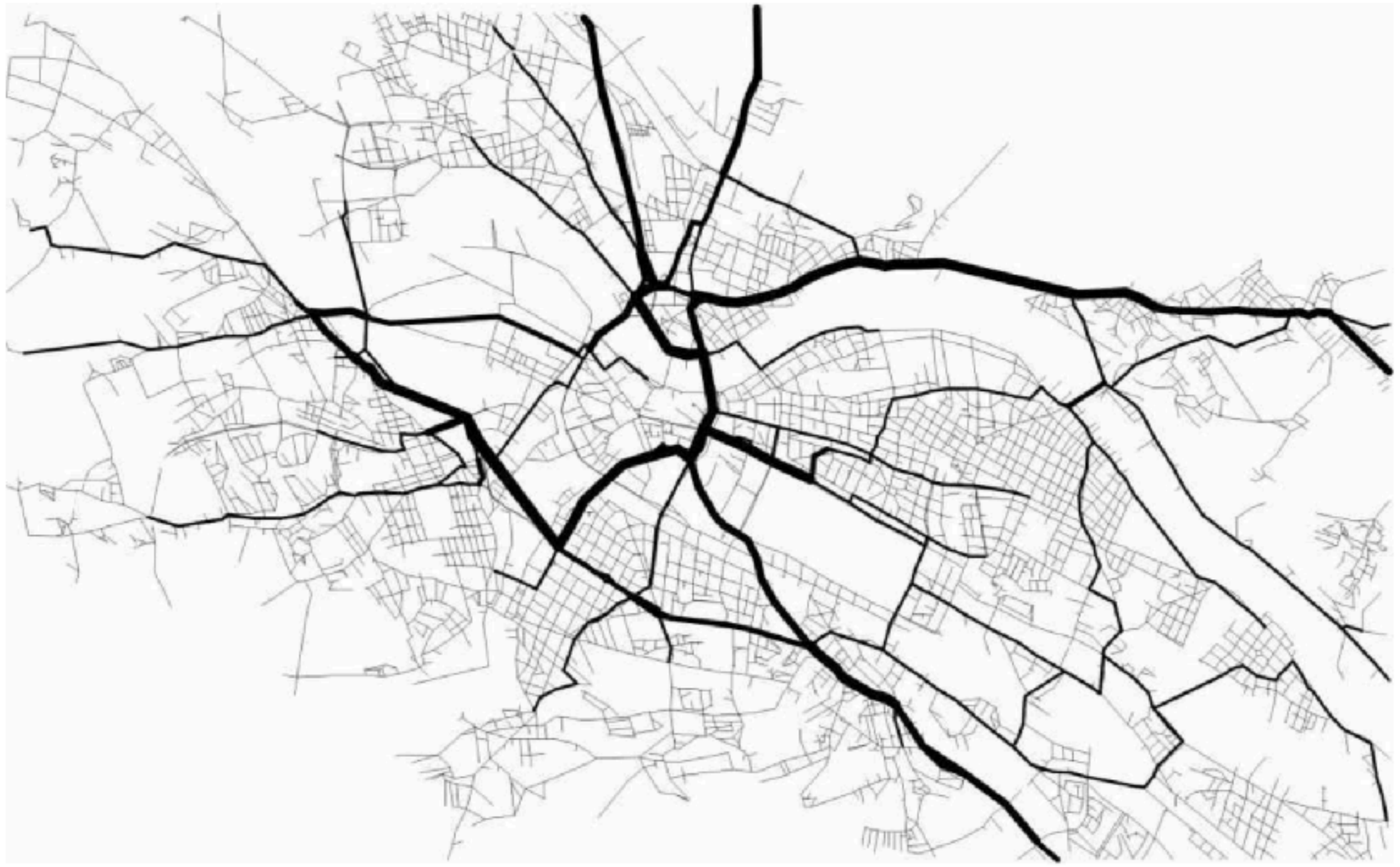
Geospatial Data Science, Spring 2022

Lecture 9: Spatial networks

Instructor: Michael Szell
April 7, 2022

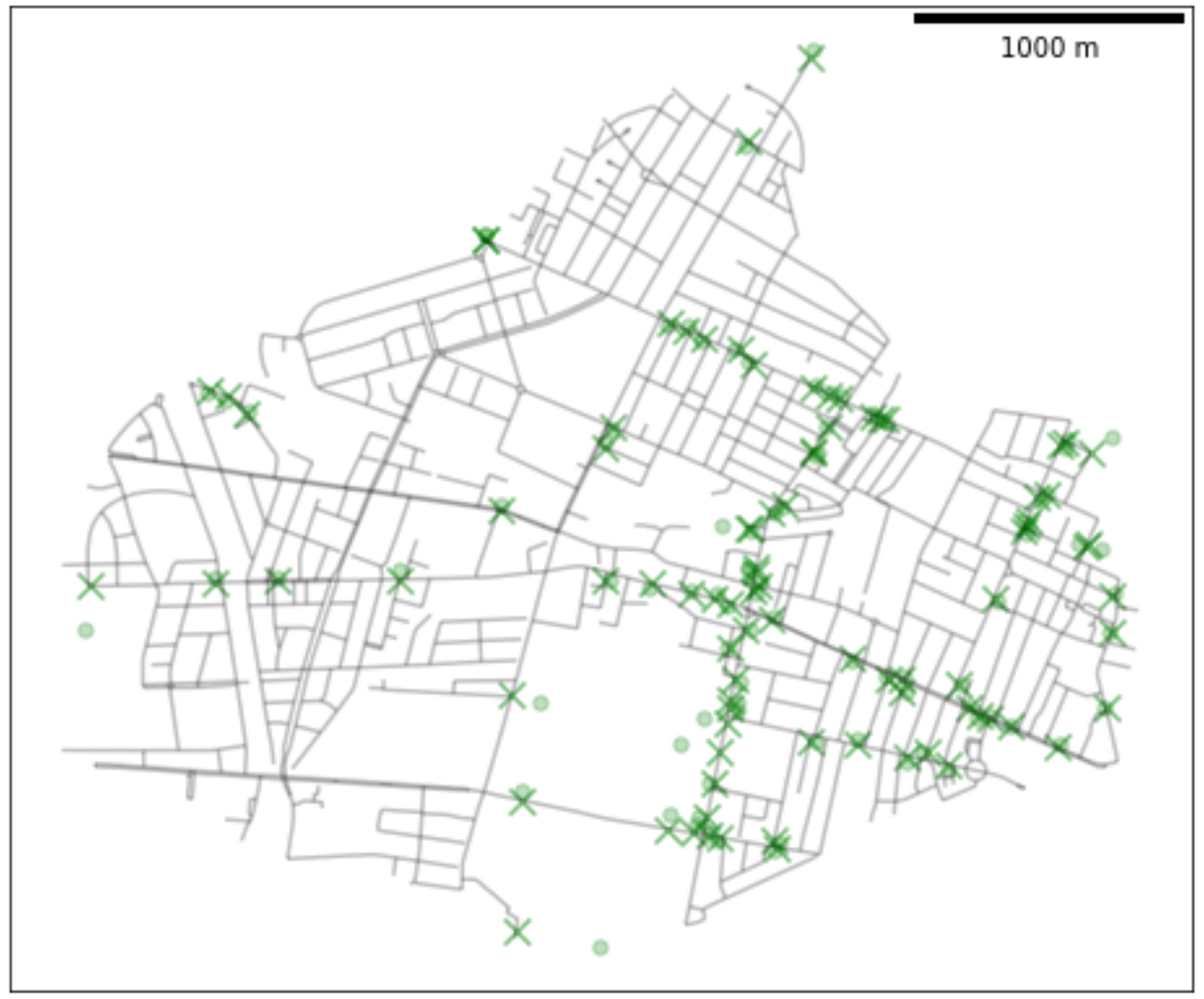


IT UNIVERSITY OF COPENHAGEN

Today you will learn about spatial networks

Difference to non-spatial networks

spaghetti



Optimal + transport networks and metrics

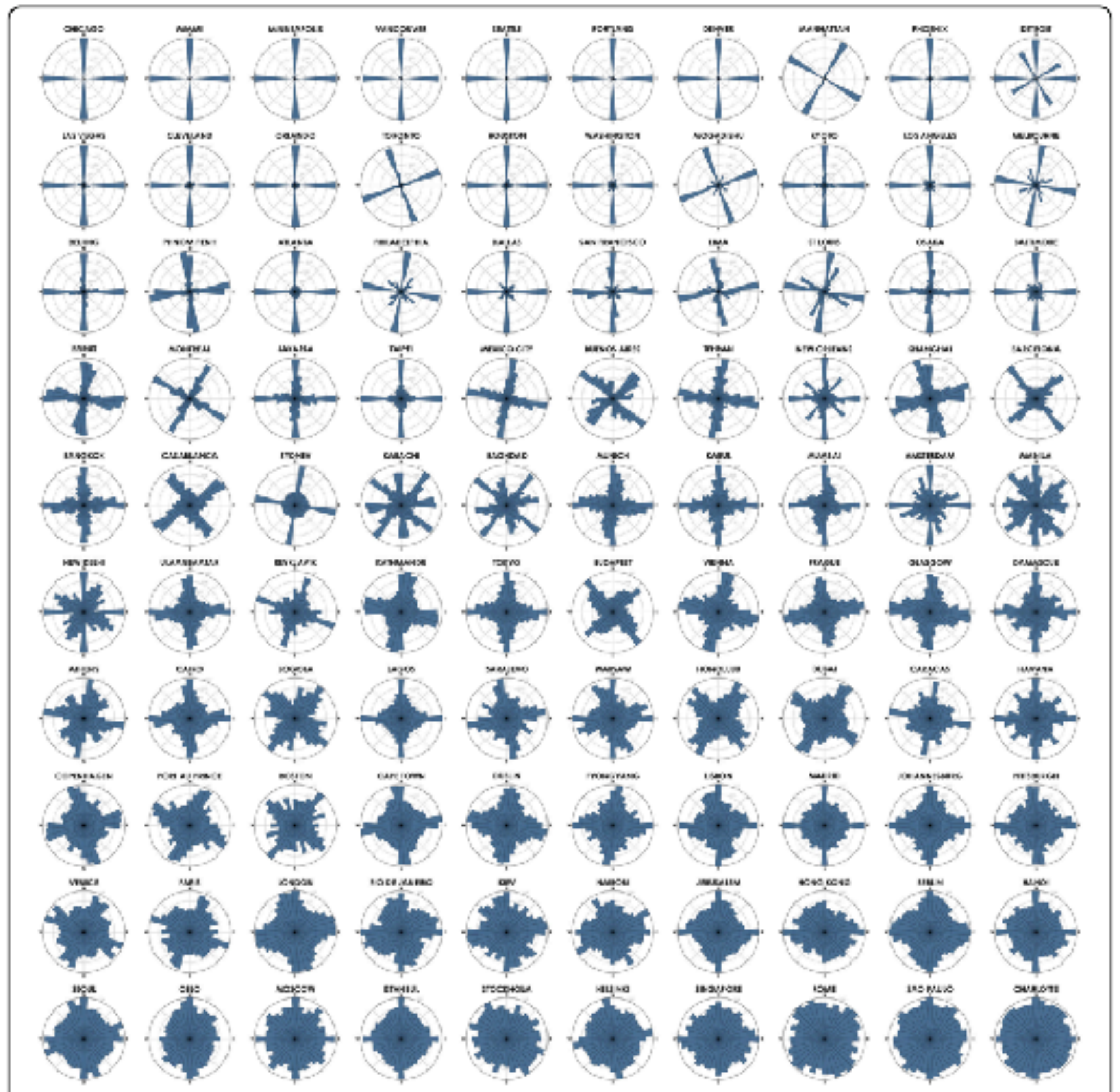
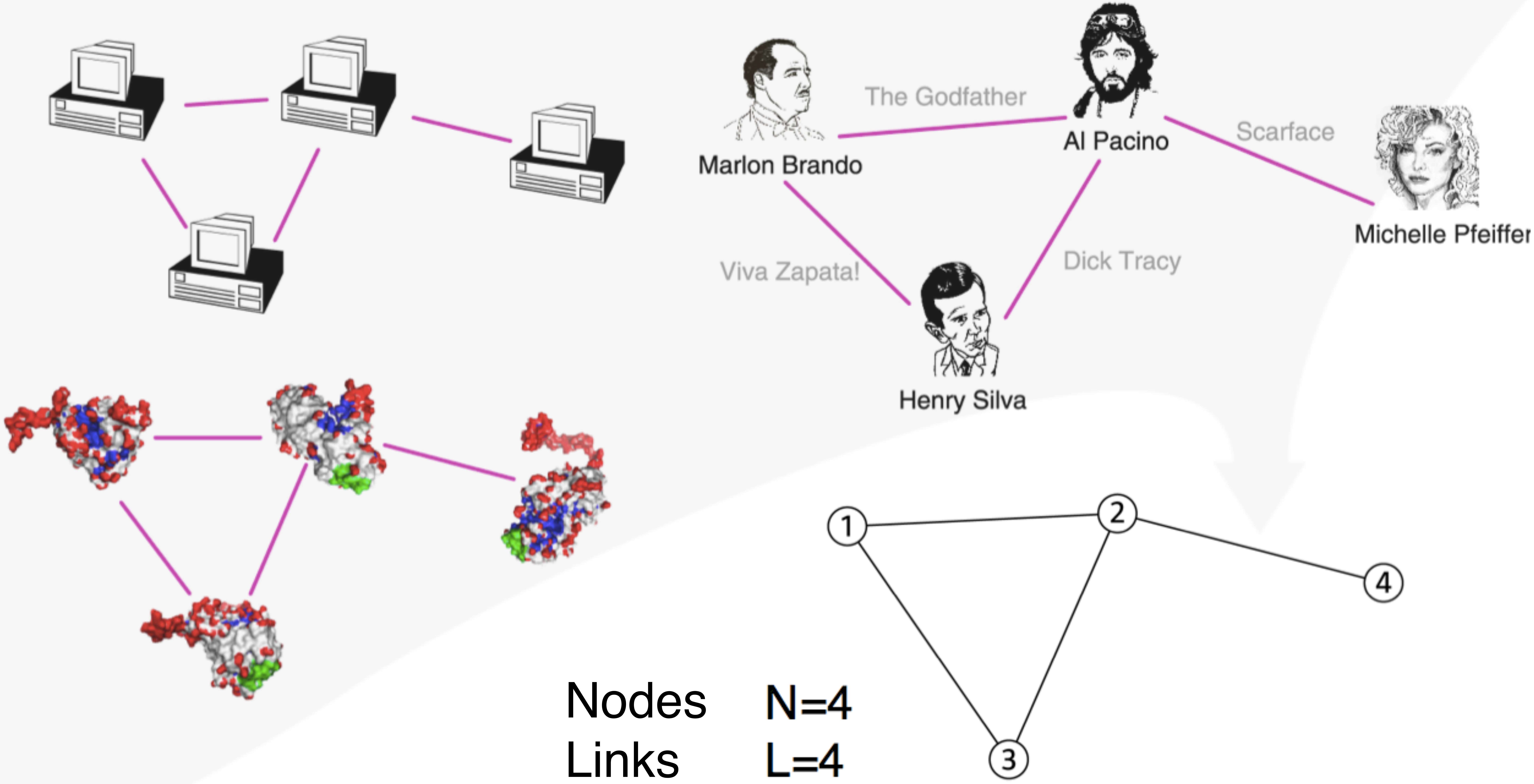


Fig. 5 Polar histograms from Fig. 4, resorted by descending φ from most to least grid-like (equivalent to least to greatest entropy)

