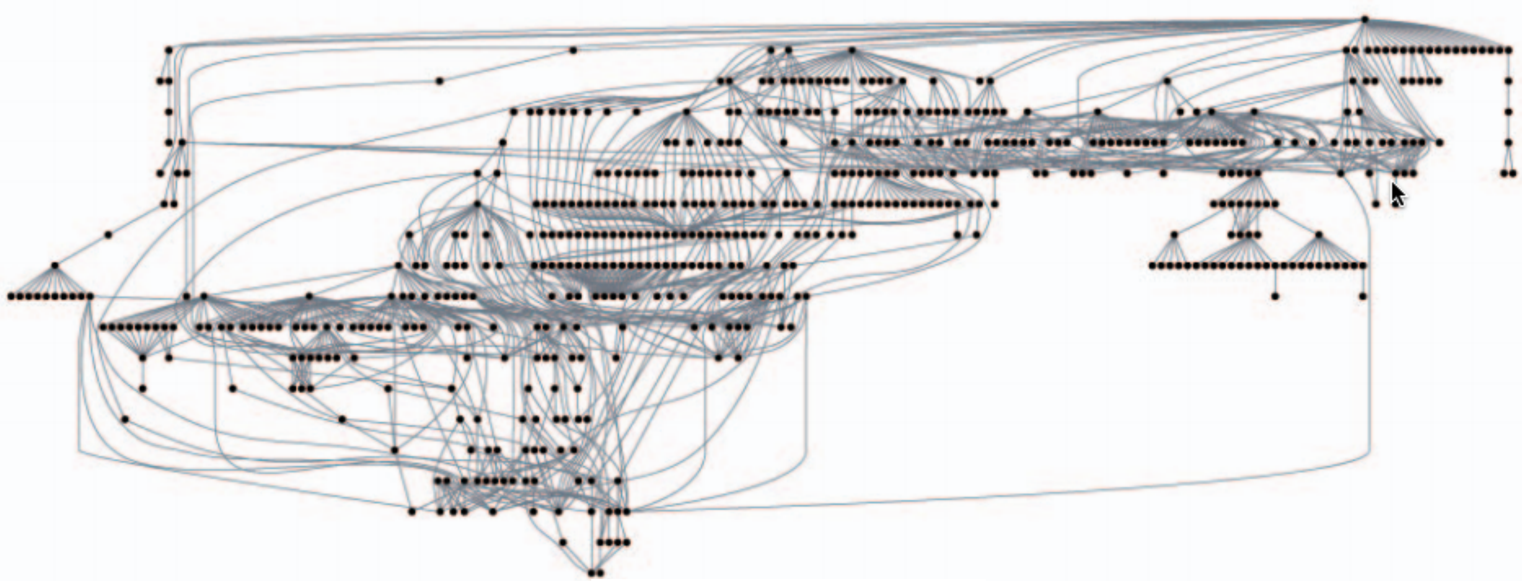


(c) 4 bands



(d) 2 bands

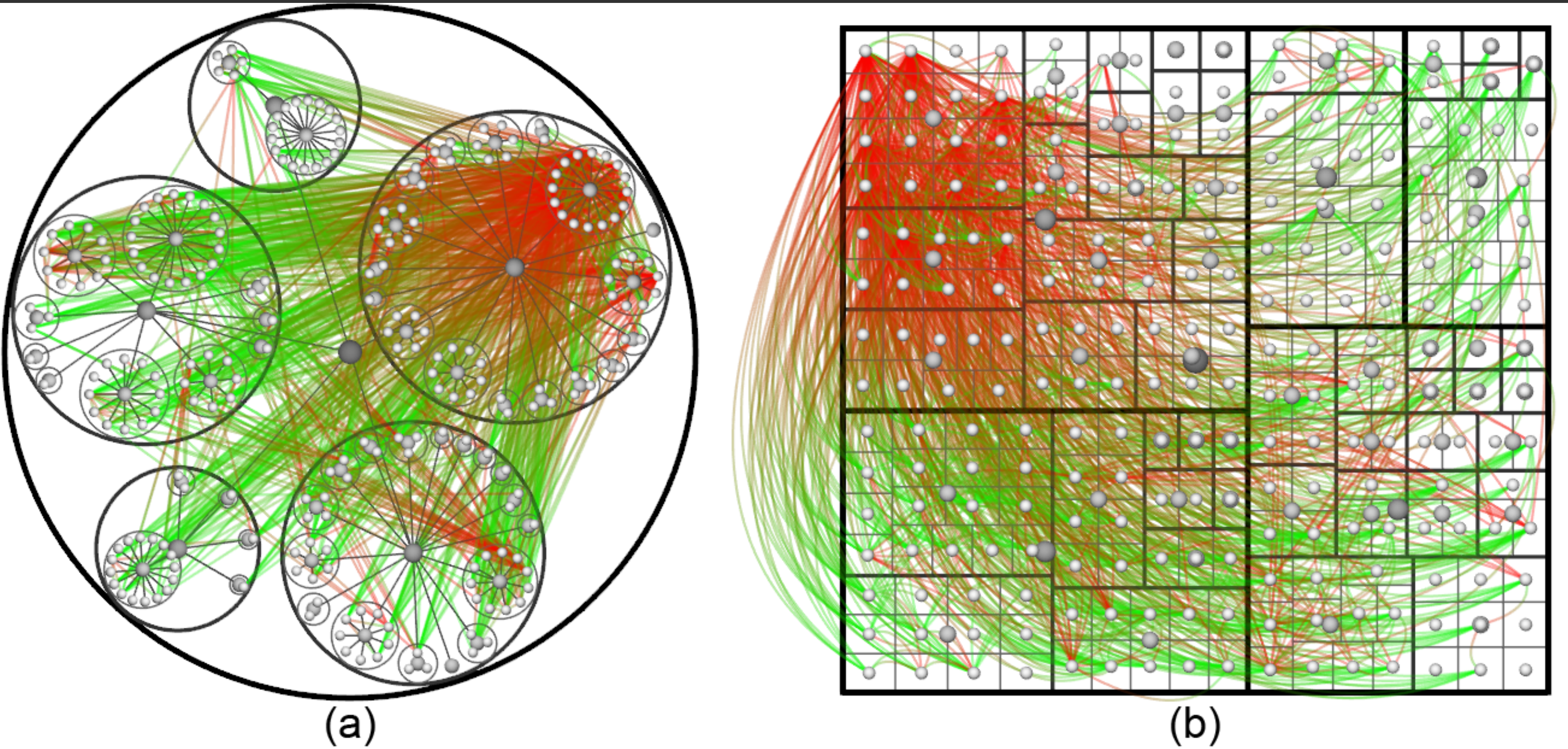
DiG-CoLa Layout



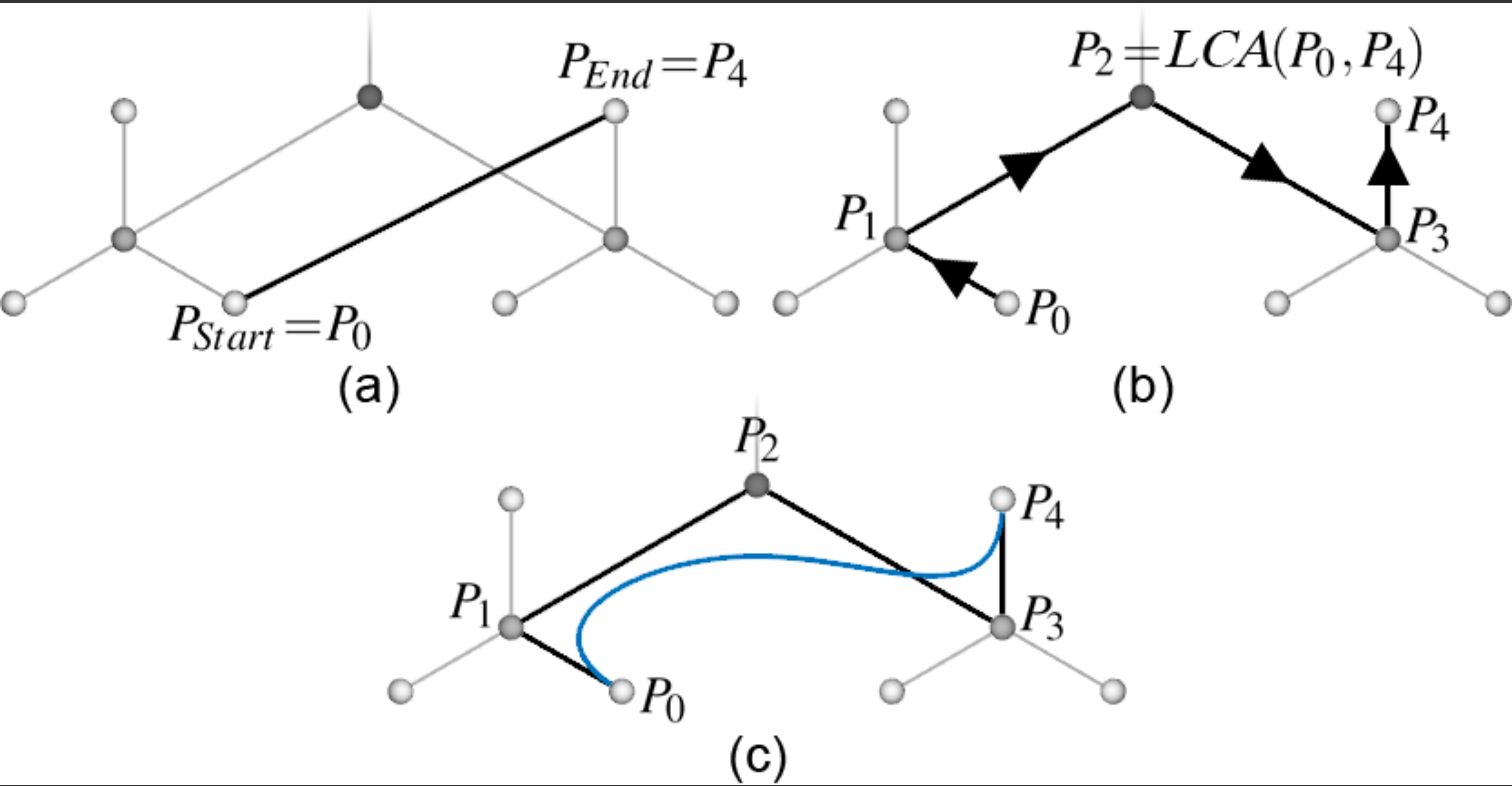
Sugiyama Layout

Hierarchical Edge Bundling

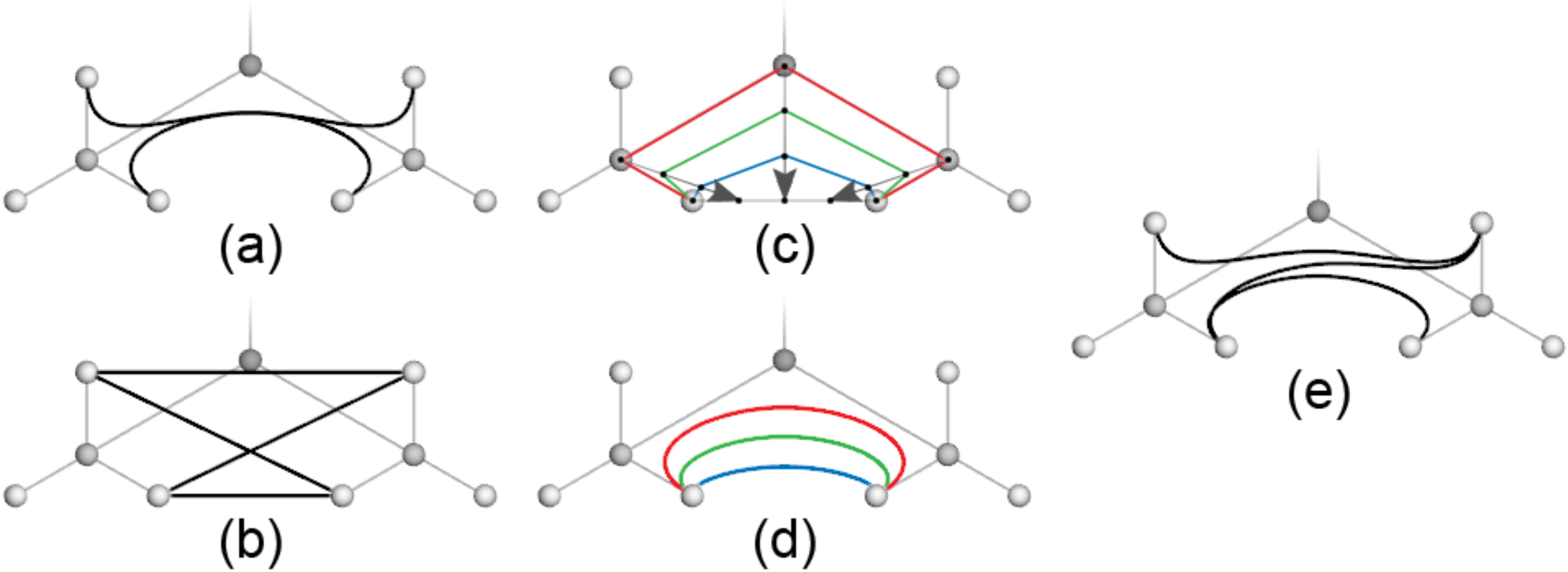
Trees plus Adjacency Relations

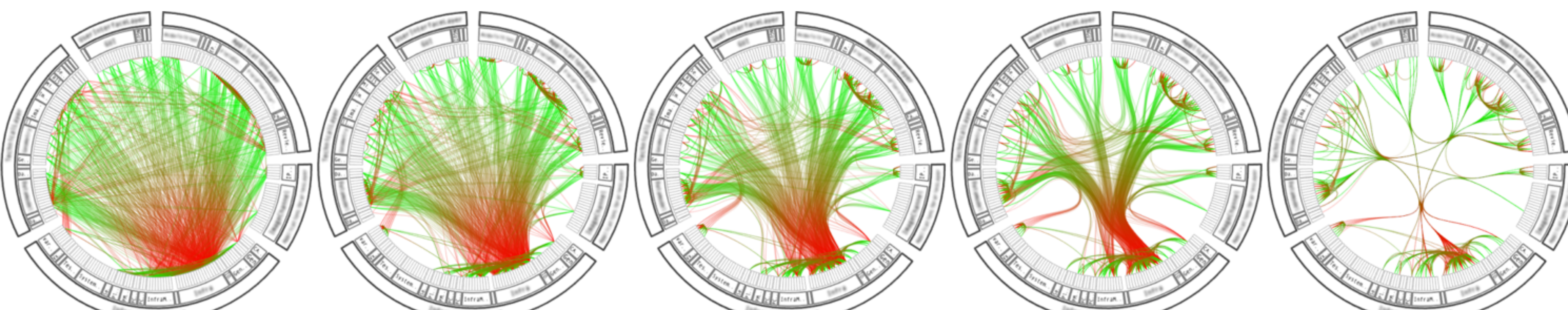


Bundle Edges Along Tree Paths



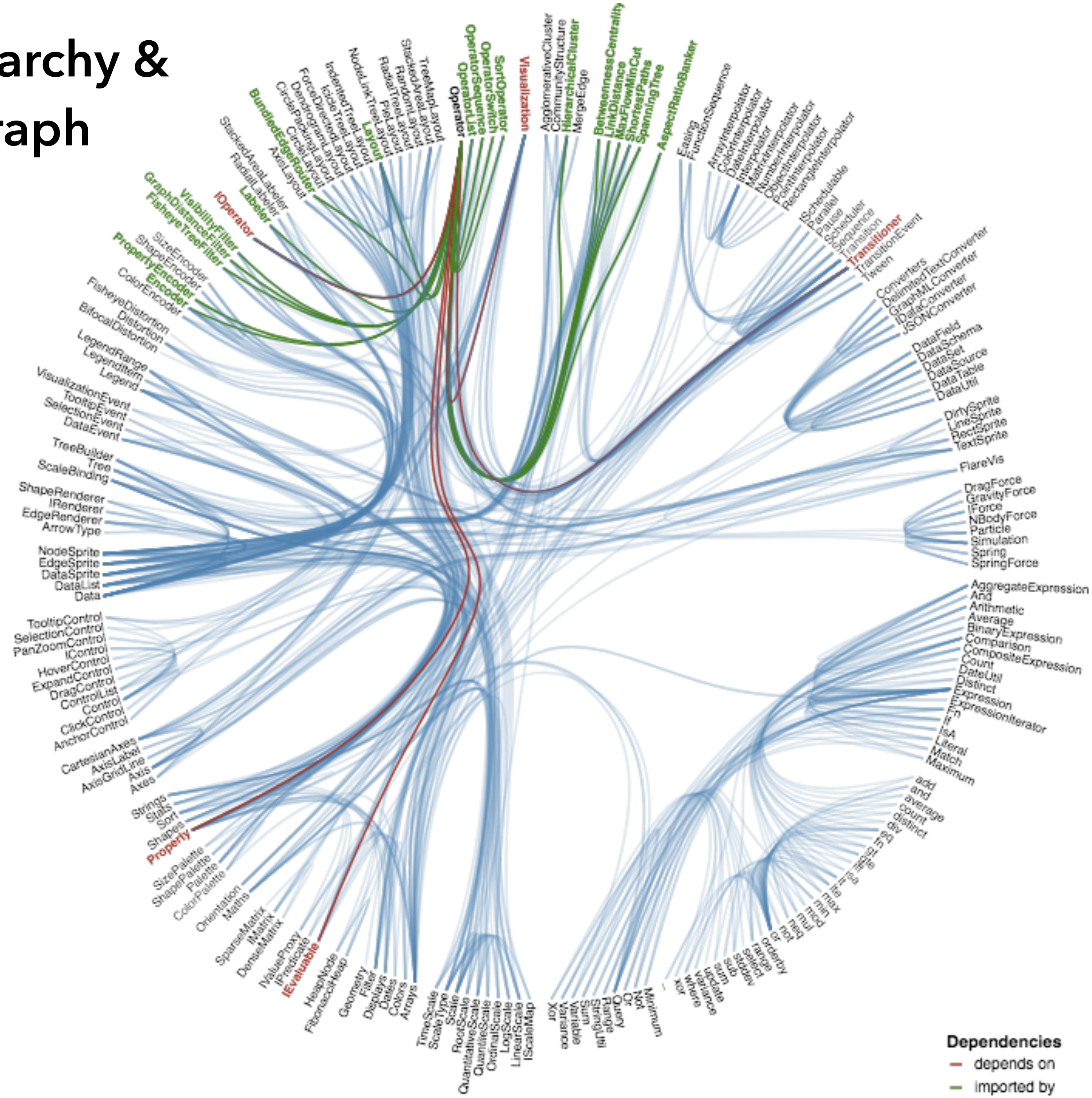
Configuring Edge Tension





$\beta = 0$ $\beta = 0.25$ $\beta = 0.5$ $\beta = 0.75$ $\beta = 1$

Flare Class Hierarchy & Dependency Graph



Administrivia

Final Project Deliverables

Interactive Web Page
Working (near-final) version due Wed 5/31.
Final version due by showcase on Mon 6/5.

Demonstration Video (<= 2 min)
Due Wed 5/31. We will show in-class on 6/1!

Poster & Demo for Final Showcase