Networks are a common language for different applications

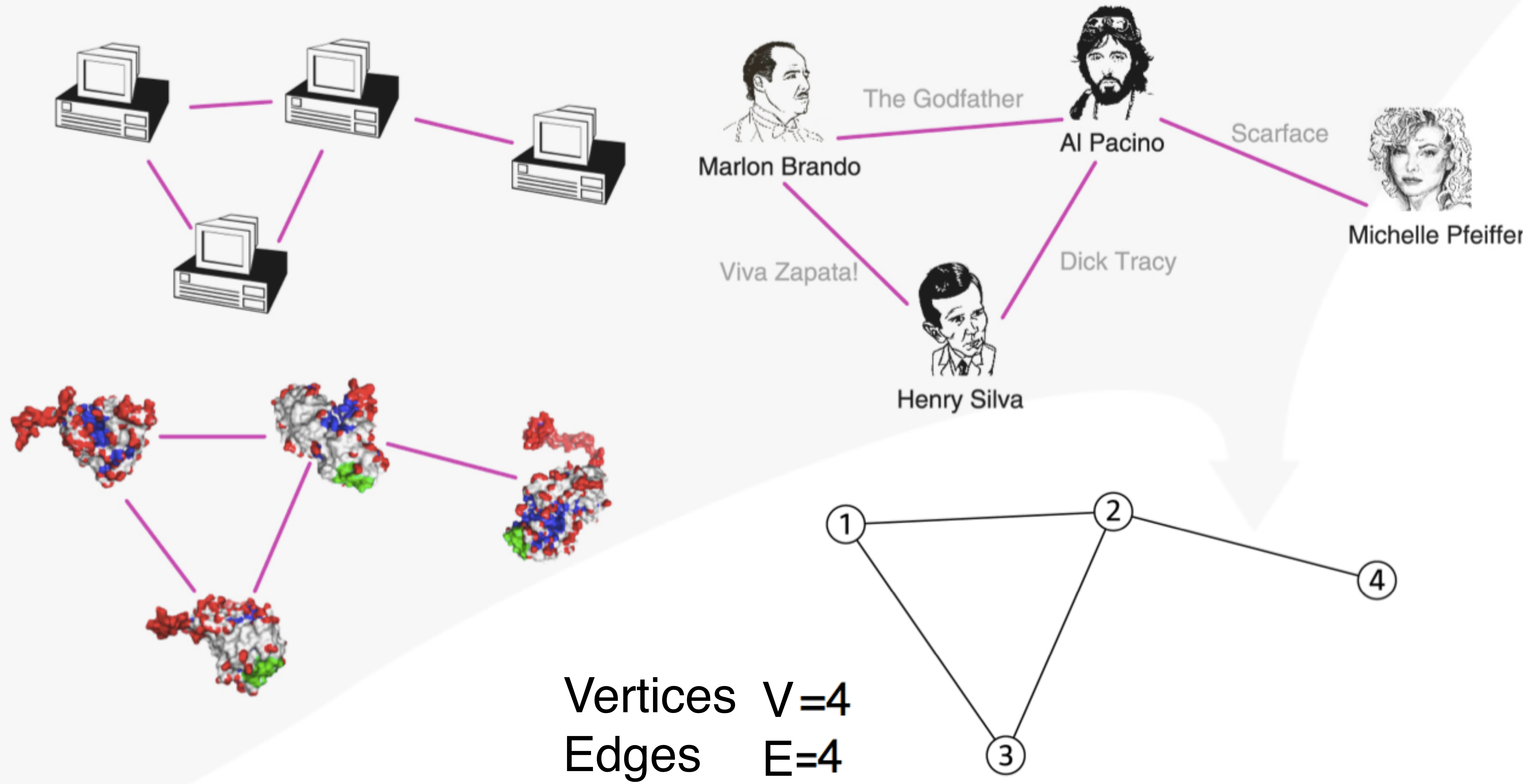


Nodes N=4

Links L=4

Graph $\mathcal{G} = (\mathcal{N}, \mathcal{L})$

Network = Graph + real-world meaning

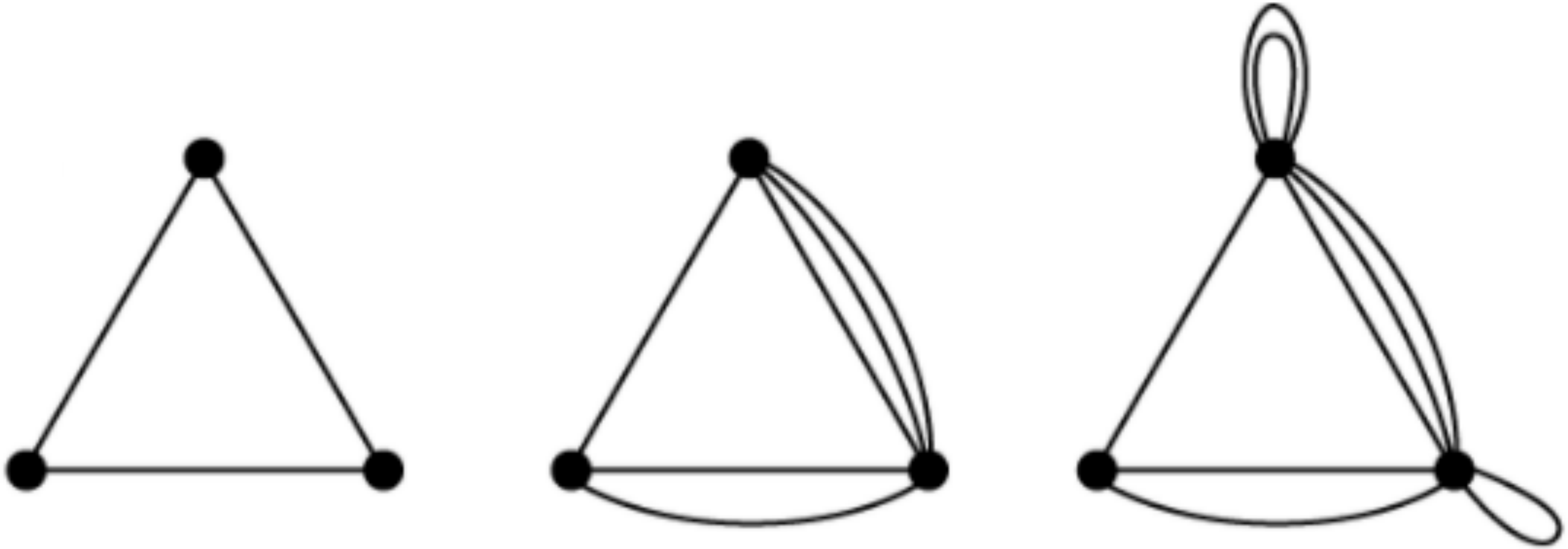


Vertices $V=4$

Edges $E=4$

Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$

Real spatial networks are usually pseudographs



Simple graph

+ parallel links

= Multigraph

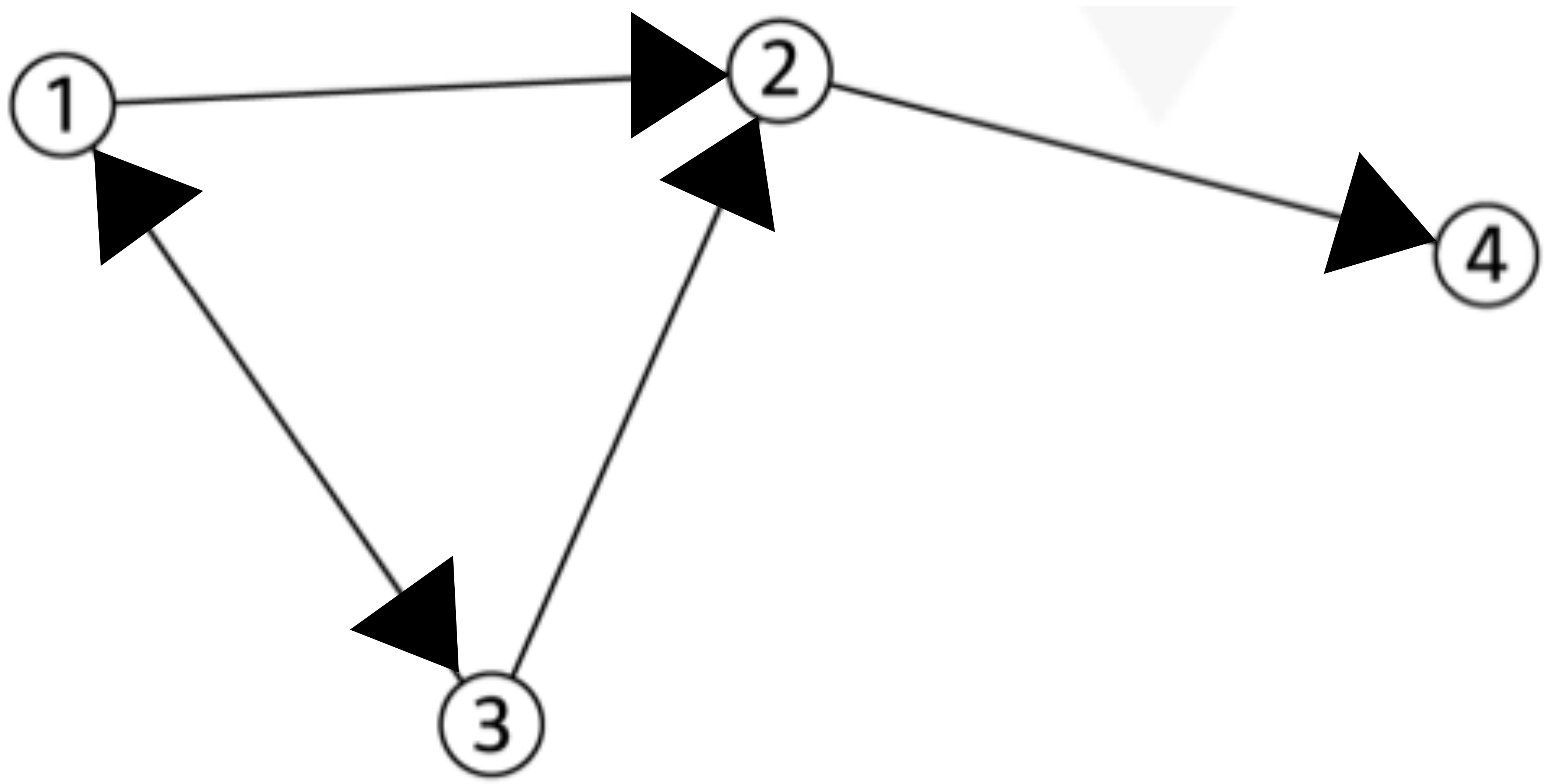
+ self-loops

= Pseudograph

OSMnx calls this MultiGraph

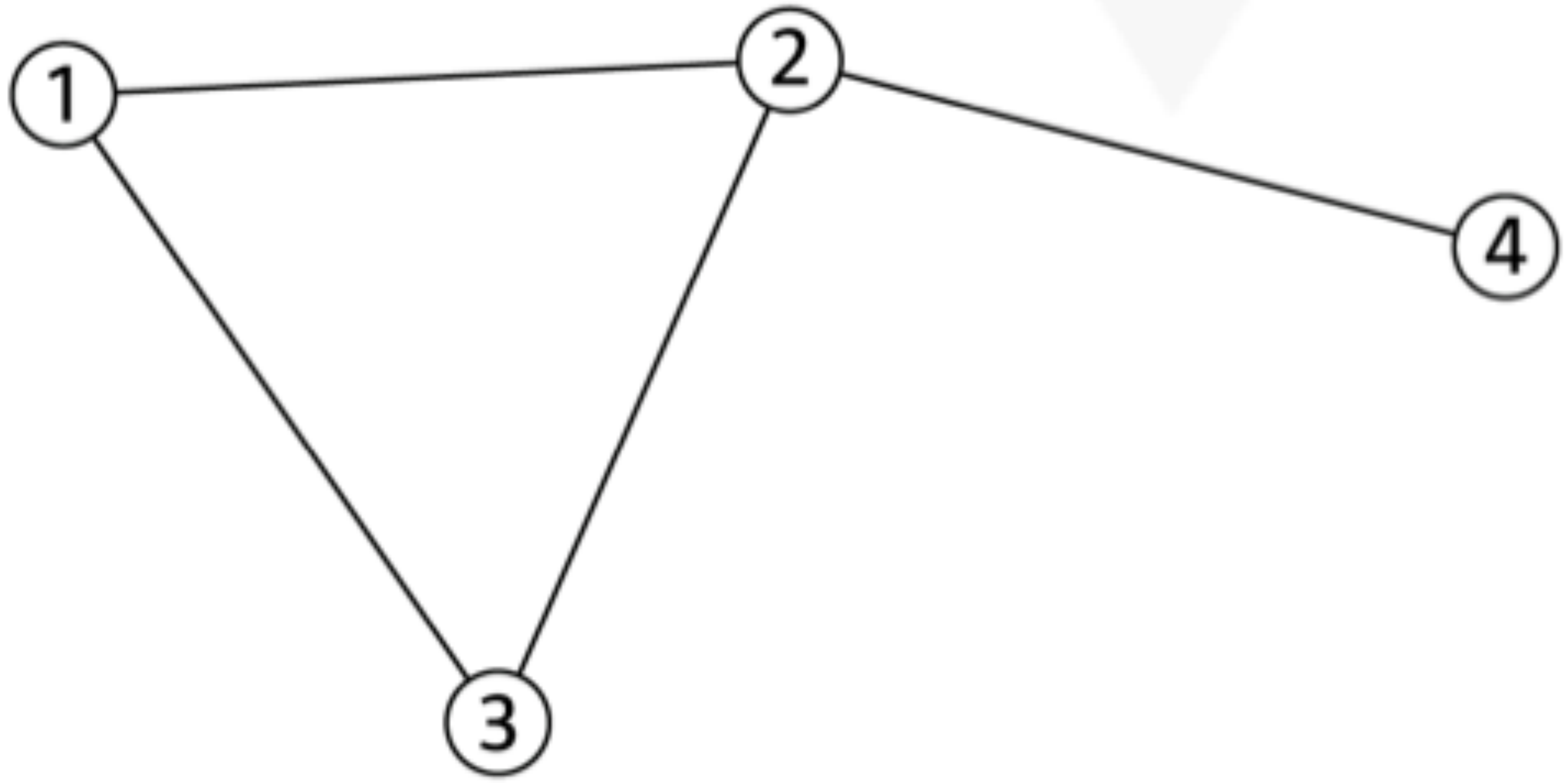
A directed graph (digraph) has links with a direction

Nodes N=4
Links (Arcs) L=5



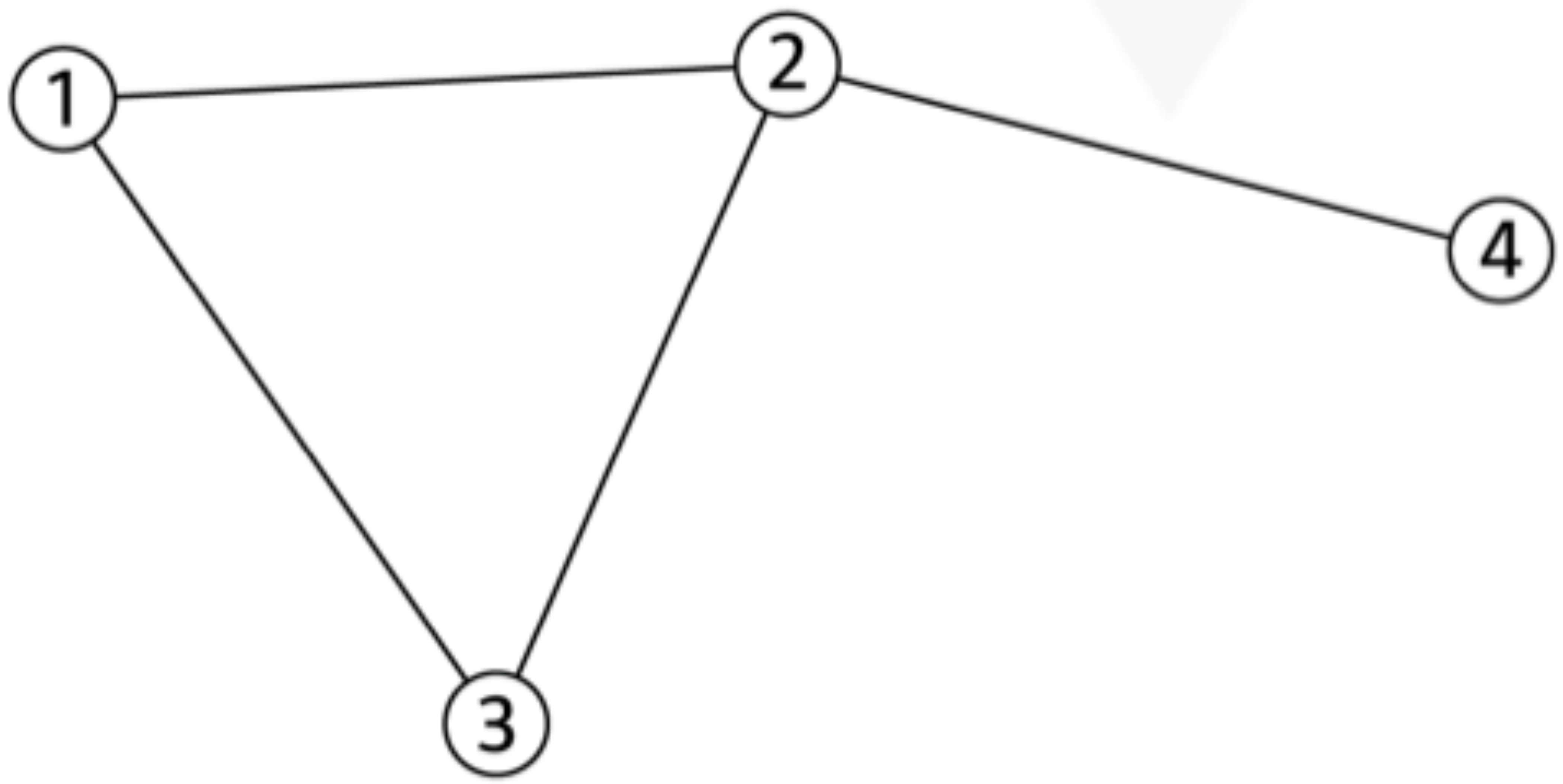
The degree k_i of a node i is the number of incident links

$$k_1 = 2 \qquad k_3 = 2$$
$$k_2 = 3 \qquad k_4 = 1$$



Every network has an average degree $\langle k \rangle$

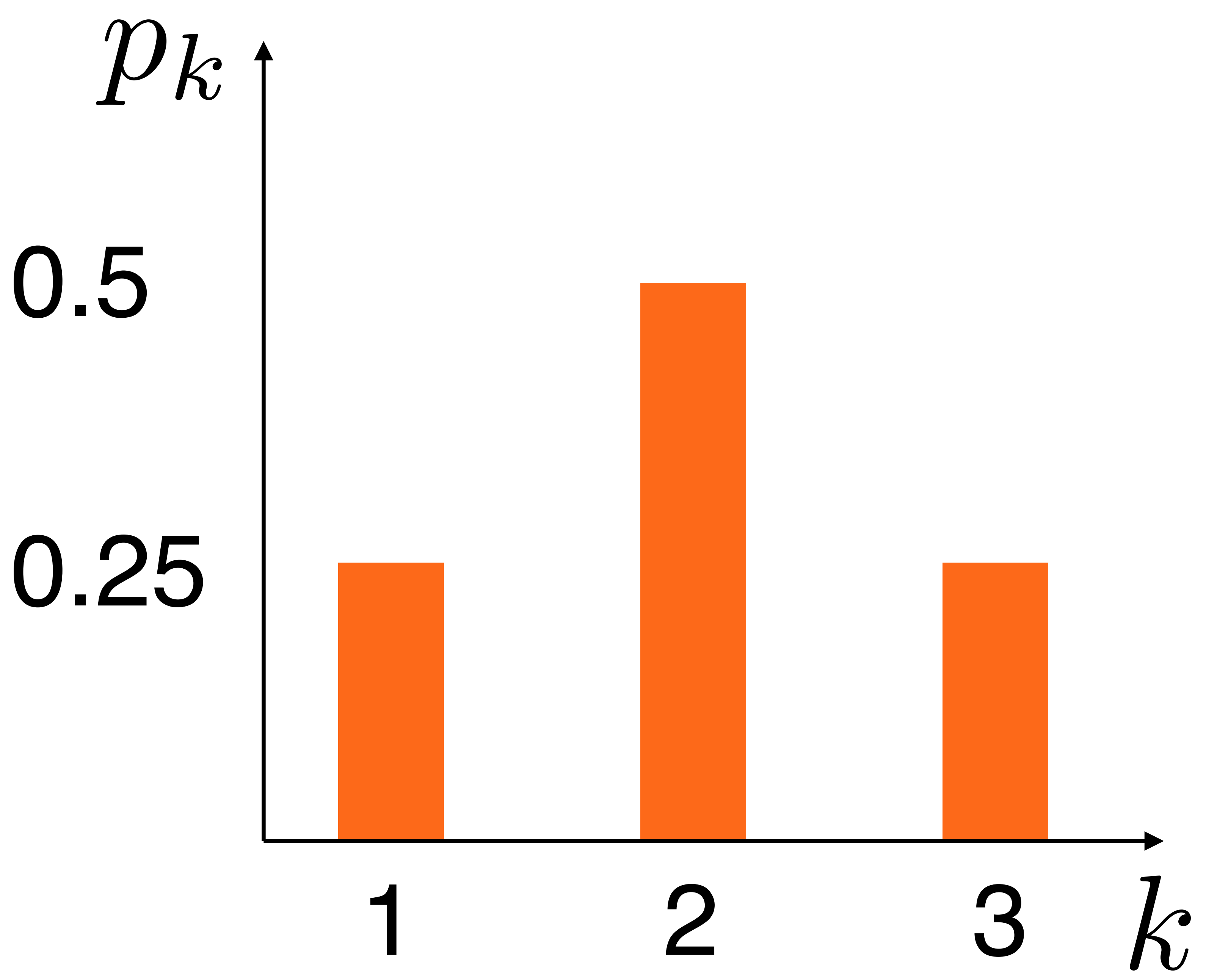
$$k_1 = 2 \qquad k_3 = 2$$
$$k_2 = 3 \qquad k_4 = 1$$

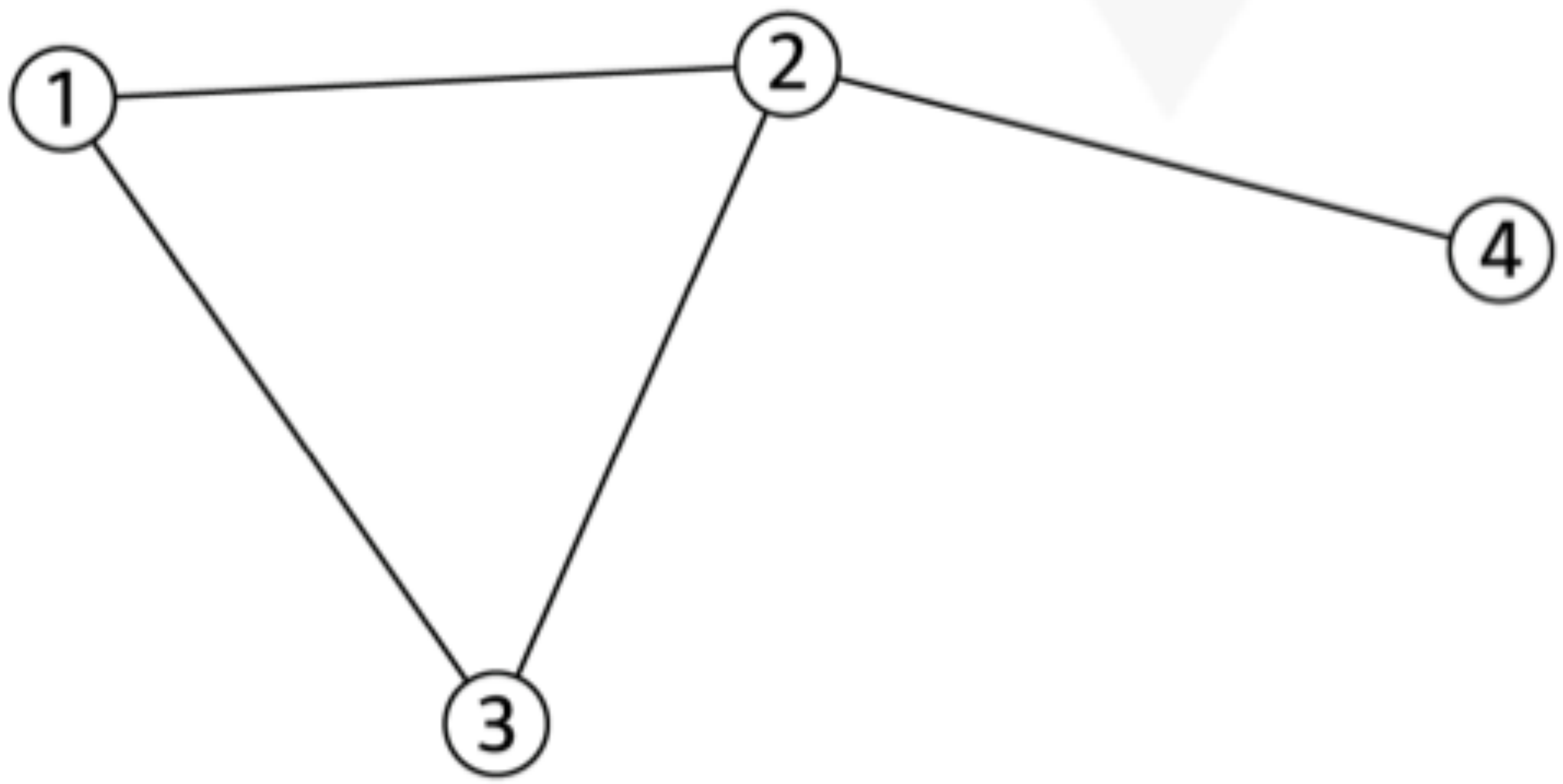


$$\langle k \rangle = \frac{1}{N} \sum_{i=1}^{N} k_i \qquad \langle k \rangle = \frac{2+3+2+1}{4} = 2$$

The degree distribution $p_k = N_k/N$ captures the probabilities that a node has a certain degree

$$k_1 = 2 \qquad k_3 = 2$$
$$k_2 = 3 \qquad k_4 = 1$$





Usually:

A network's degree distribution tells us something fundamental about how individuals in the system connect

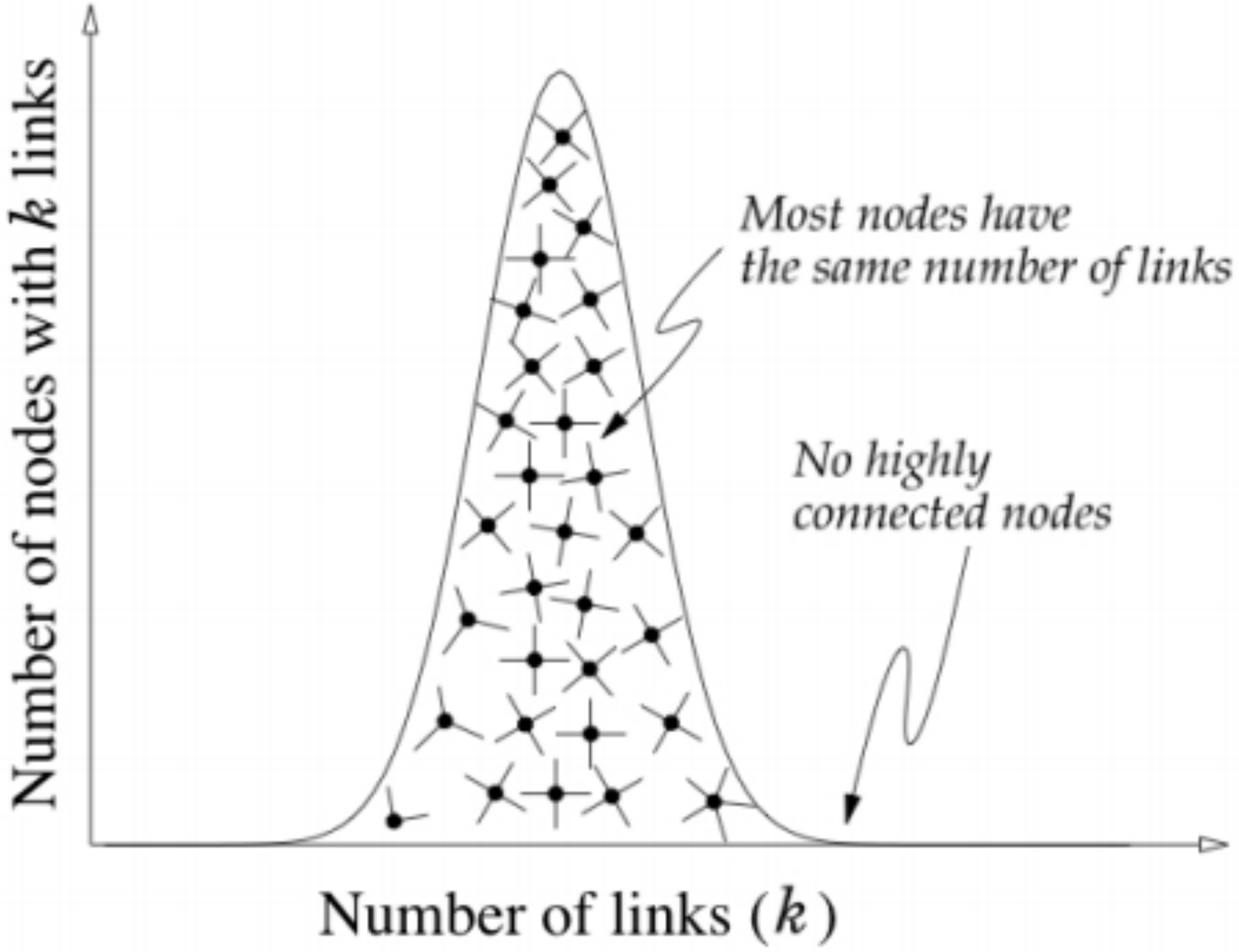
Usually:

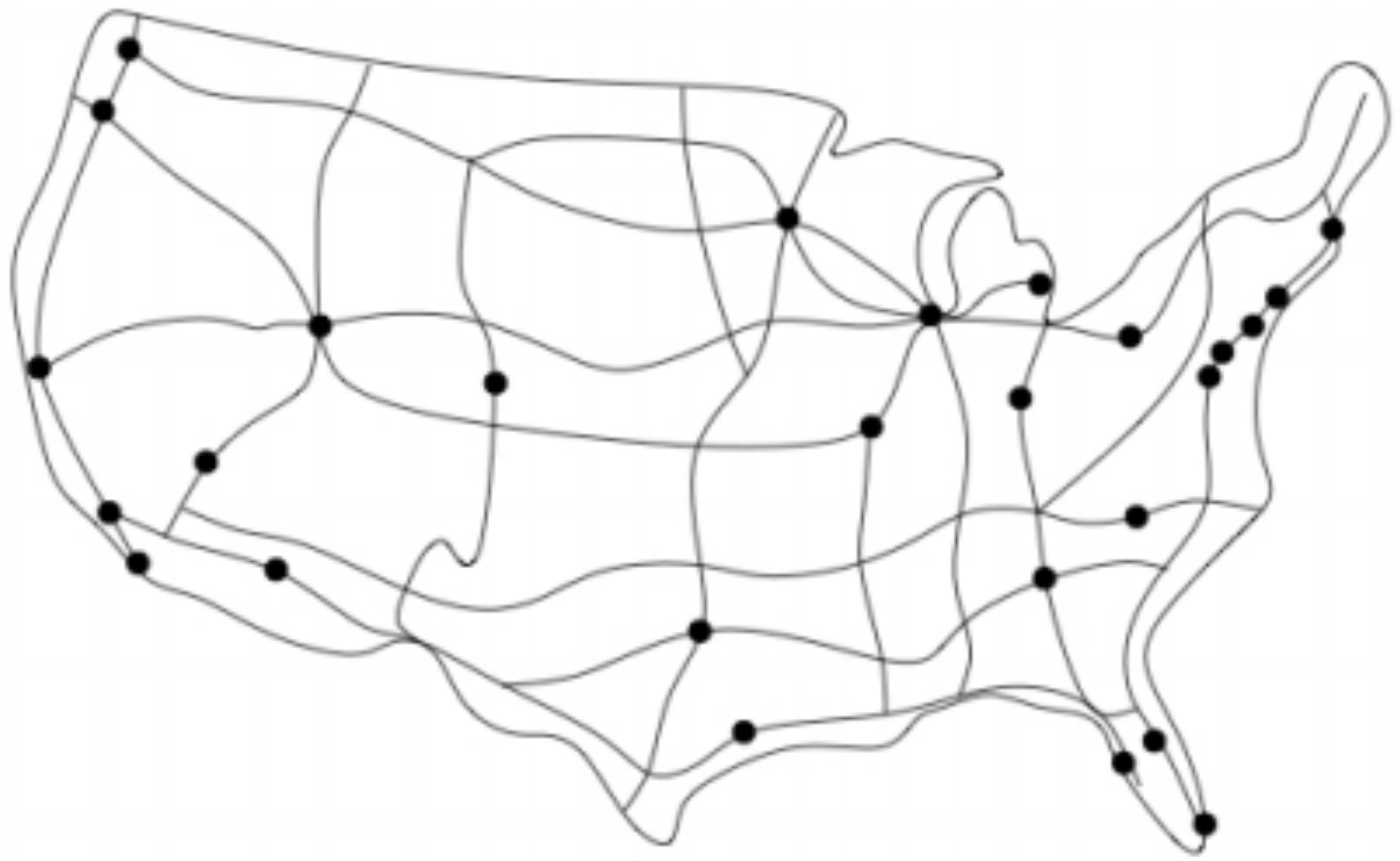
A network's degree distribution tells us something fundamental about how individuals in the system connect