Monday 6/5, 10:30am-1pm in Allen Center atrium.
External judges will award top projects!

<u>Read assignment description for more!</u>

Final Project Showcase

When: Monday June 5, 10:30am - 1pm.
Where: Allen Center Atrium

The event is open to the public. Invite your friends!

Public showing begins at 11am. **Arrive at 10:30am** to set up your poster and demo. Be prepared to give a **~3 min. presentation + demo** to visitors.

Invited judges will rate & award the top projects.

Refreshments will be served!

Tips for a Successful Project

Focus on a compelling **real-world use**.
Who is your user? How do you gauge success?

Consider **multiple design alternatives**.
Prototype quickly (use Tableau, R, Gephi…).

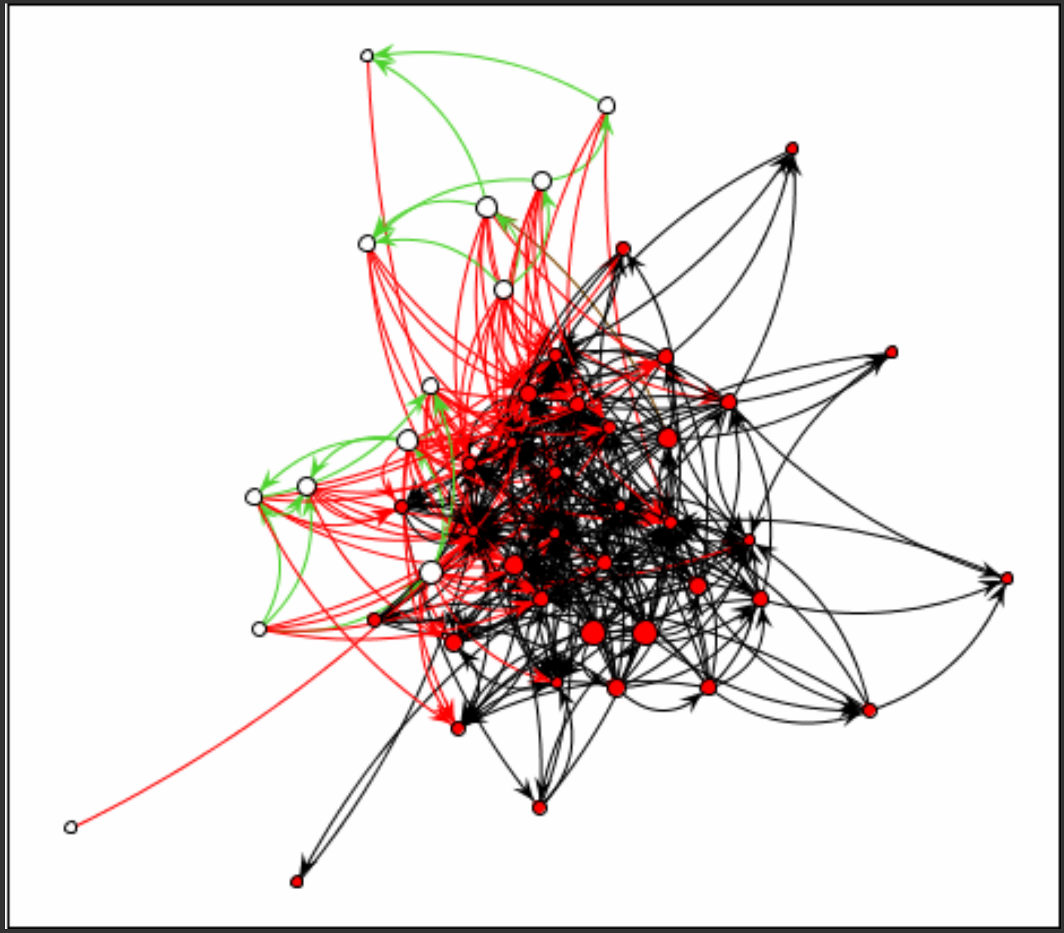
Seek feedback (representative users, peers, …).
Even informal usage can provide insights.