However:

In many spatial networks the degree distribution is far less interesting

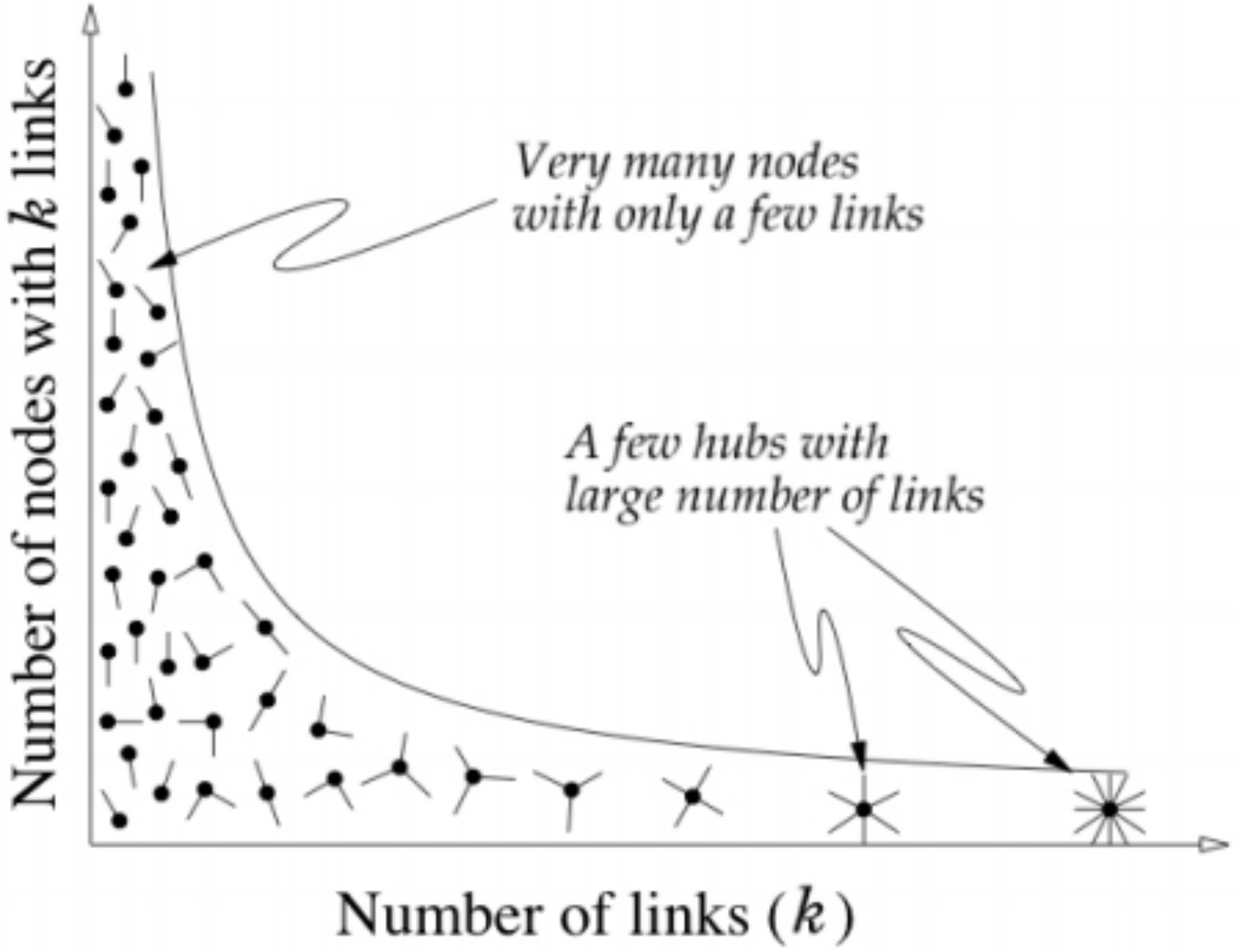
Thin-tailed network

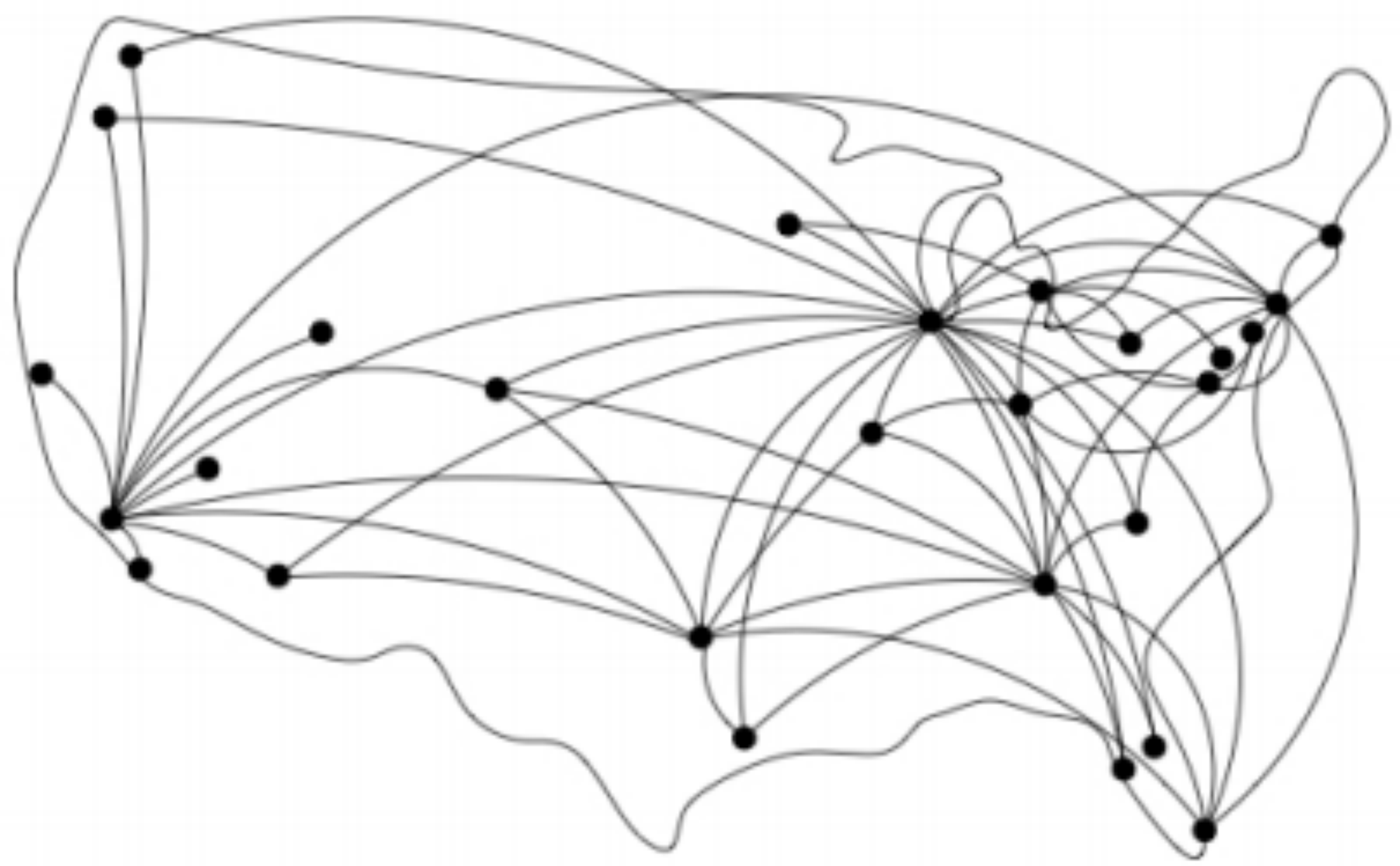




Streets

Heavy-tailed network



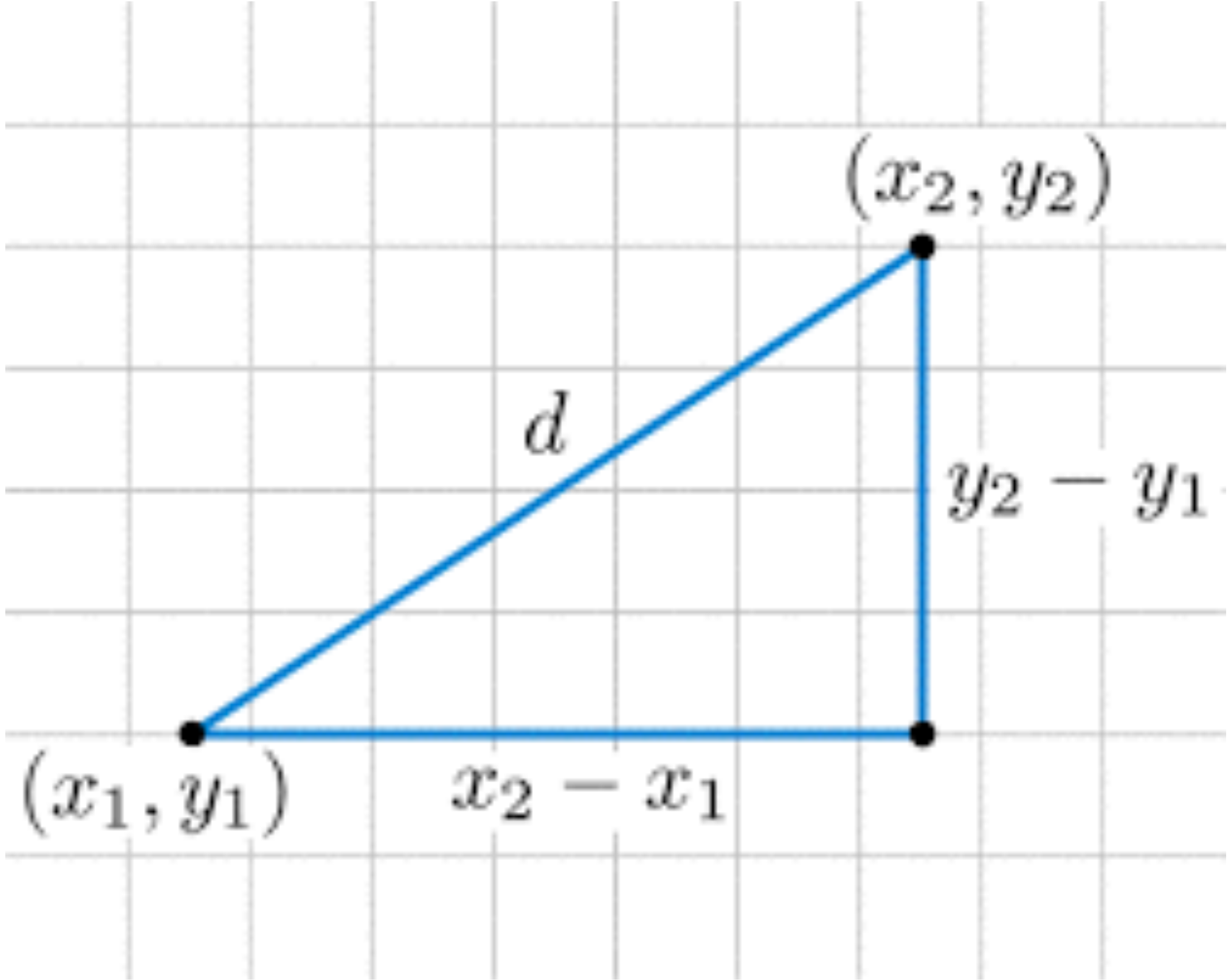


Airlines

A spatial network has nodes in a metric space

You can measure a distance between nodes

In practice: 2D-space with Euclidian distance



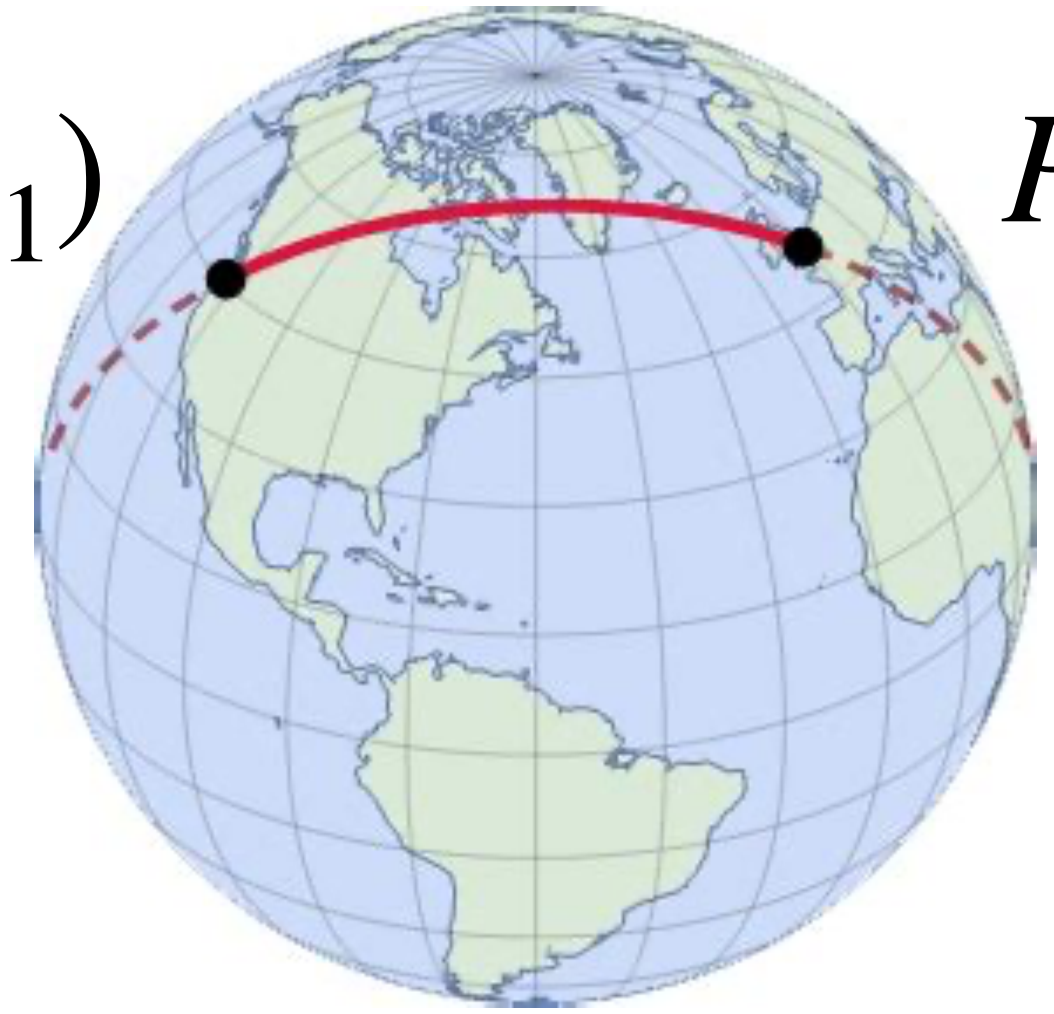
$$d(P_1, P_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

A spatial network has nodes in a metric space

You can measure a distance between nodes

For larger distances on a sphere use the Haversine distance

$P_1 = (\varphi_1, \lambda_1)$



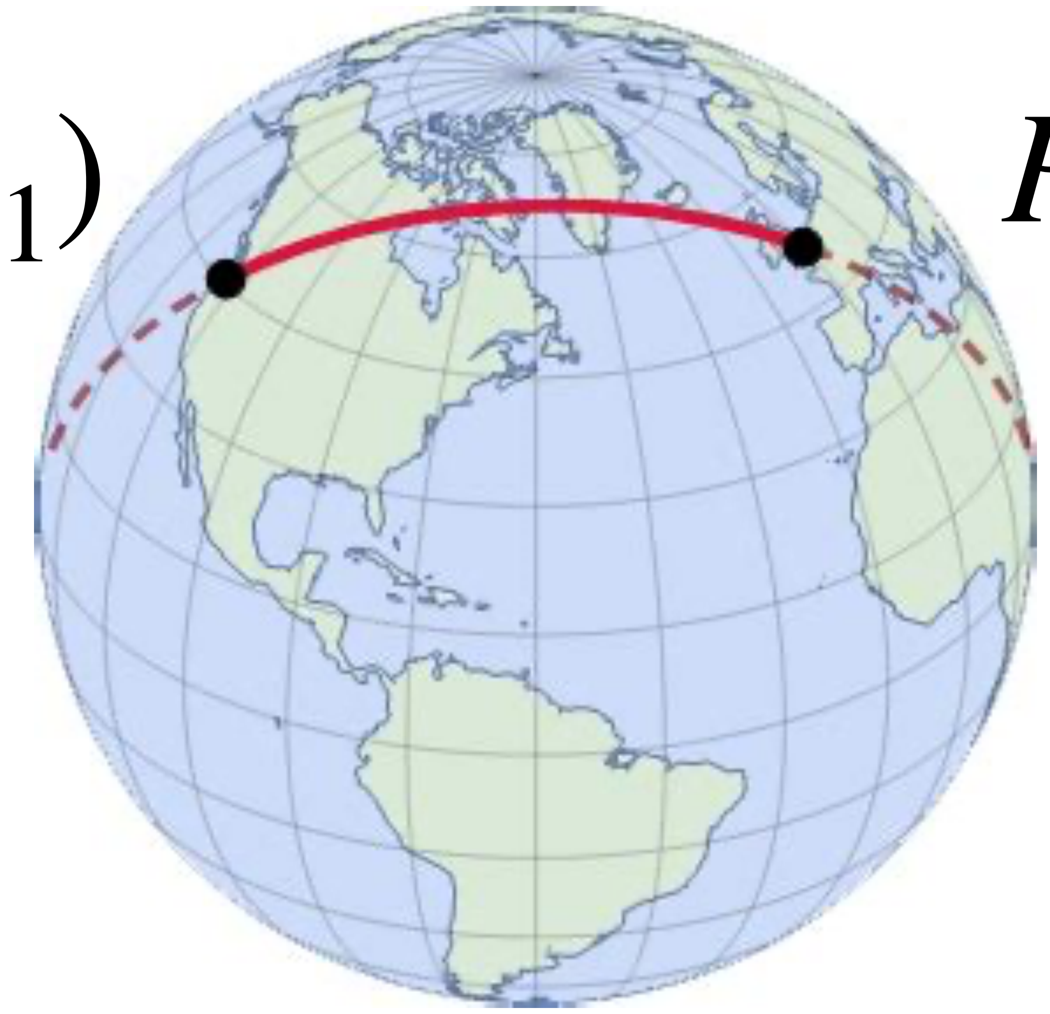
$P_2 = (\varphi_2, \lambda_2)$

A spatial network has nodes in a metric space

You can measure a distance between nodes

For larger distances on a sphere use the Haversine distance

$P_1 = (\varphi_1, \lambda_1)$



$P_2 = (\varphi_2, \lambda_2)$

$$d = 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\varphi_2 - \varphi_1}{2}\right) + \cos\varphi_1 \cdot \cos\varphi_2 \cdot \sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right)$$

Planarity

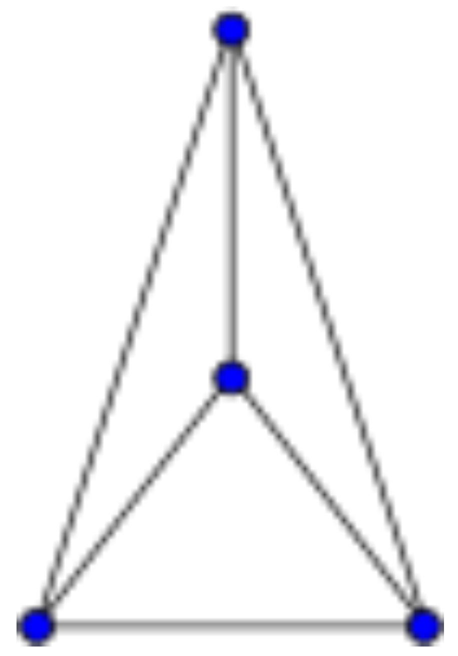
Three utilities problem: Can you link each house to each utility?

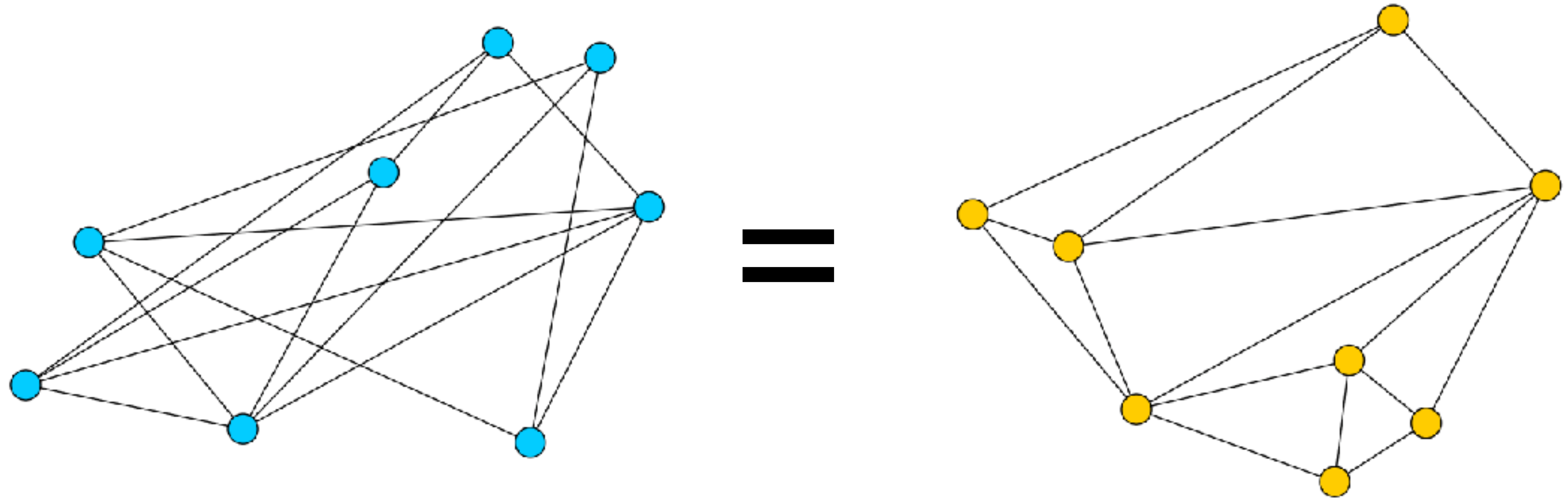
Three utilities problem: Can you link each house to each utility?