Choose **appropriate team roles**.

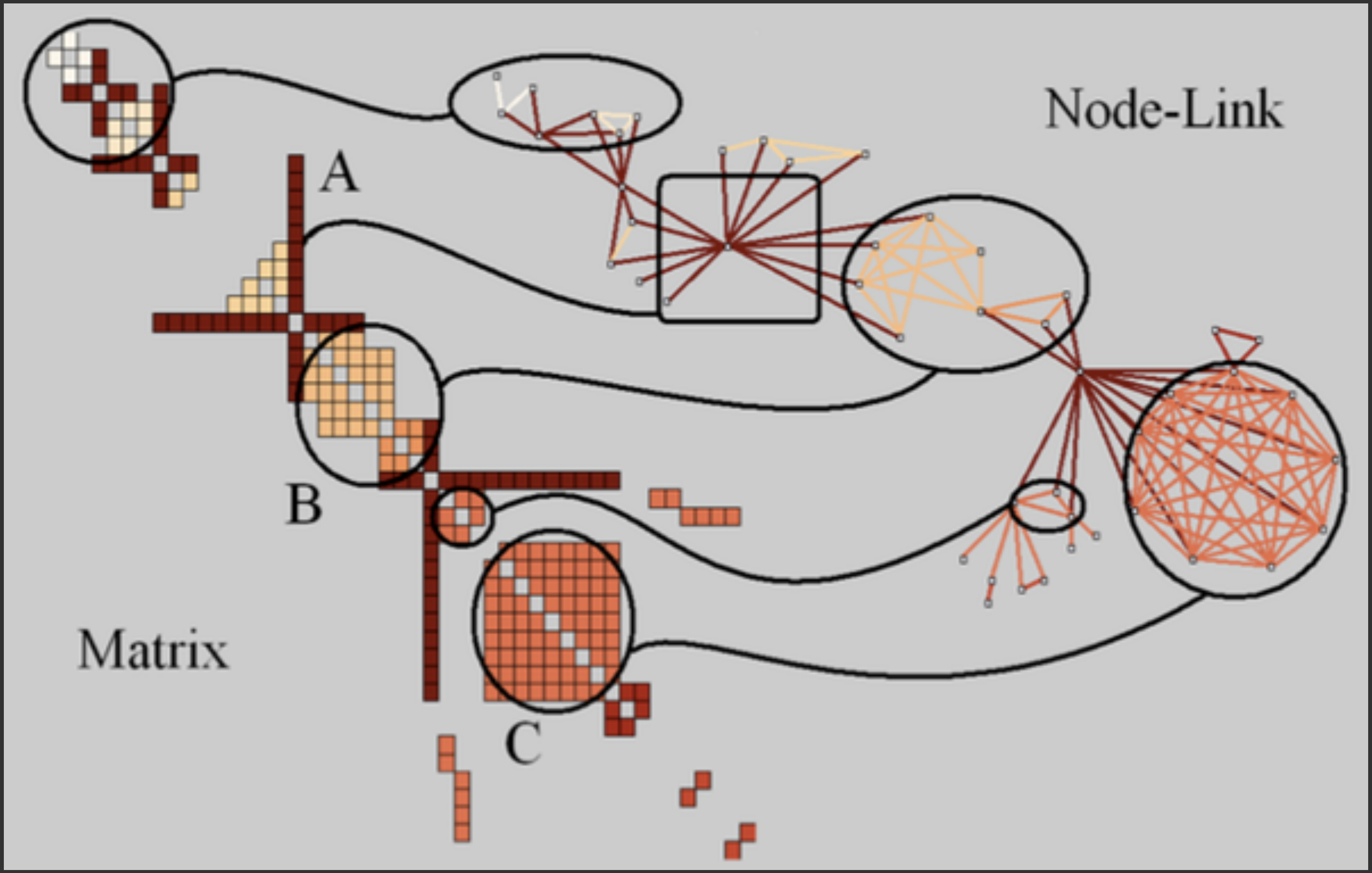
Start early (and read the suggested paper!)

Matrix Diagrams

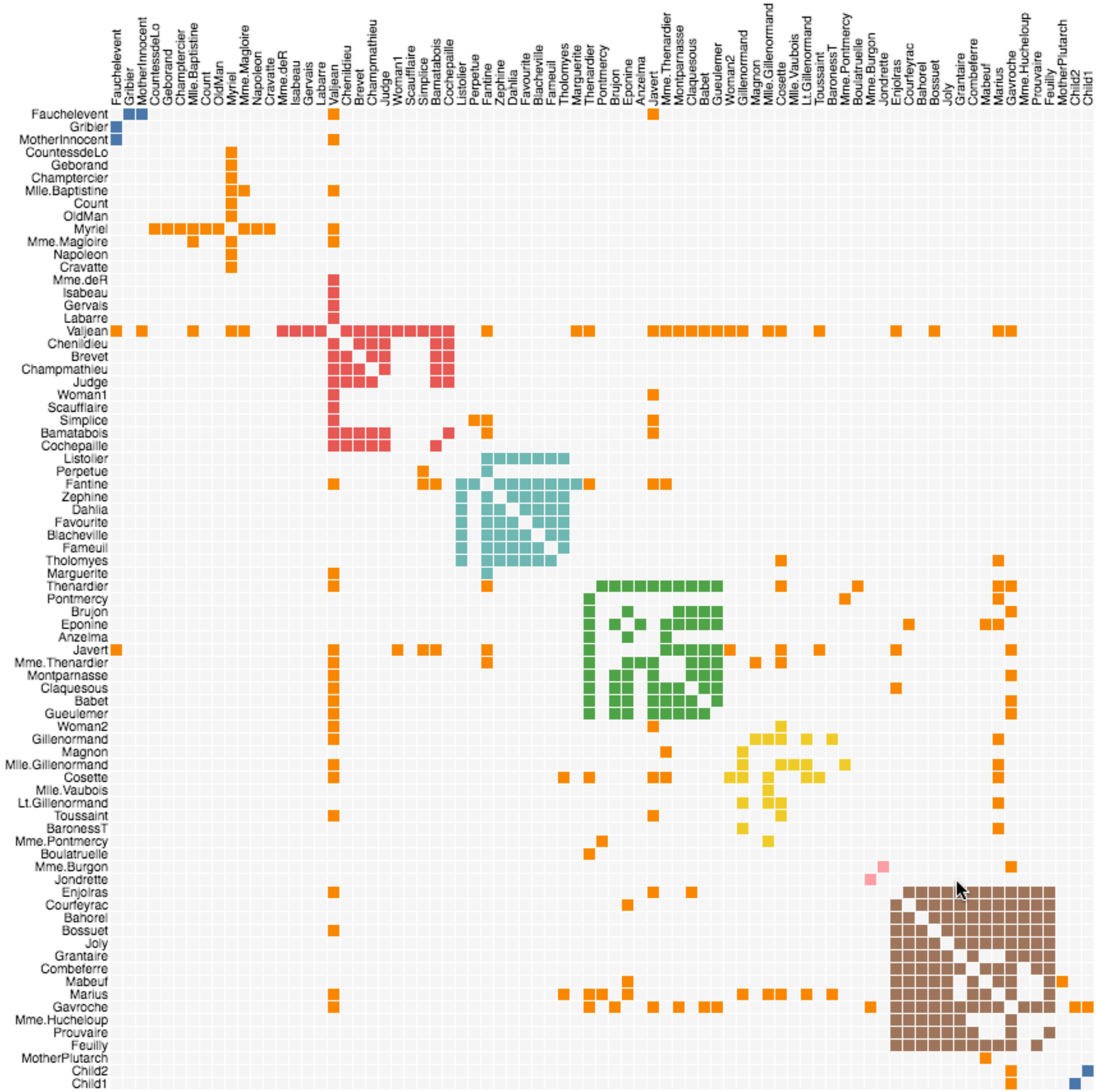
Limitations of Node-Link Layouts

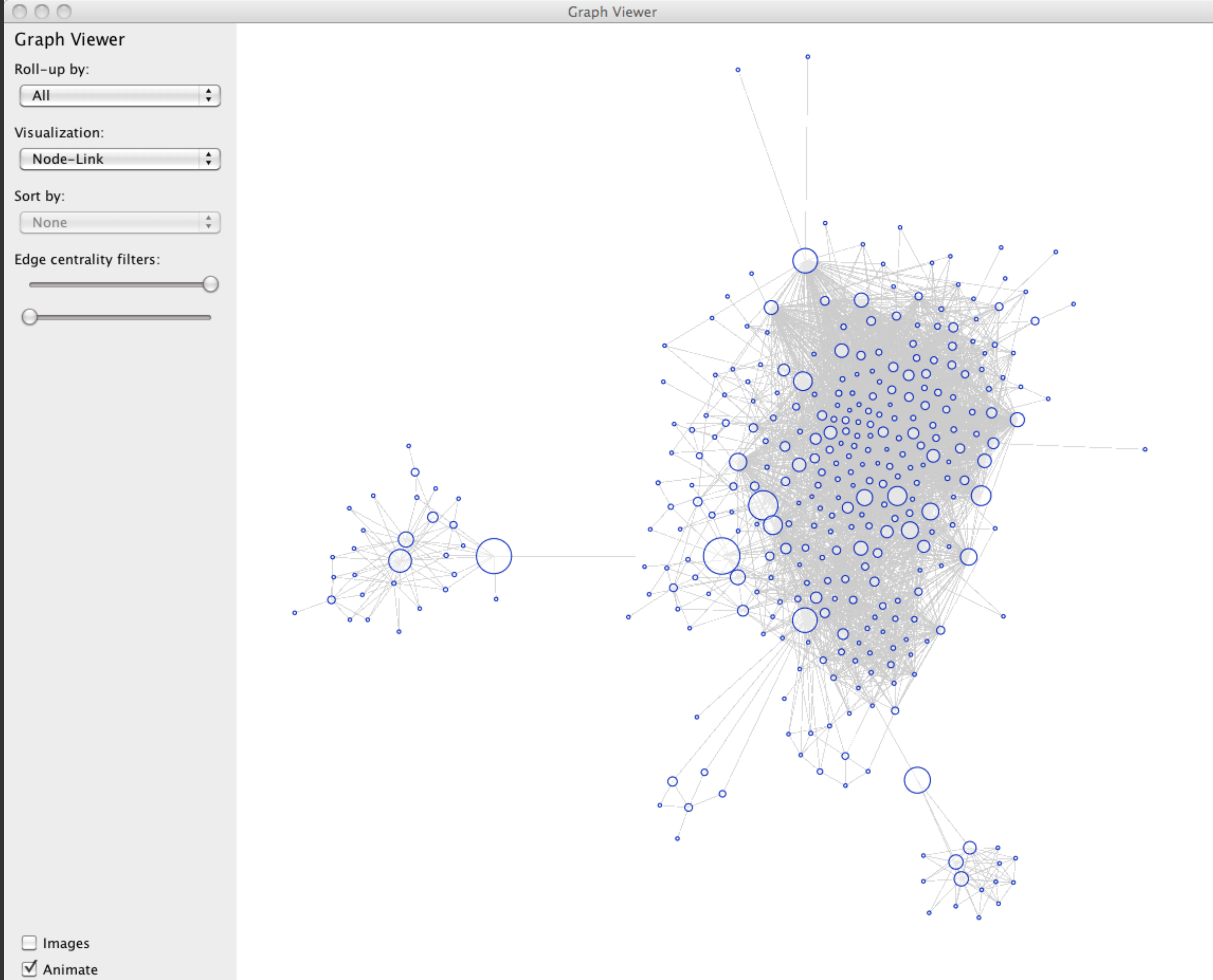


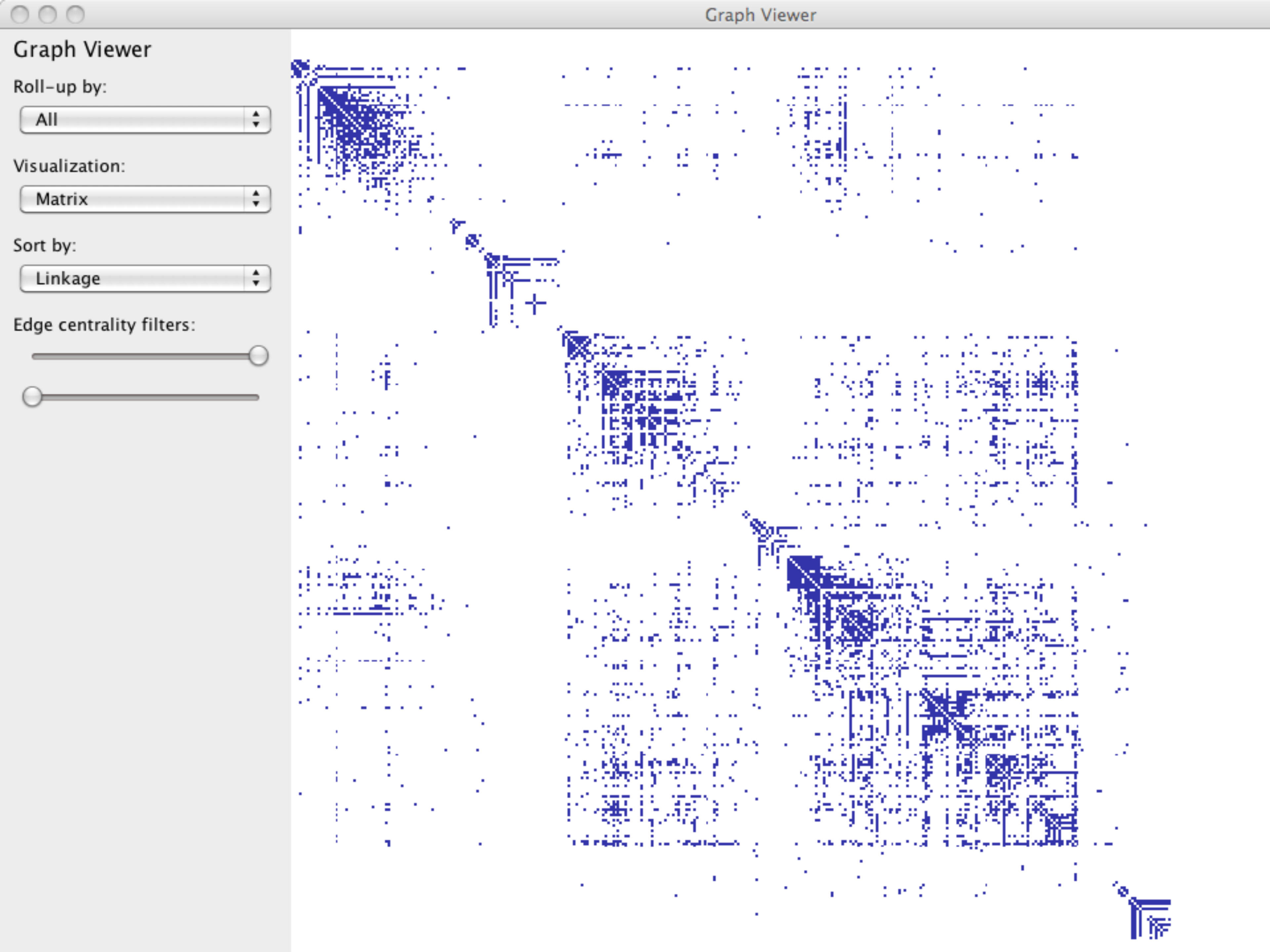
Edge-crossings and occlusion! Poor scalability….

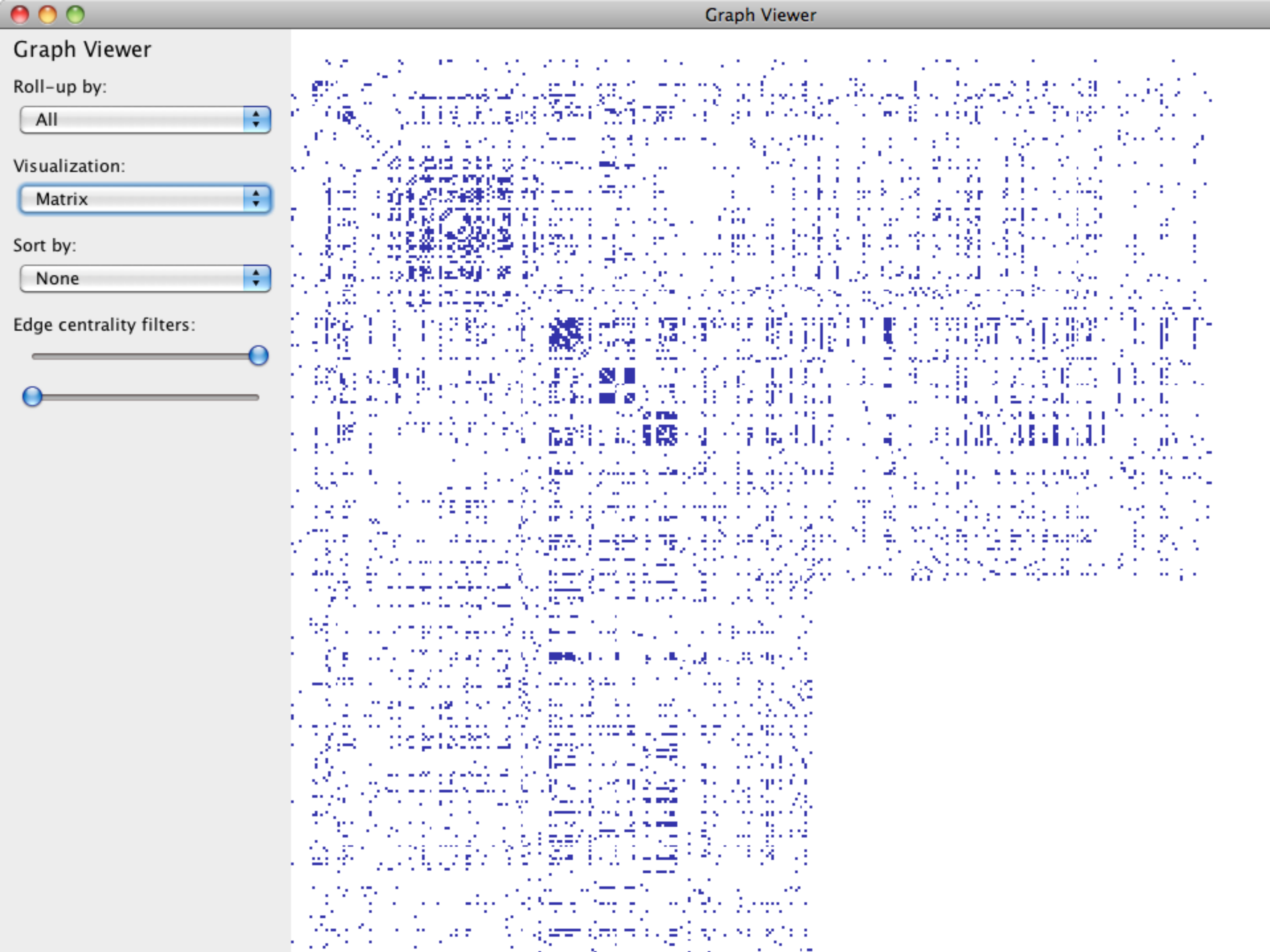


Adjacency Matrices


Fauchelevent
Gribier
MotherInnocent
CountessdeLo
Geborand
Champtercier
Mlle.Baptistine
Count
OldMan
Myriel
Mme.Magloire
Napoleon
Cravatte
Mme.deR
Isabeau
Gervais
Labarre
Valjean
Chenildieu
Brevet
Champmathieu
Judge
Woman1
Scaufflaire
Simplice
Bamatabois
Cochepaille
Listolier
Perpetue
Fantine
Zephine
Dahlia
Favourite
Blacheville
Fameuil
Tholomyes
Marguerite
Thenardier
Pontmercy
Brujon
Eponine
Anzelma
Javert
Mme.Thenardier
Montparnasse
Claquesous
Babet
Gueulemer
Woman2
Gillenormand
Magnon
Mlle.Gillenormand
Cosette
Mlle.Vaubois
Lt.Gillenormand
Toussaint
BaronessT
Mme.Pontmercy
Boulatruelle
Mme.Burgon
Jondrette
Enjolras
Courfeyrac
Bahorel
Bossuet
Joly
Grantaire
Combeferre
Mabeuf
Marius
Gavroche
Mme.Hucheloup
Prouvaire
Feuilly
MotherPlutarch
Child2
Child1


Graph Viewer
Graph Viewer
Roll-up by:
All
Visualization:
Node-Link
Sort by:
None
Edge centrality filters:
Images
Animate


Graph Viewer
Graph Viewer
Roll-up by:
All
Visualization:
Matrix
Sort by:
Linkage
Edge centrality filters:


Graph Viewer
Graph Viewer
Roll-up by:
All
Visualization:
Matrix
Sort by:
None
Edge centrality filters:

Seriation / Ordination / Permutation

Goal: Ensure similar items placed near each other.
E.g., minimize sum of distances of adjacent items.