100+ years old, was solved in 1930 by Kuratowski: No

A graph is planar is if can be emdedded in the plane

Planar: You can move the nodes around until no links cross

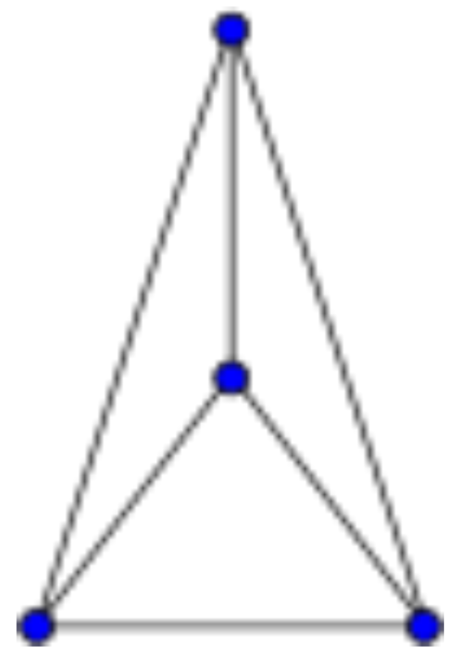


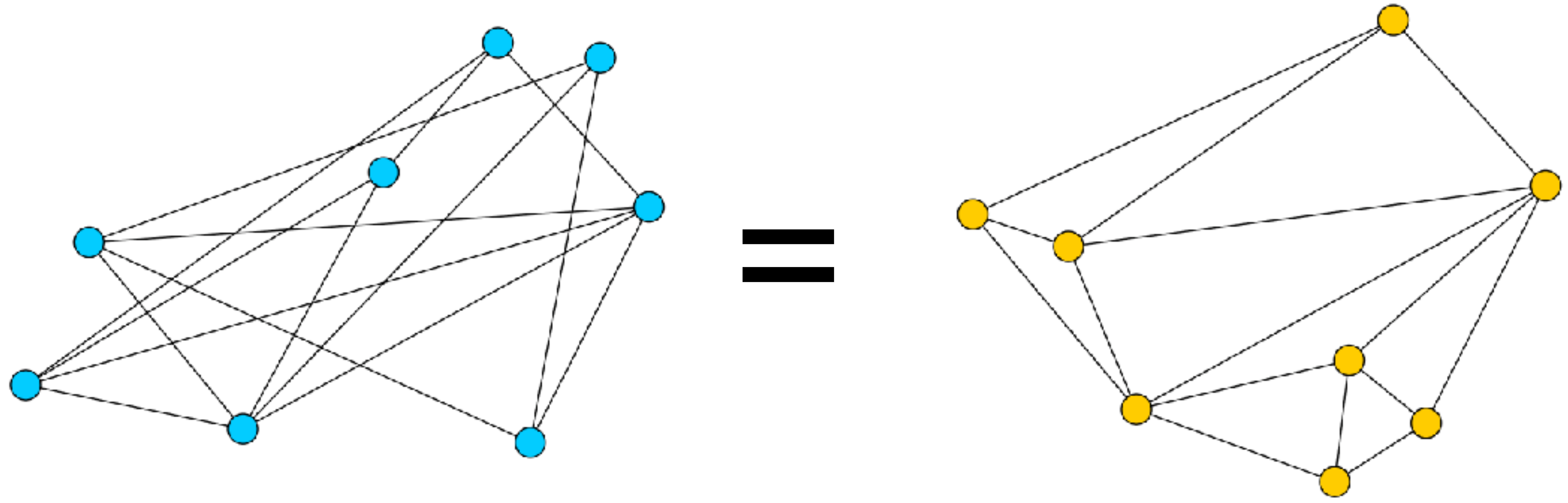


https://www.jasondavies.com/planarity/

A graph is planar is if can be emdedded in the plane

Planar: You can move the nodes around until no links cross

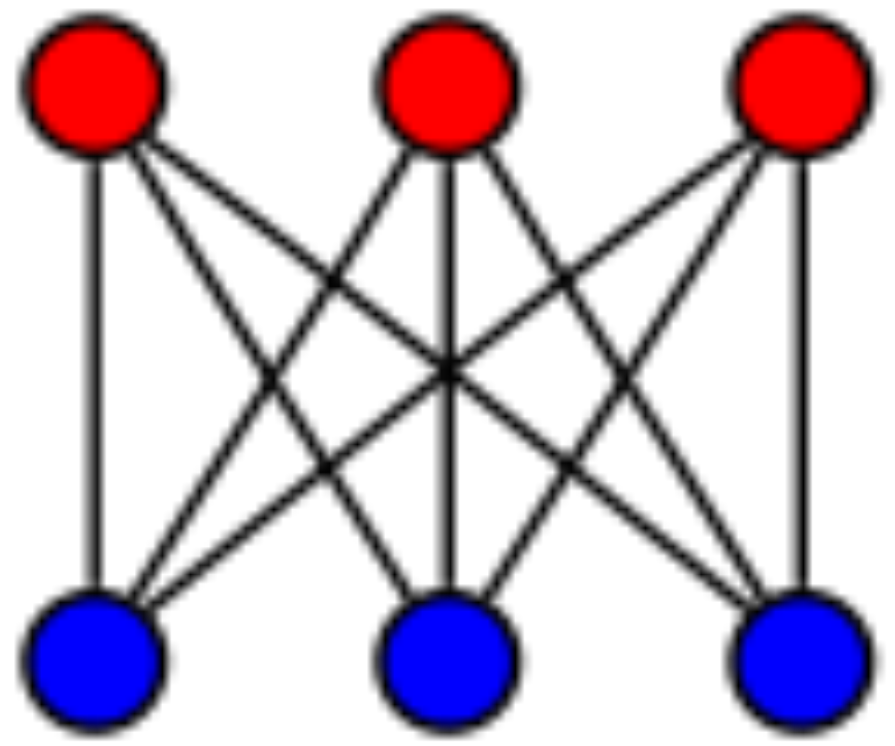


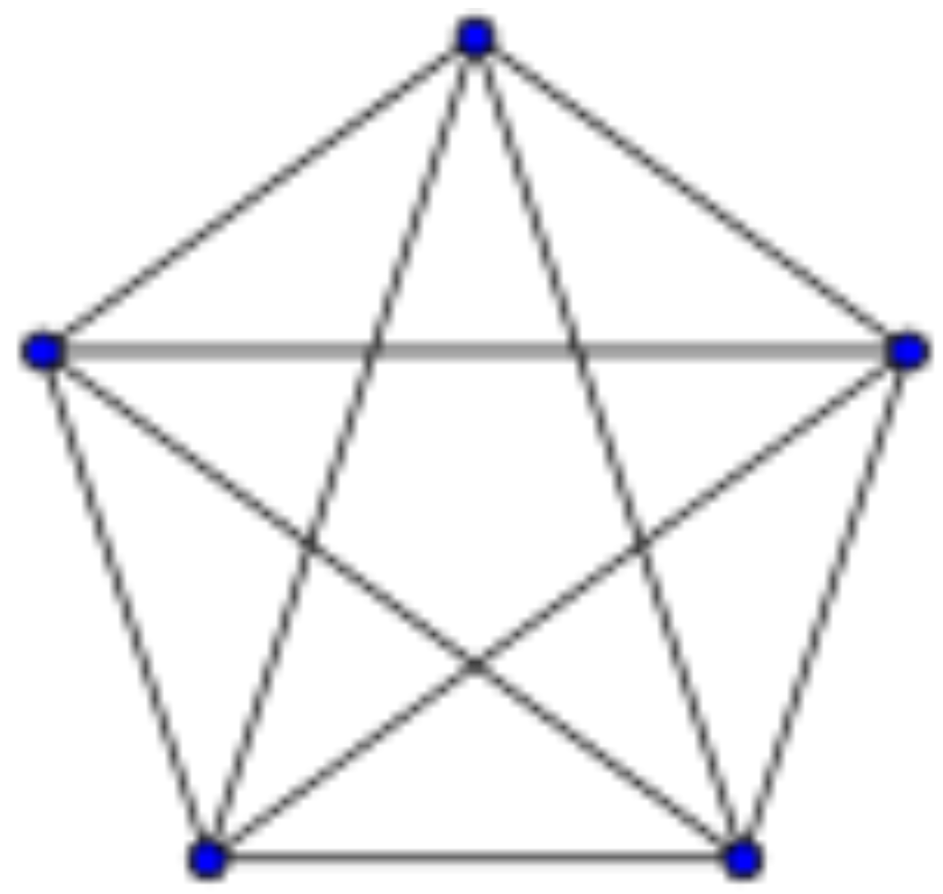


https://www.jasondavies.com/planarity/

Not planar: you can't

Utility graph



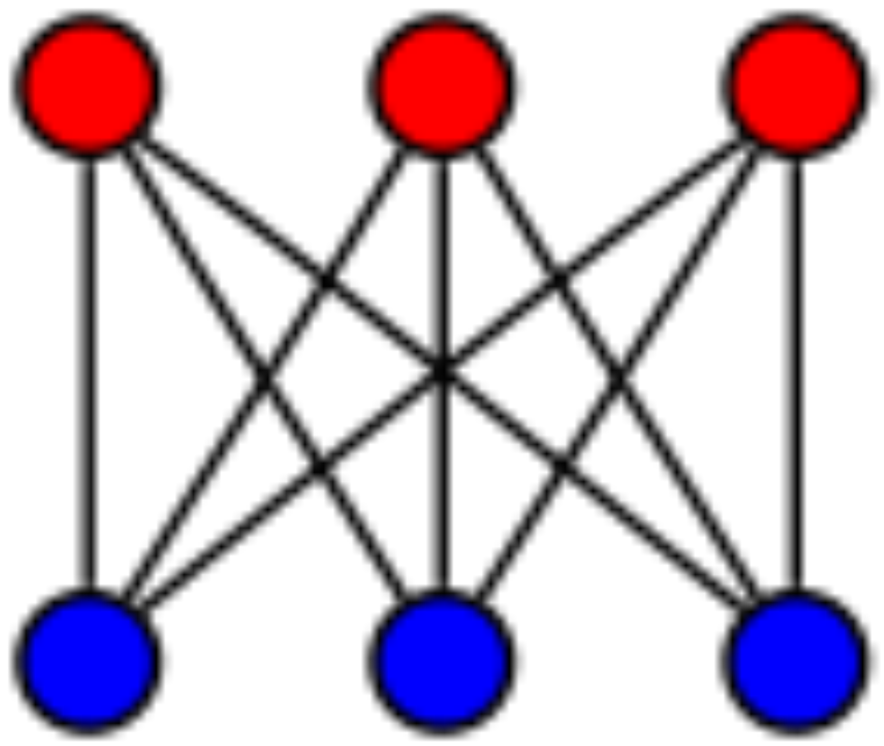


Kuratowskis's theorem gives an easy way to check planarity

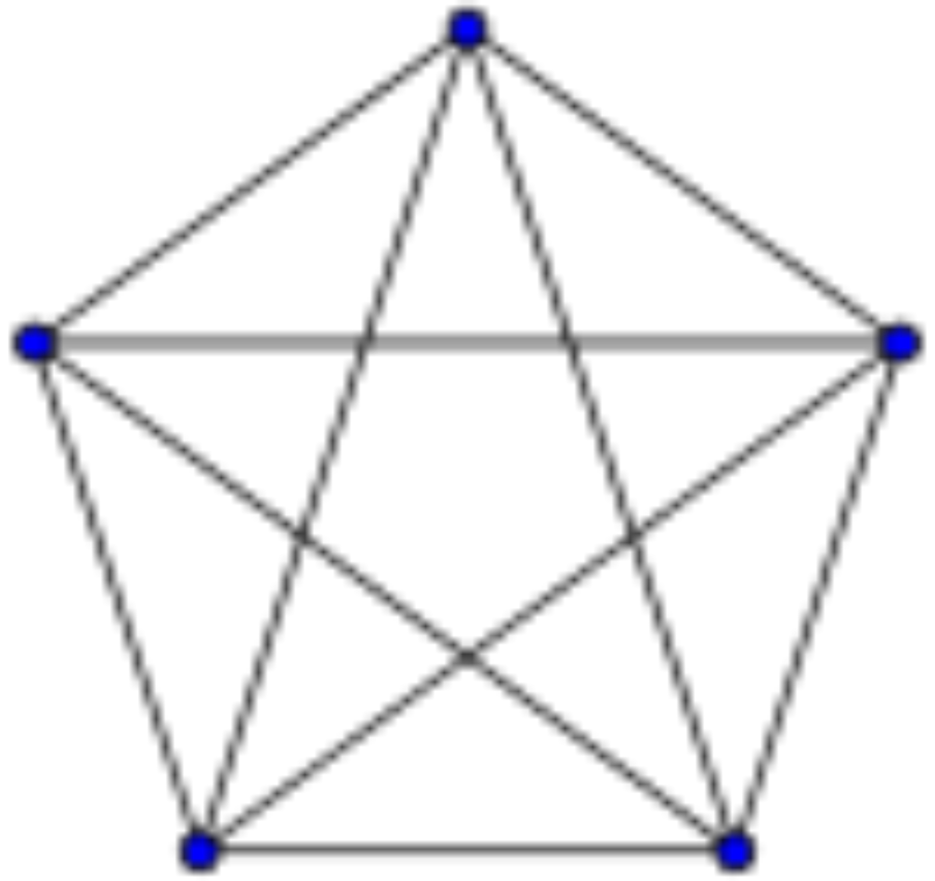
Kuratowski's theorem

A graph is planar if and only if it does not contain (a subgraph that is a subdivision of) K_5 or $K_{3,3}$

$K_{3,3}$



K_5



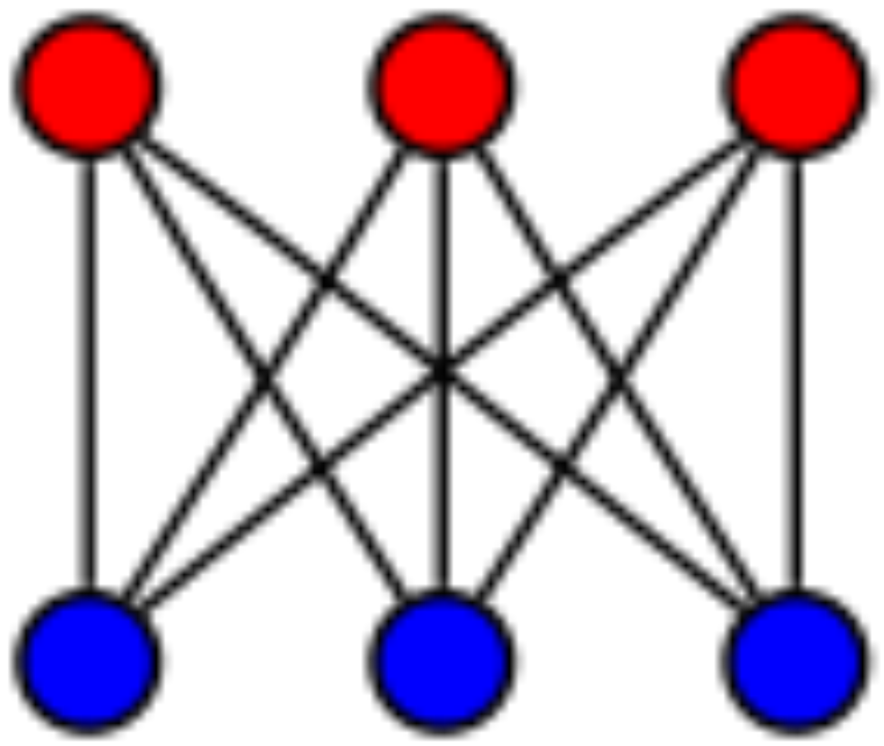
Kuratowskis's theorem gives an easy way to check planarity

Kuratowski's theorem

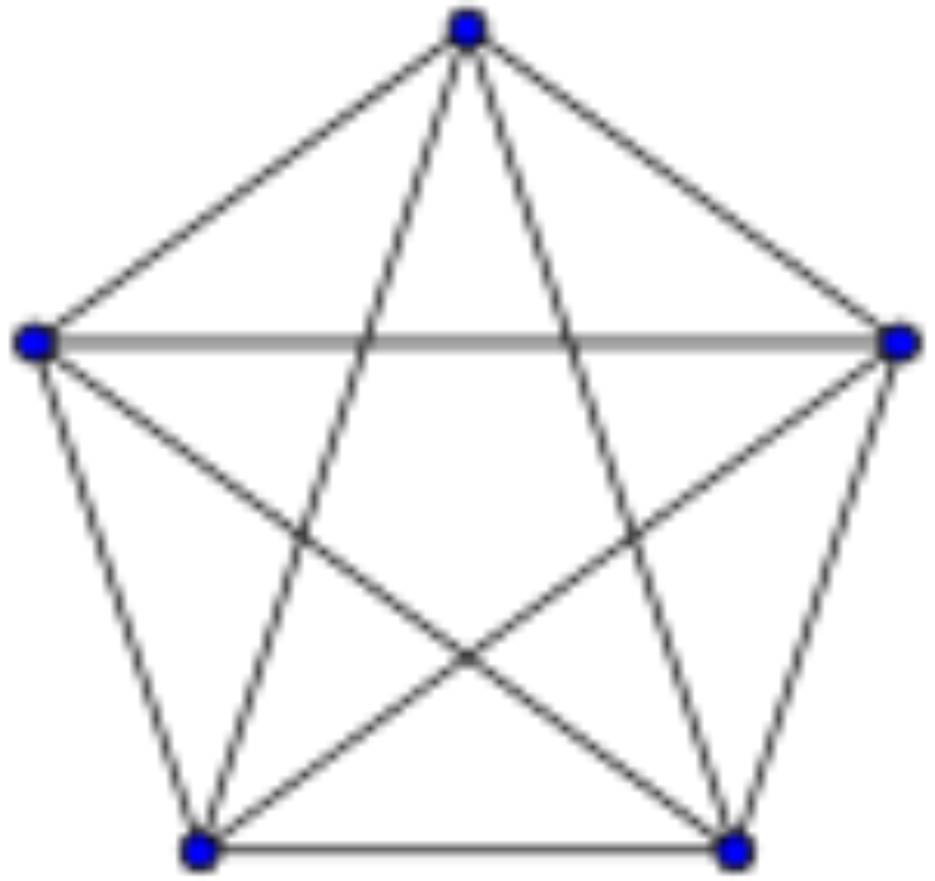
A graph is planar if and only if it does not contain (a subgraph that is a subdivision of) K_5 or $K_{3,3}$

Allows testing planarity in linear time.

$K_{3,3}$

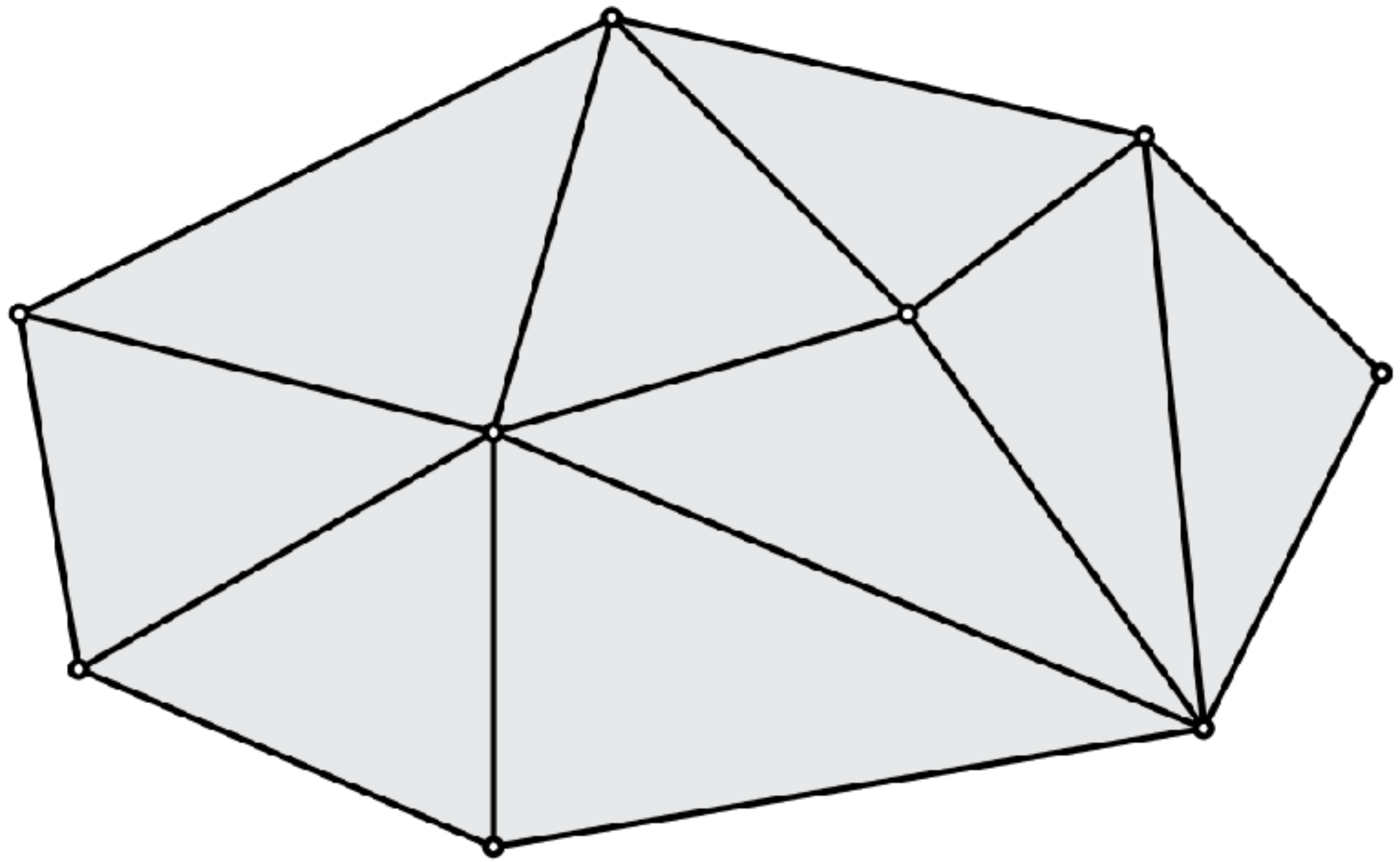


K_5



Optimal networks

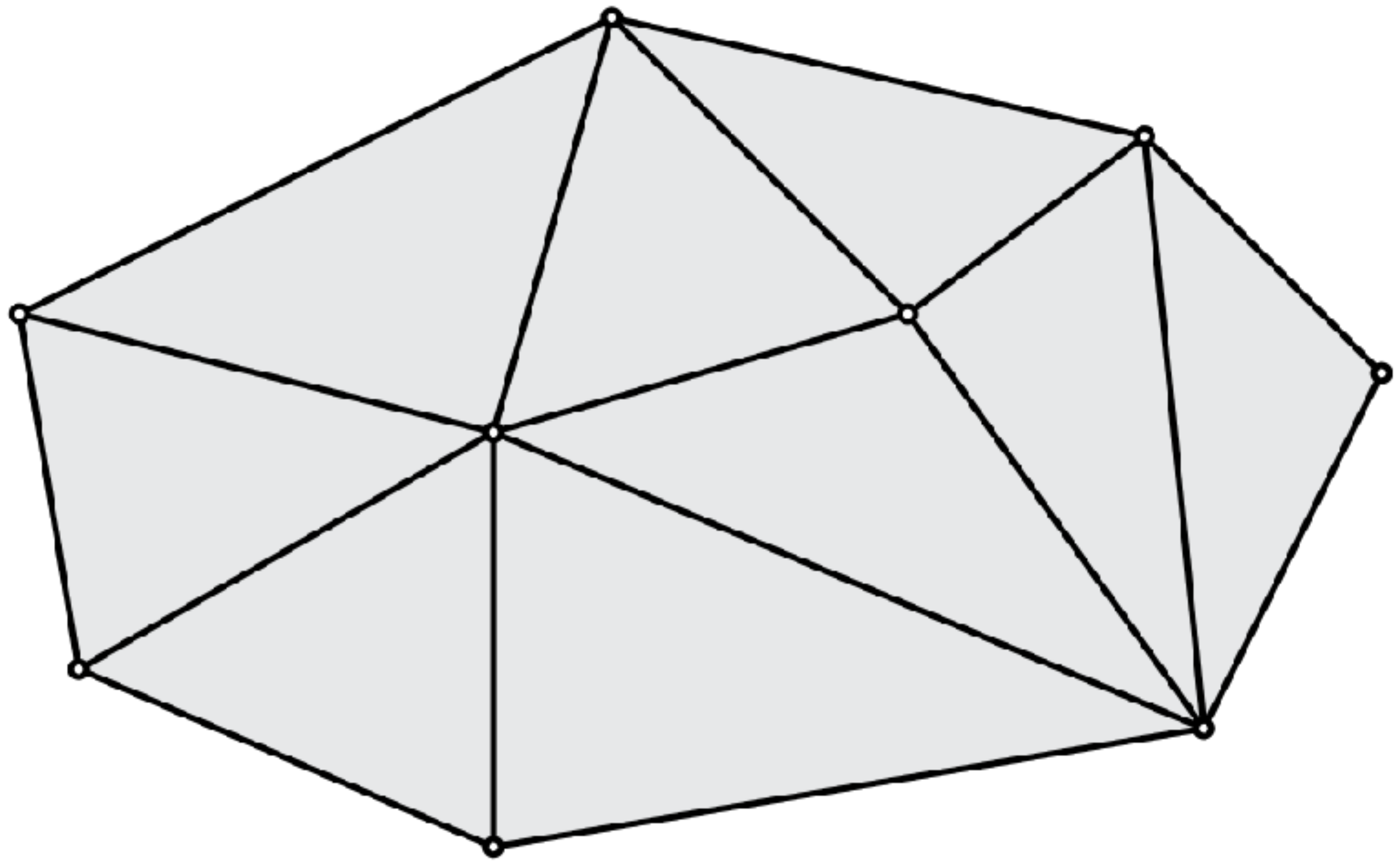
A triangulation of 2D points is a maximal set of non-crossing edges



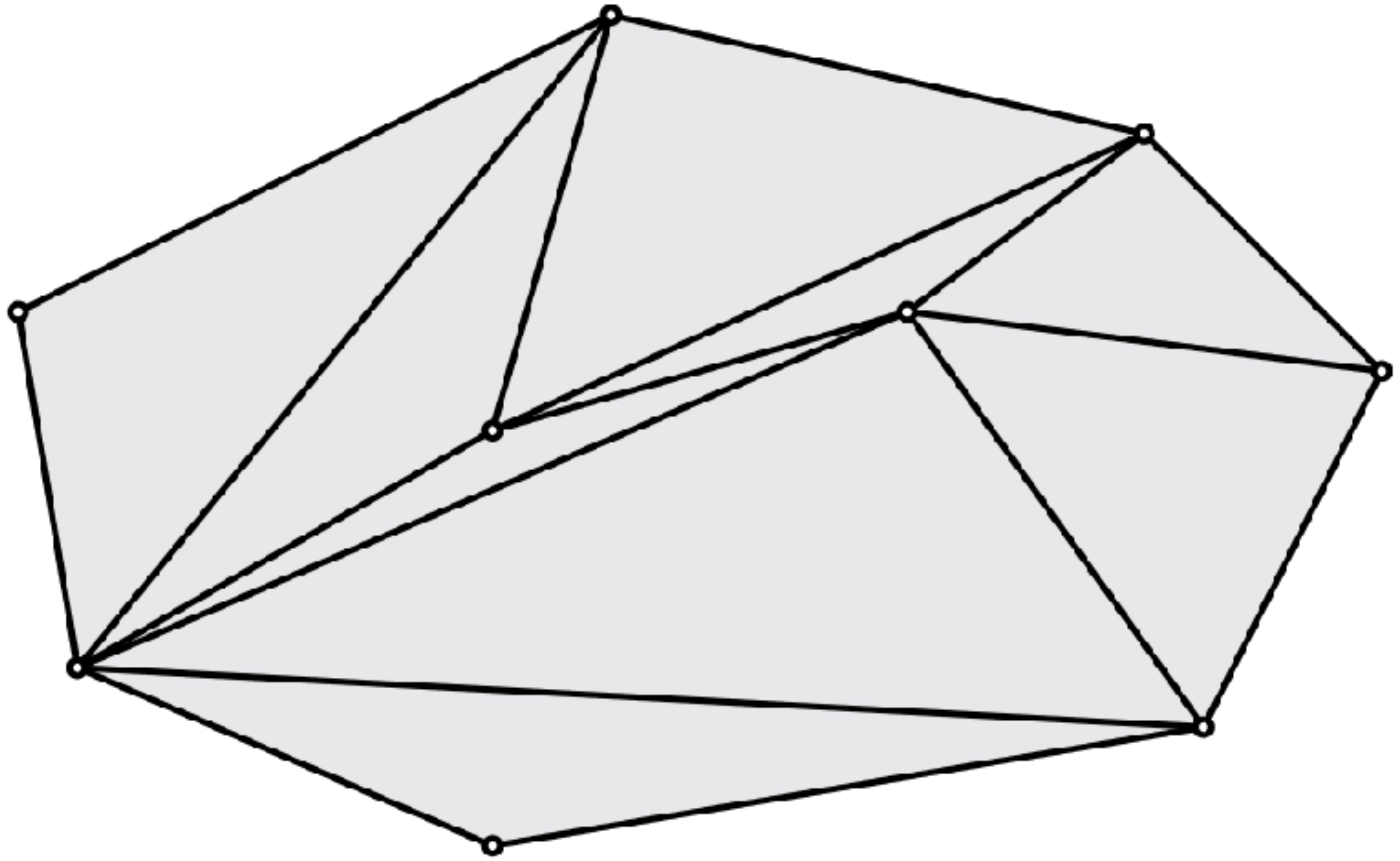
It is planar

https://en.wikipedia.org/wiki/Point-set_triangulation

A triangulation of 2D points is a maximal set of non-crossing edges



It is planar

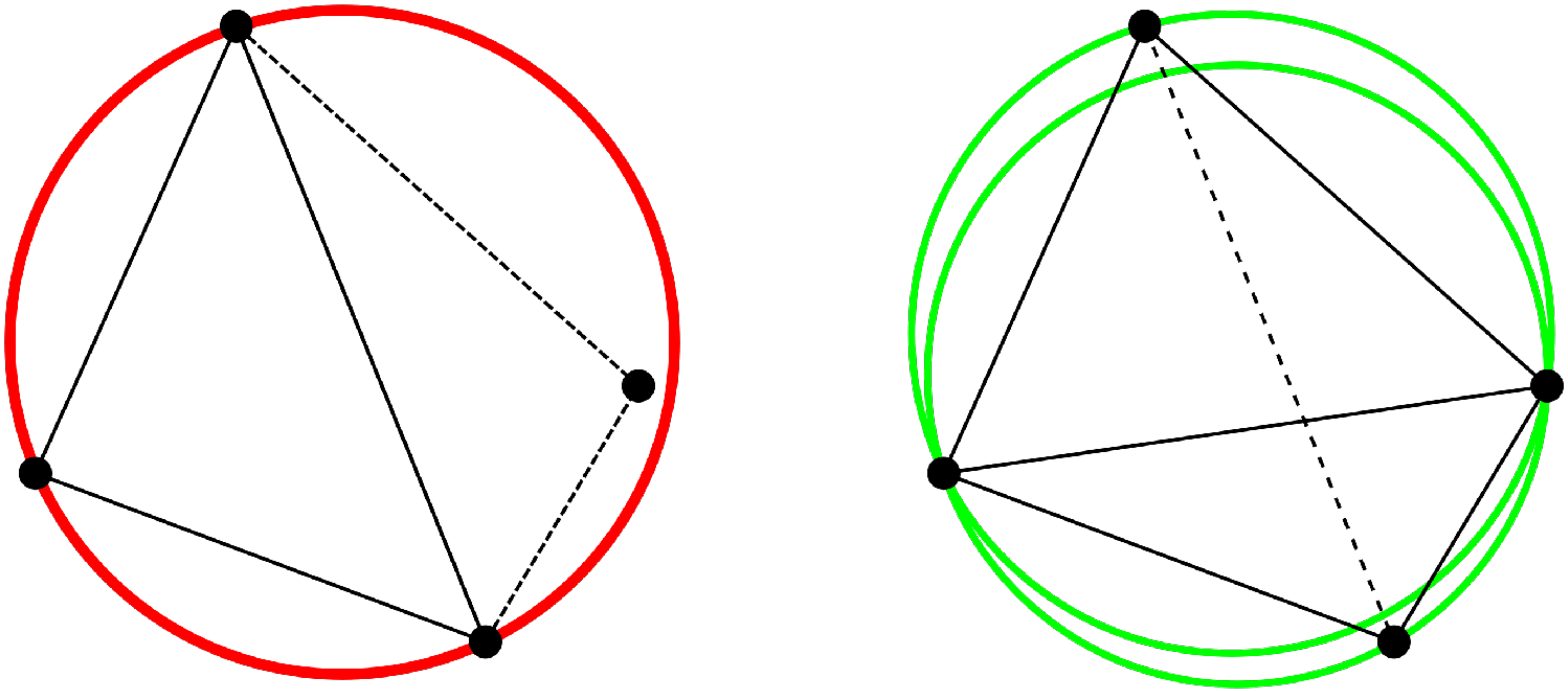


It is not unique

https://en.wikipedia.org/wiki/Point-set_triangulation

A Delaunay triangulation maximizes the minimum of all angles

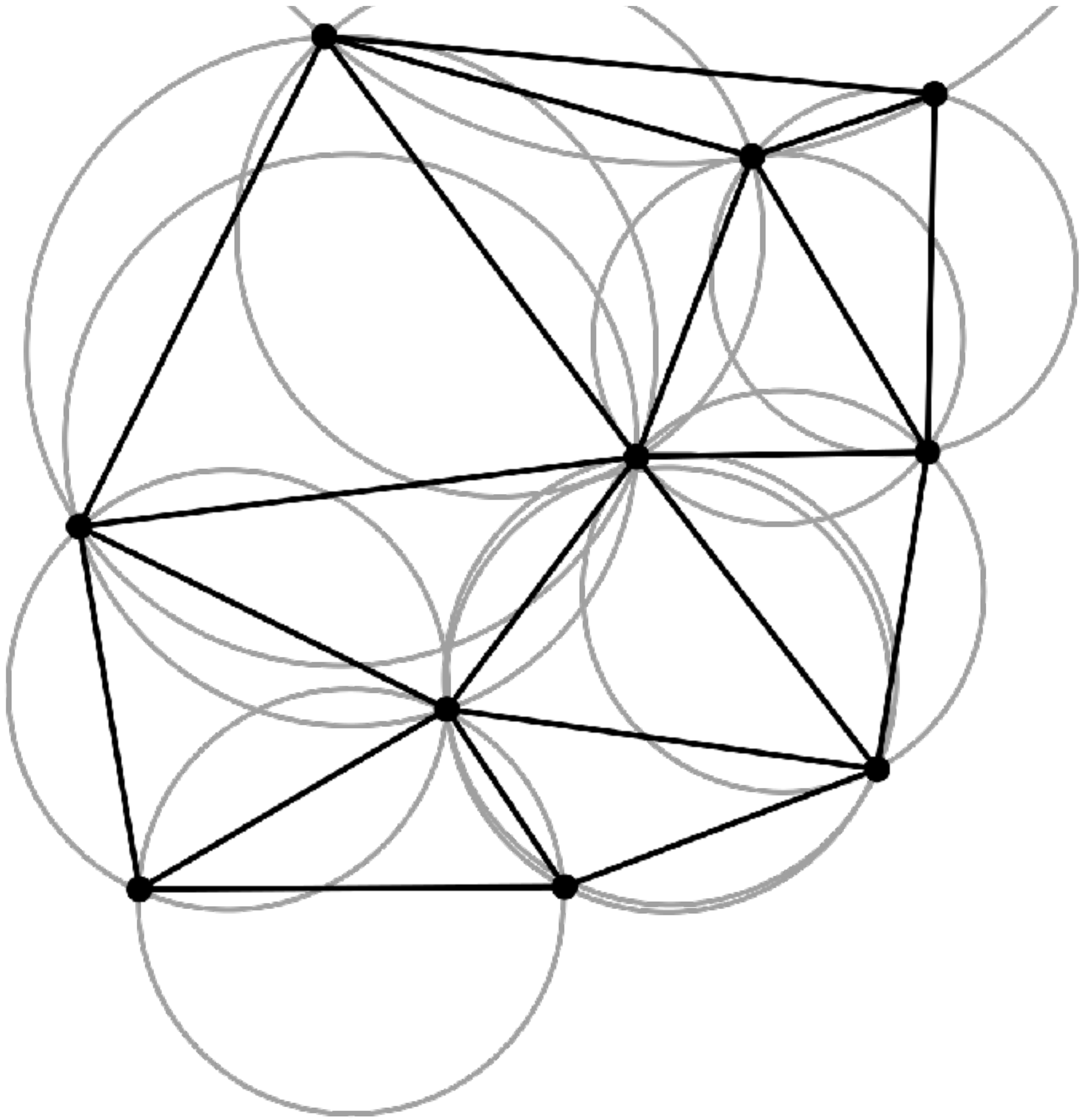
No point can be inside the circumcircle of any triangle

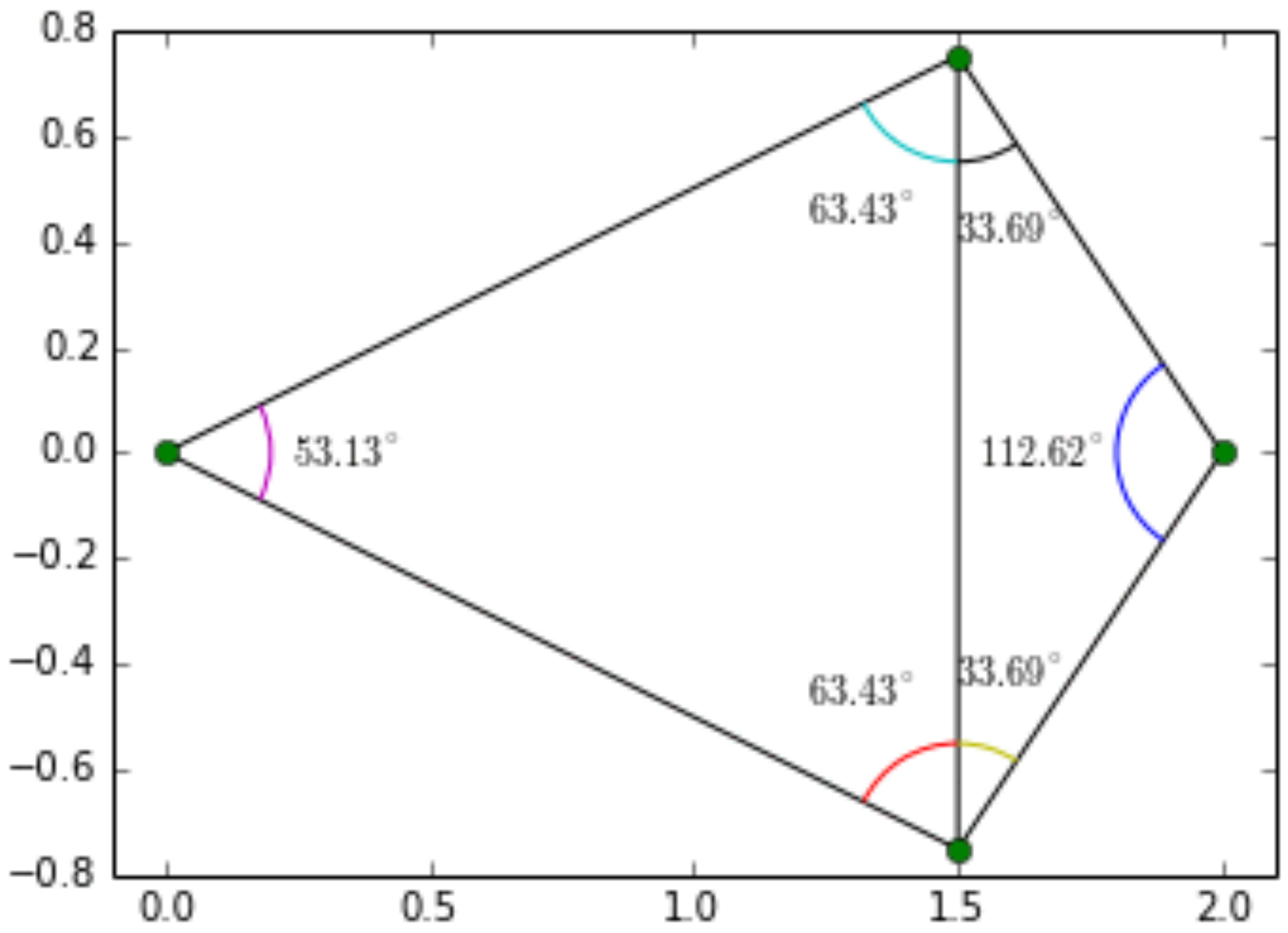


https://en.wikipedia.org/wiki/Delaunay_triangulation

A Delaunay triangulation maximizes the minimum of all angles

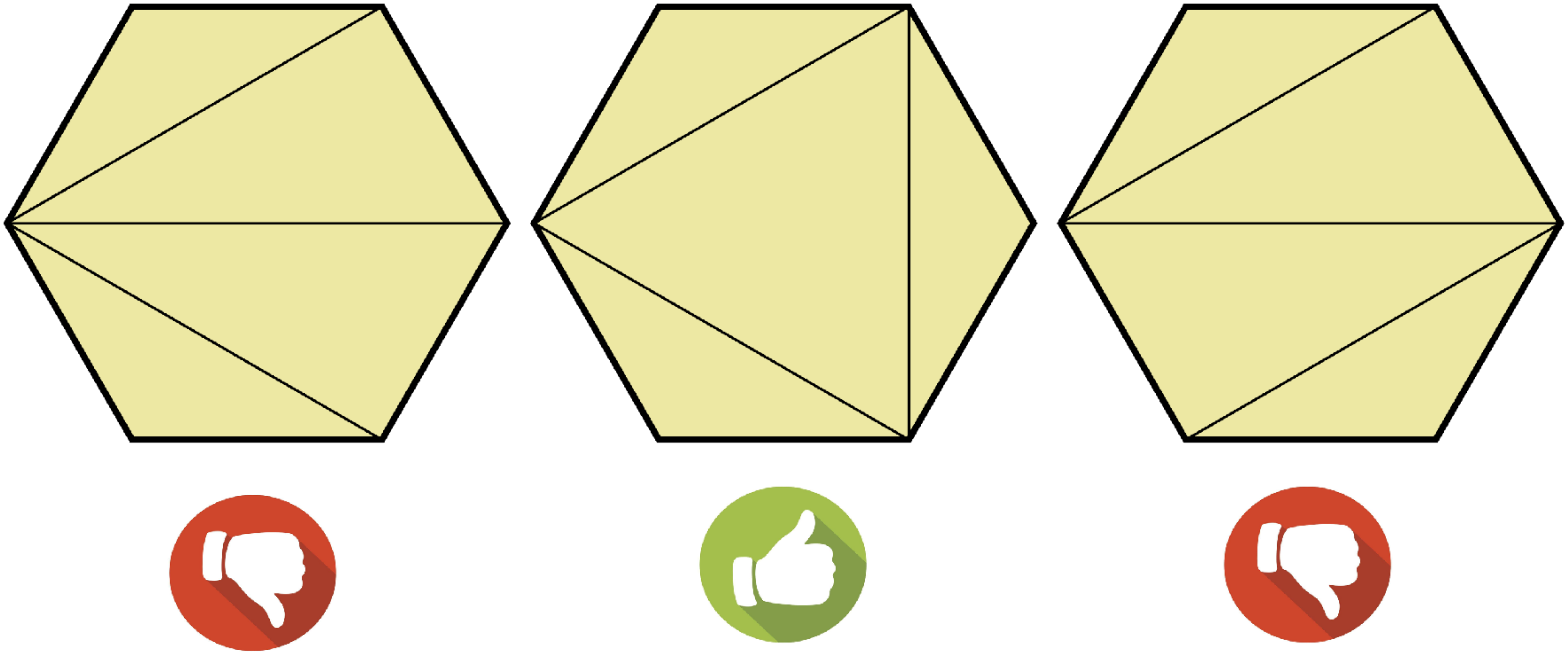
No point can be inside the circumcircle of any triangle