Requires combinatorial optimization: **NP-Hard!**

Instead, approximate / heuristic approaches used:
- Perform hierarchical clustering, sort cluster tree.
- Apply approximate traveling salesperson solver.

Seriation initially used in **archaeology** for relative dating of artifacts based on observed properties.

Attribute-Driven Layout

Attribute-Driven Layout

Large node-link diagrams **get messy**!
Is there additional structure we can exploit?

Idea: Use **data attributes** to perform layout
For example, scatter plot based on node values

Attributes may be associated with nodes or edges
or may be statistical properties of the graph.

Use dynamic queries / brushing to explore…

Attribute-Driven Layout

The “Skitter” Layout

Internet Connectivity

Radial Scatterplot

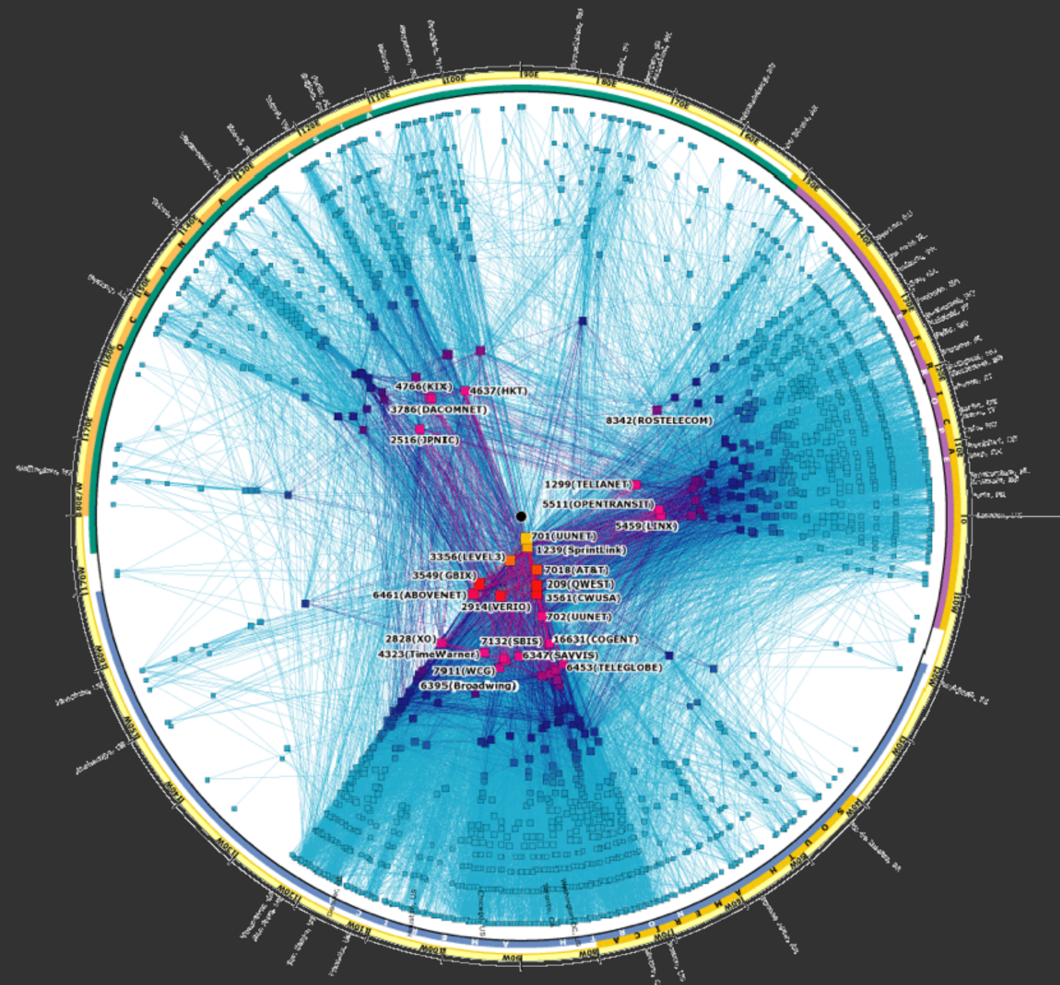
Angle = Longitude

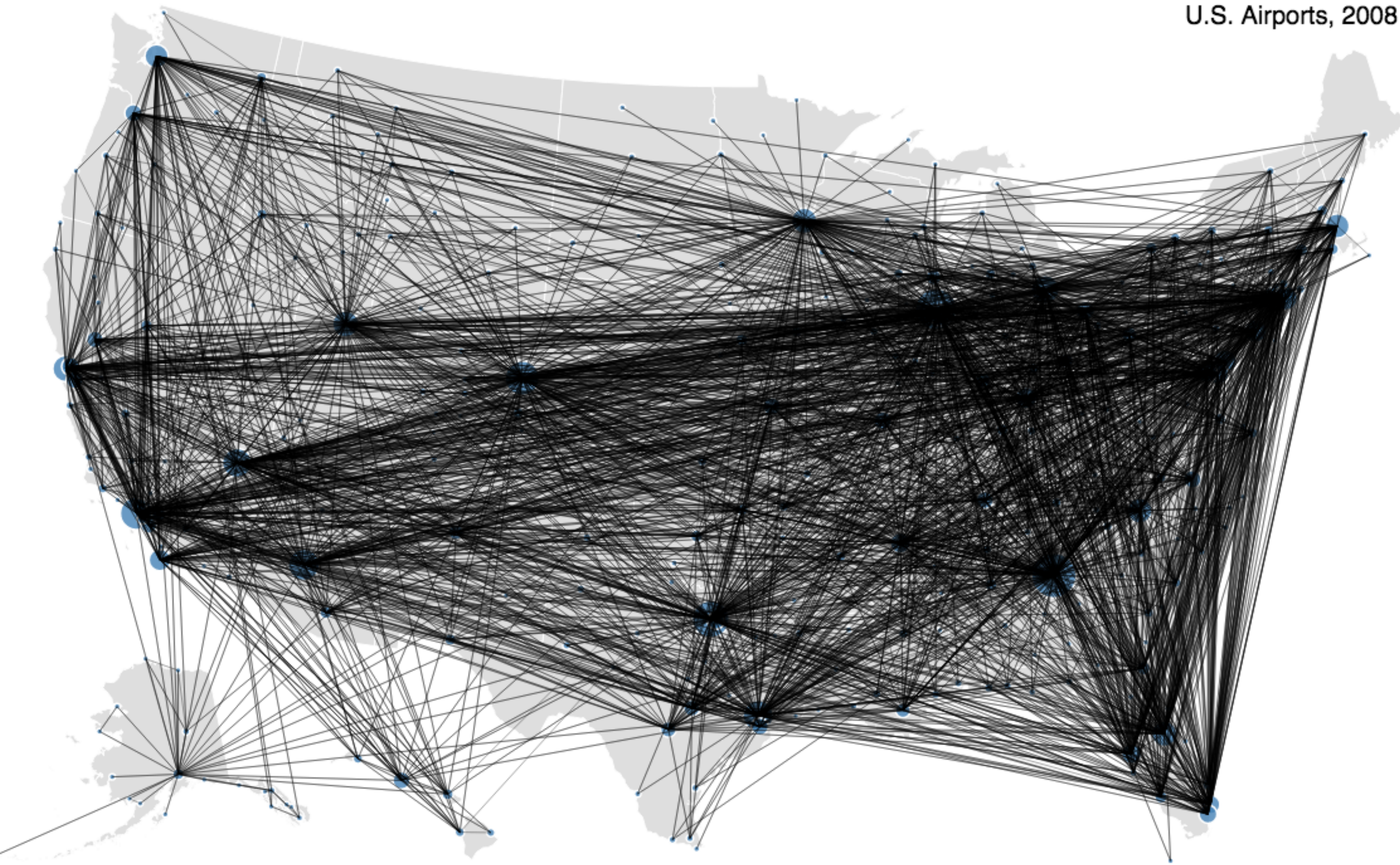
Geography

Radius = Degree

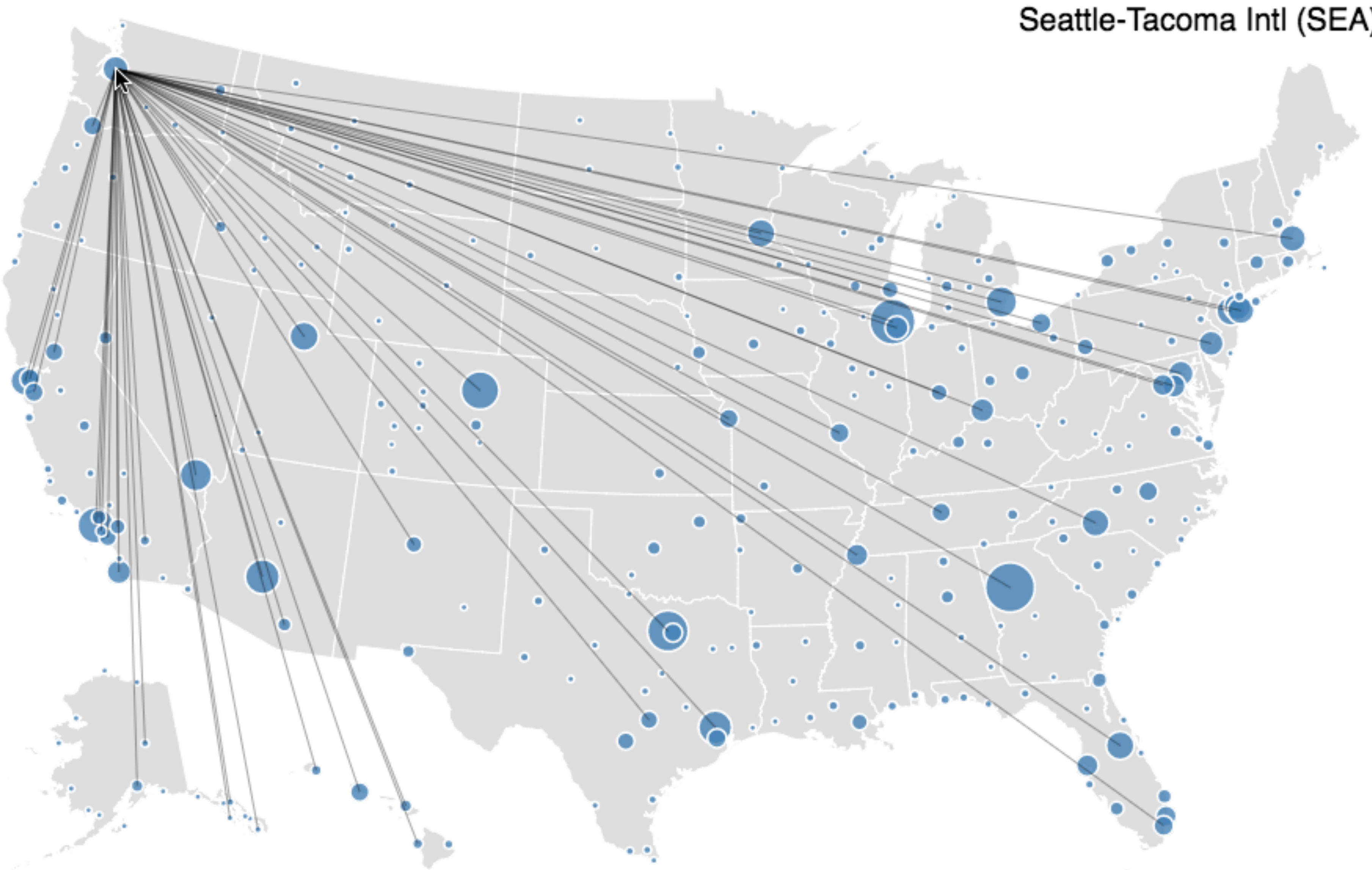
# of connections

(a statistic of the nodes)



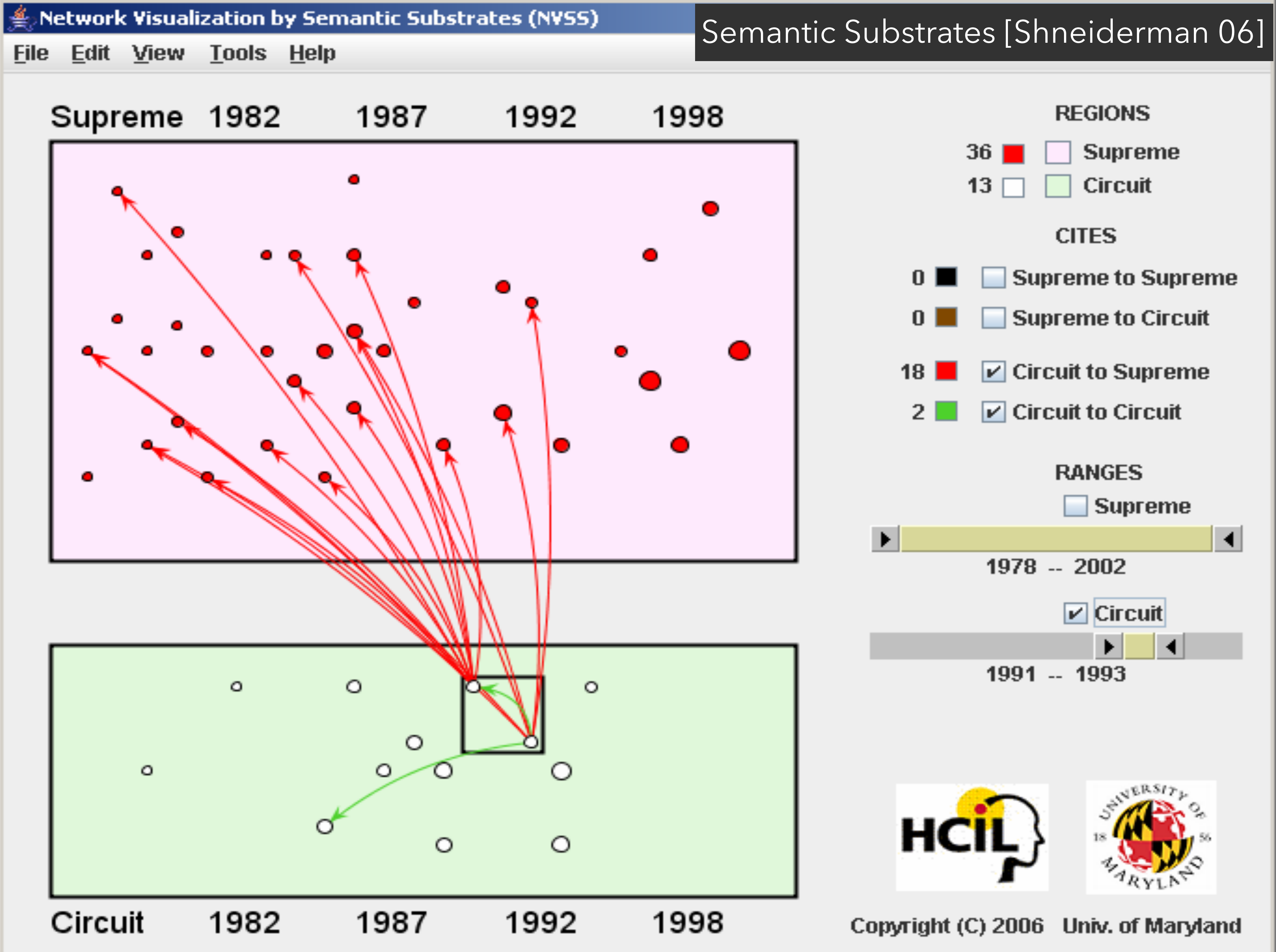


Drawing all edges is not particularly useful here…

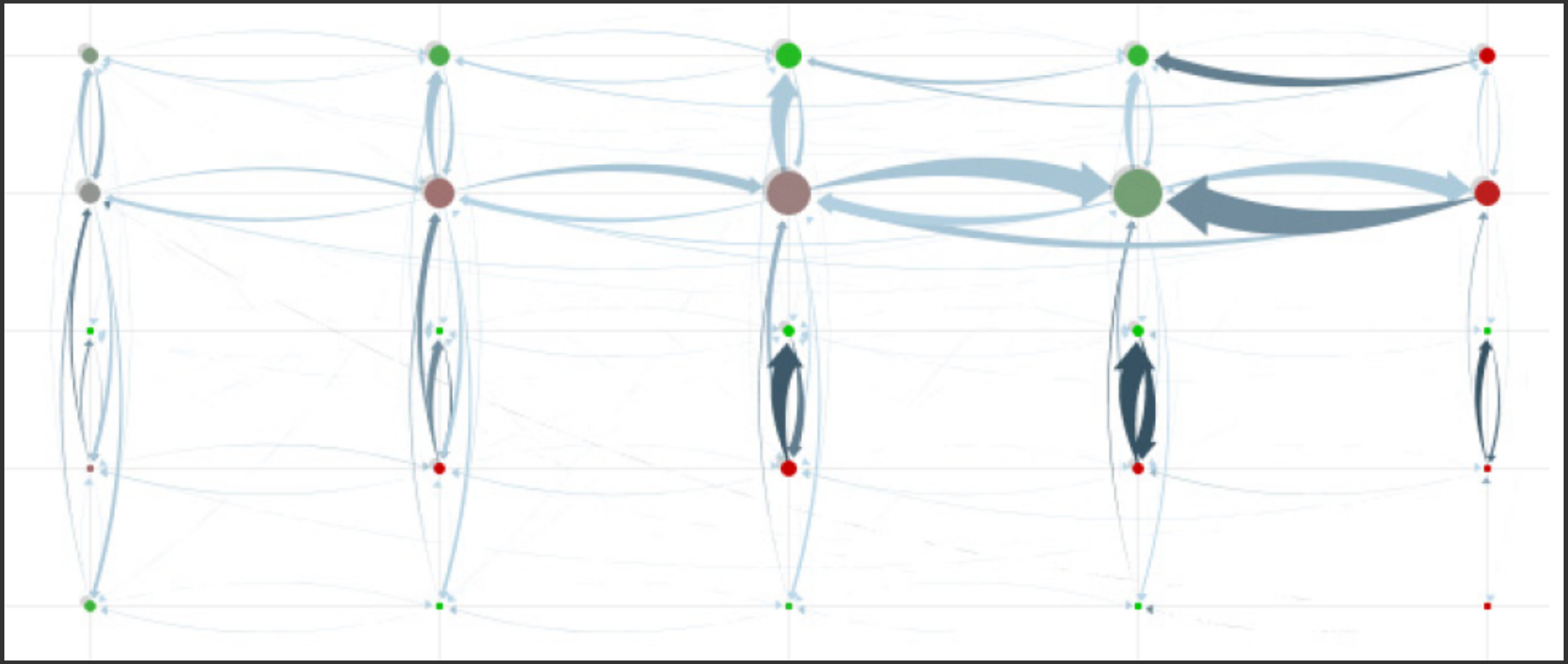


Node layout determined by geographic location.
Adjacent edges shown on node selection.

Semantic Substrates [Shneiderman 06]

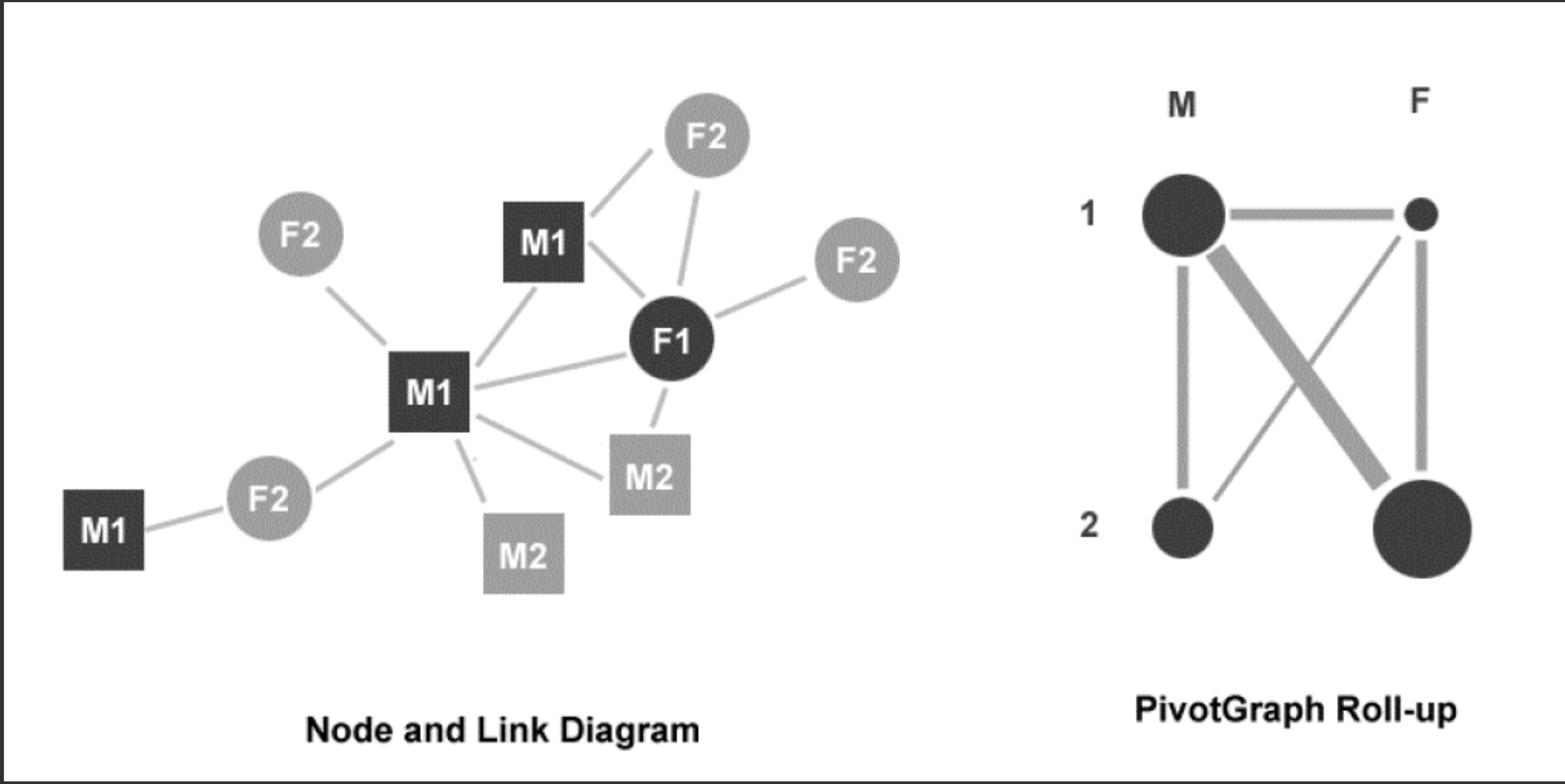


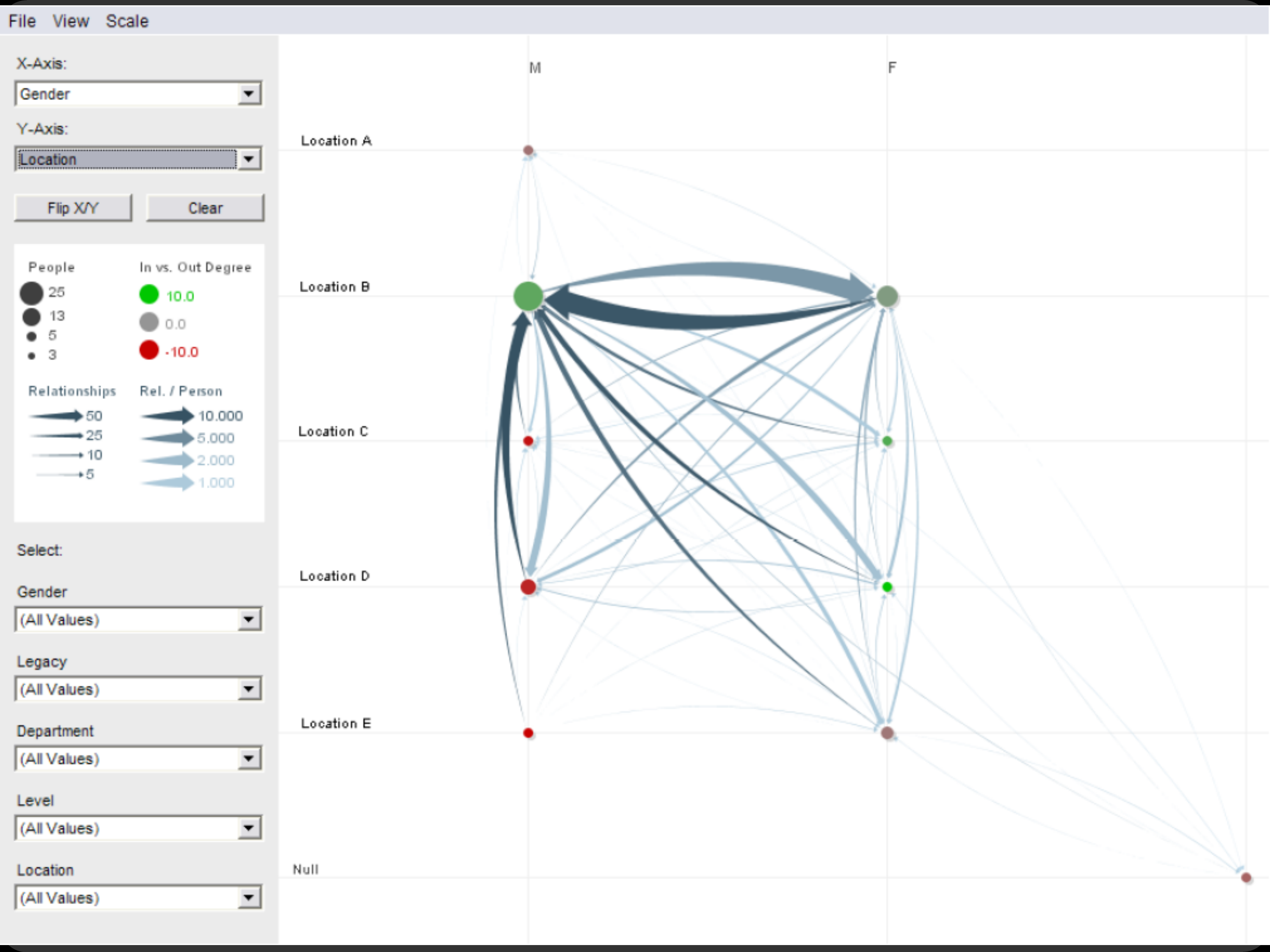
PivotGraph [Wattenberg '06]



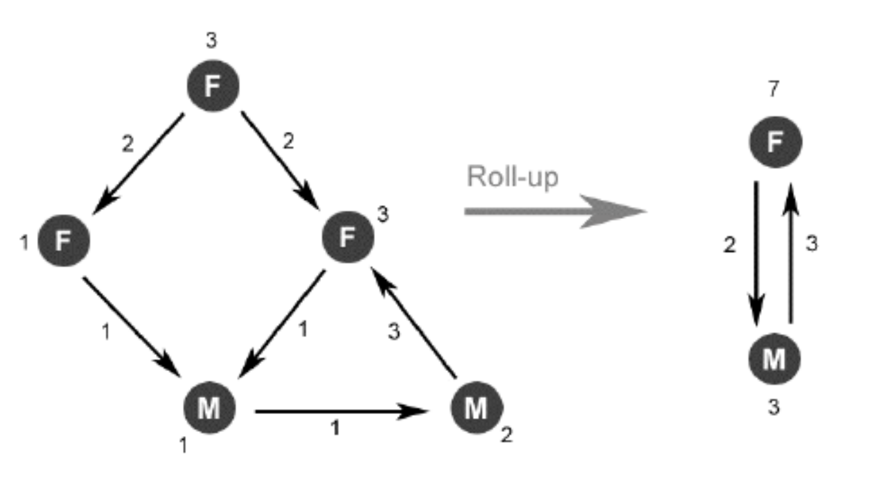
Layout aggregate graphs using node attributes.
Analogous to pivot tables and trellis display.

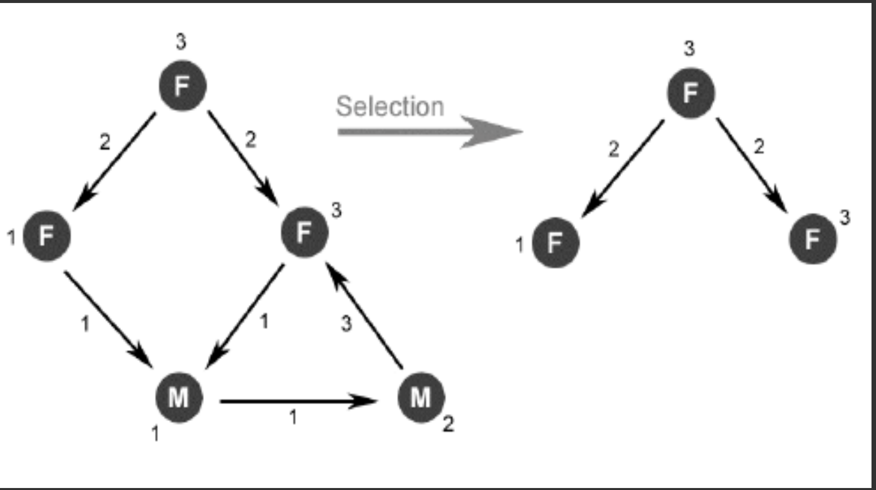
PivotGraph


F2
F2
M1
F2
F1
M1
M2
F2
M1
M2
Node and Link Diagram
M
F
1
2
PivotGraph Roll-up


File
View
Scale
X-Axis:
Gender
Y-Axis:
Location
Flip X/Y
Clear
People
25
13
5
3
In vs. Out Degree
10.0
0.0
-10.0
Relationships
50
25
10
5
Rel. / Person
10.000
5.000
2.000
1.000
Select:
Gender
(All Values)
Legacy
(All Values)
Department
(All Values)
Level
(All Values)
Location
(All Values)
M
F
Location A
Location B
Location C
Location D
Location E
Null

Operators



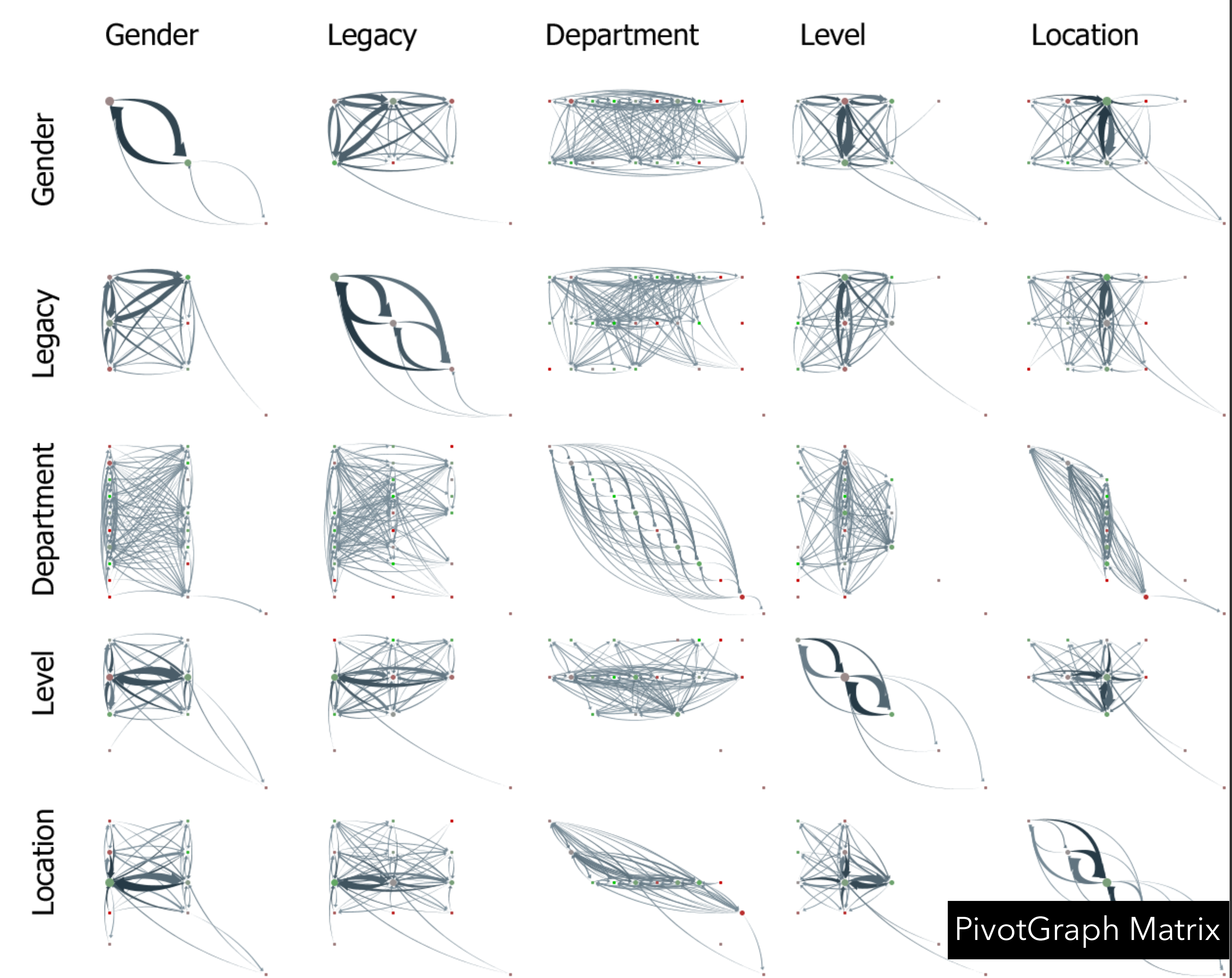


Roll-Up

Aggregate items with matching data values

Selection

Filter on data values


Gender
Legacy
Department
Level
Location
Gender
Legacy
Department
Level
Location
PivotGraph Matrix

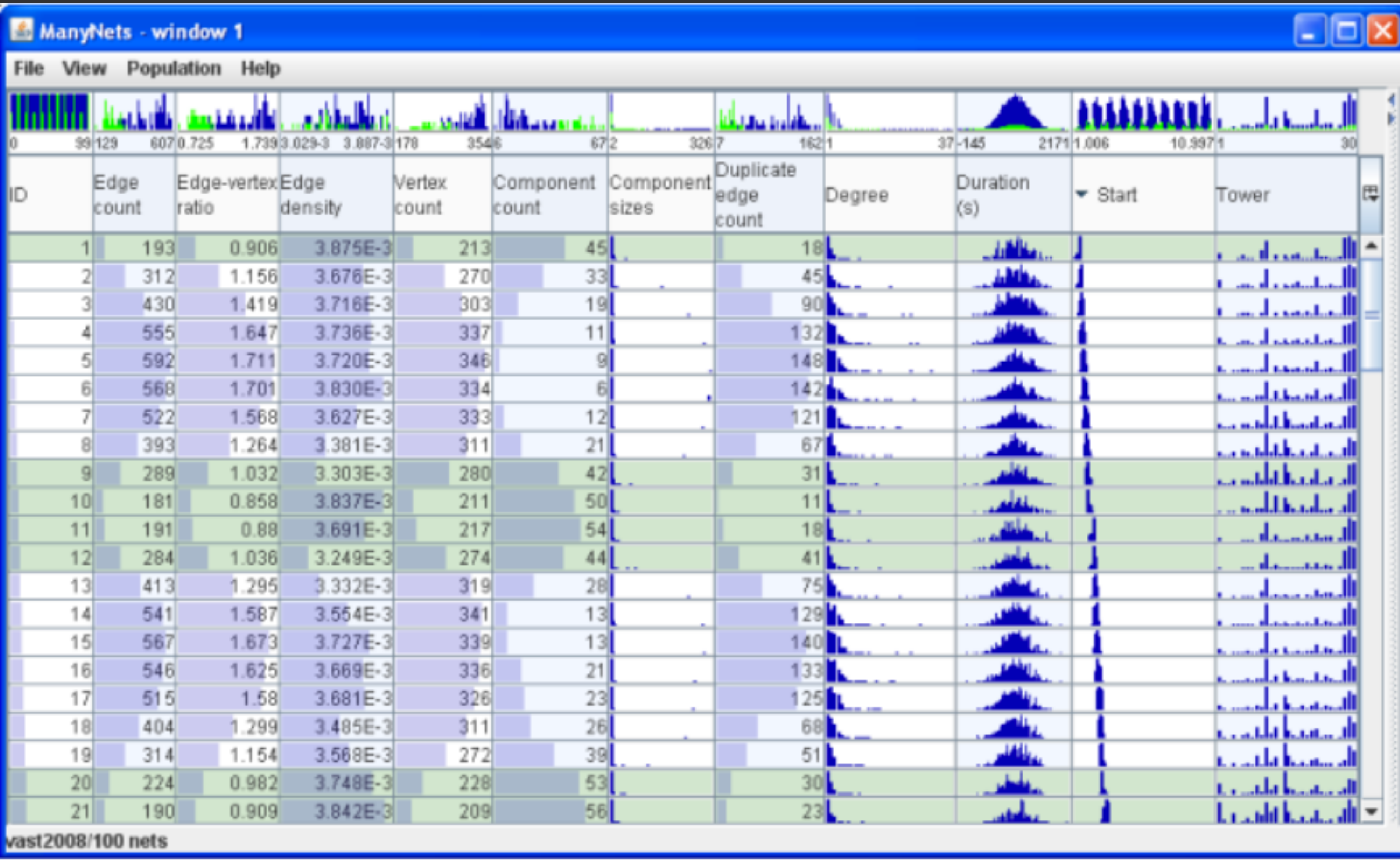
Limitations of PivotGraph

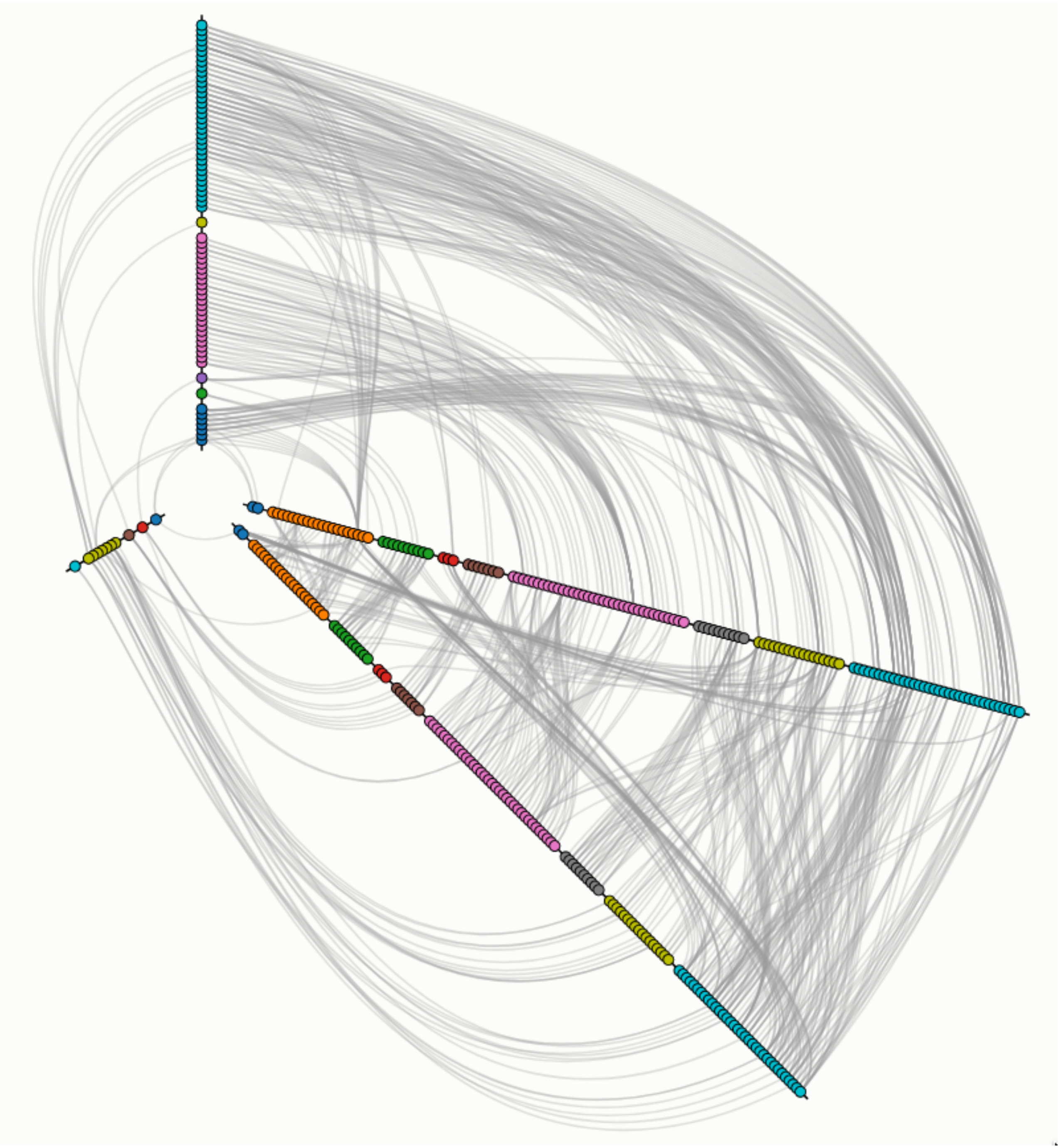
Only 2 variables (no nesting as in Tableau)

Doesn’t support continuous variables

Multivariate edges?

ManyNets [Freire et al. '10]





HivePlots

[Krzywinski '11]

Nodes (dots) may be replicated.

Nodes sorted on radial axes by network statistics (e.g., by degree).

Different axes may contain different subsets of nodes.

egweb.bcgsc.ca

Summary: Hierachies & Networks

Tree Layout

Indented / Node-Link / Enclosure / Layers

Focus+Context techniques for scale

Graph Layout

Spanning Tree Layout, “Sugiyama” Layout

Arc Diagrams

Force-Directed Layout, Optimization Methods

Matrix Diagrams

Attribute-Driven Layout