https://en.wikipedia.org/wiki/Delaunay_triangulation

A minimum weight triangulation minimizes sum of edge lengths

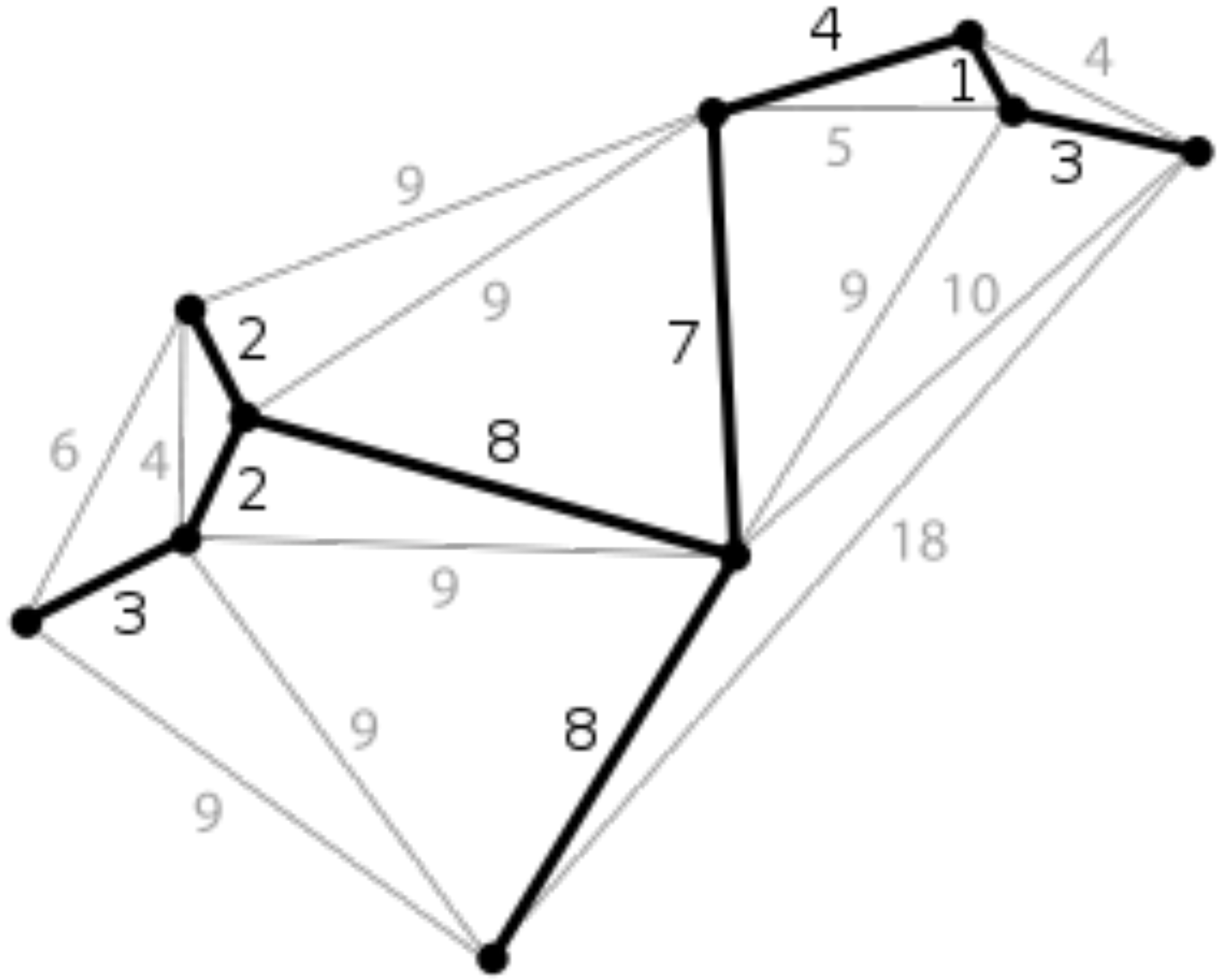


https://en.wikipedia.org/wiki/Minimum-weight_triangulation

Many other triangulations exist

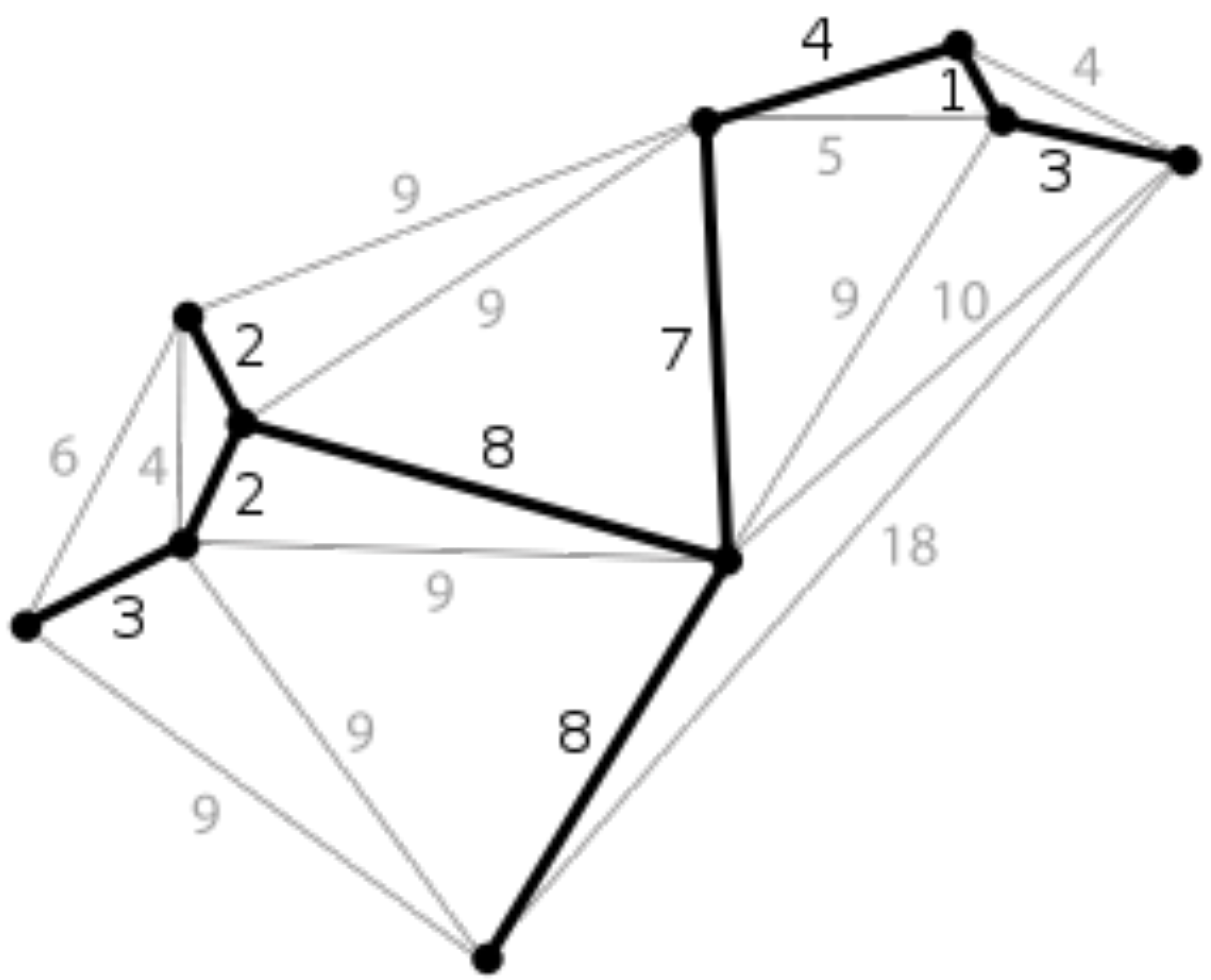
		minimize	maximize
minimum	angle		$O(n \log n)$ (Delaunay triangulation)
maximum	angle	$O(n^2 \log n)$ [8] [9]	
minimum	area	$O(n^2)$ [10]	$O(n^2 \log n)$ [11]
maximum	area	$O(n^2 \log n)$ [11]	
maximum	degree	NP-complete for degree of 7 [12]	
maximum	eccentricity	$O(n^3)$ [9]	
minimum	edge length	$O(n \log n)$ (Closest pair of points problem)	NP-complete [13]
maximum	edge length	$O(n^2)$ [14]	$O(n \log n)$ (using the Convex hull)
sum of	edge length	NP-hard (Minimum-weight triangulation)	
minimum	height		$O(n^2 \log n)$ [9]
maximum	slope	$O(n^3)$ [9]	

https://en.wikipedia.org/wiki/Point-set_triangulation



Given a connected, undirected, weighted graph, a spanning tree of the graph is a subgraph that is a tree and connects all the nodes.

The MST is a subgraph that connects all nodes optimally

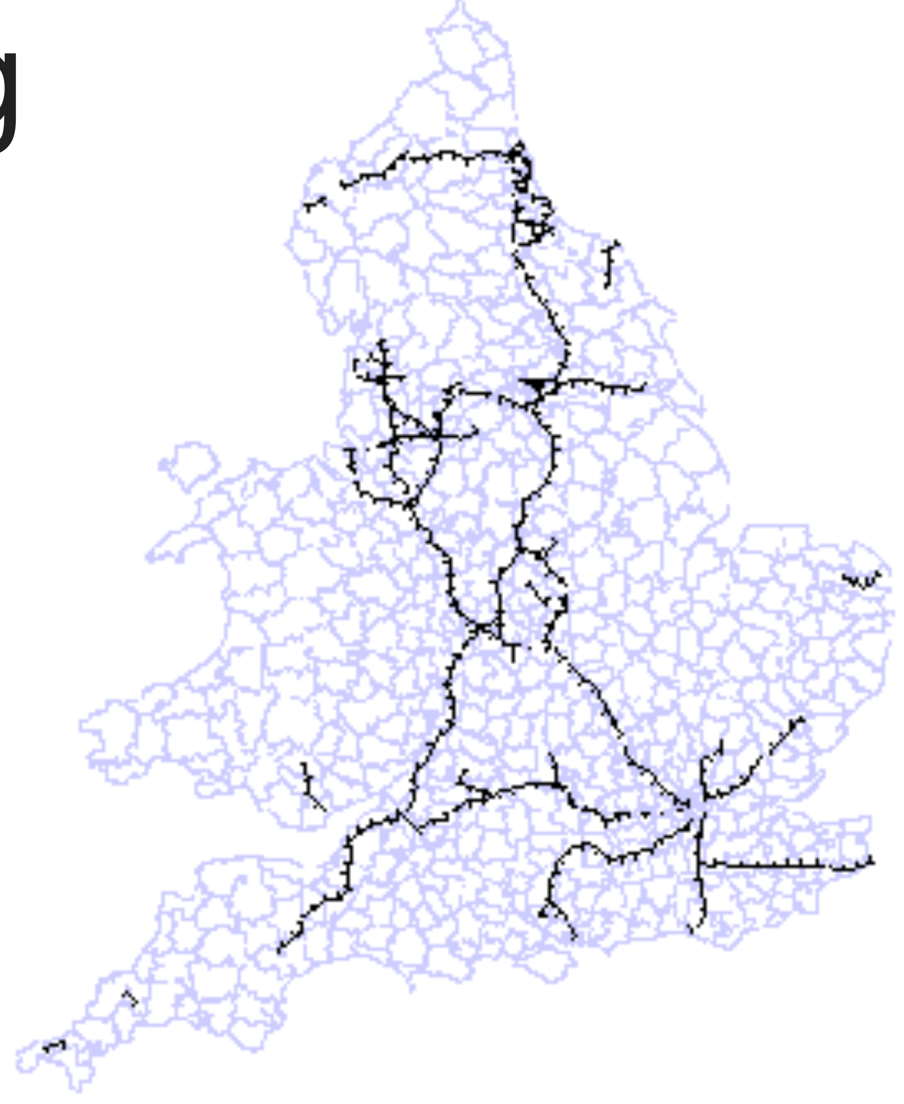


Given a connected, undirected, weighted graph, a spanning tree of the graph is a subgraph that is a tree and connects all the nodes.

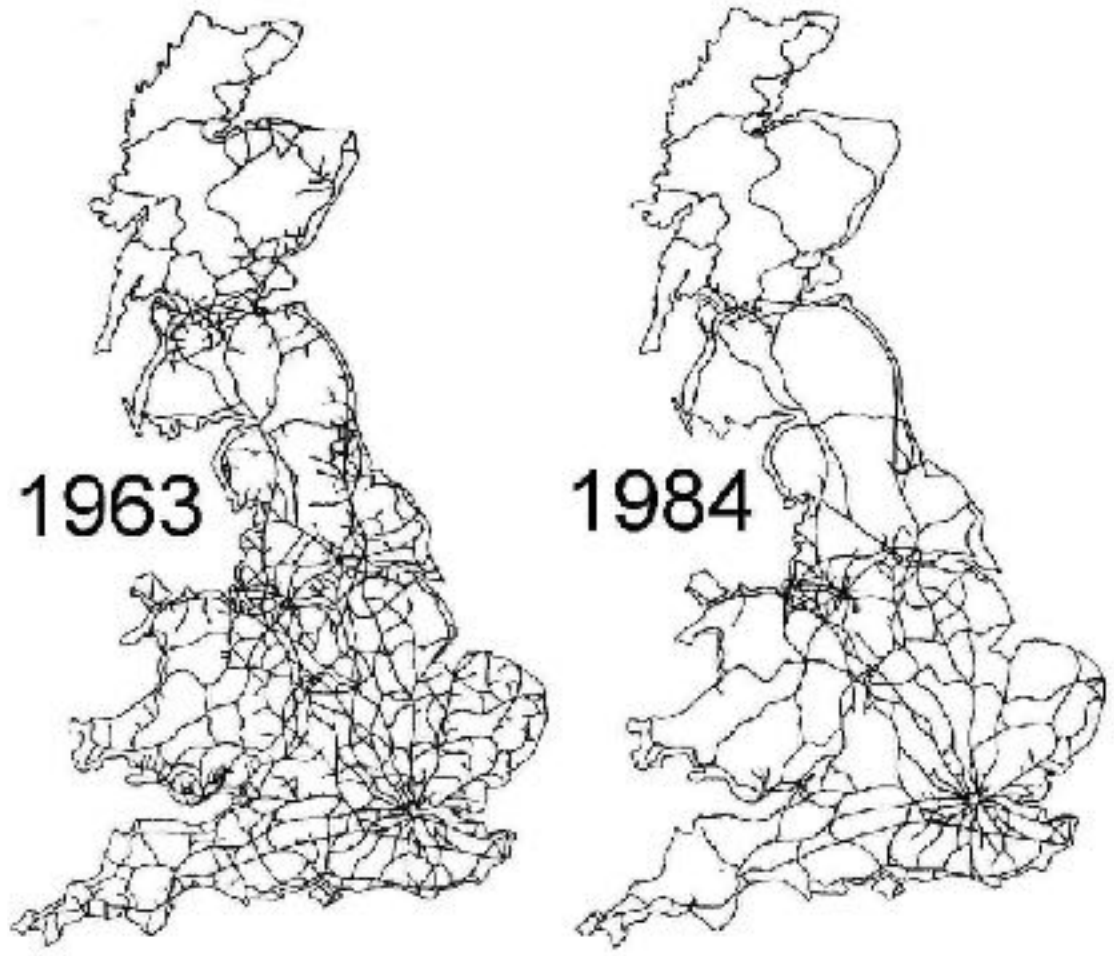
A minimum spanning tree (MST) is a spanning tree such that the sum of its link lengths is smaller than or equal to the sum of the link lengths of any other spanning tree.

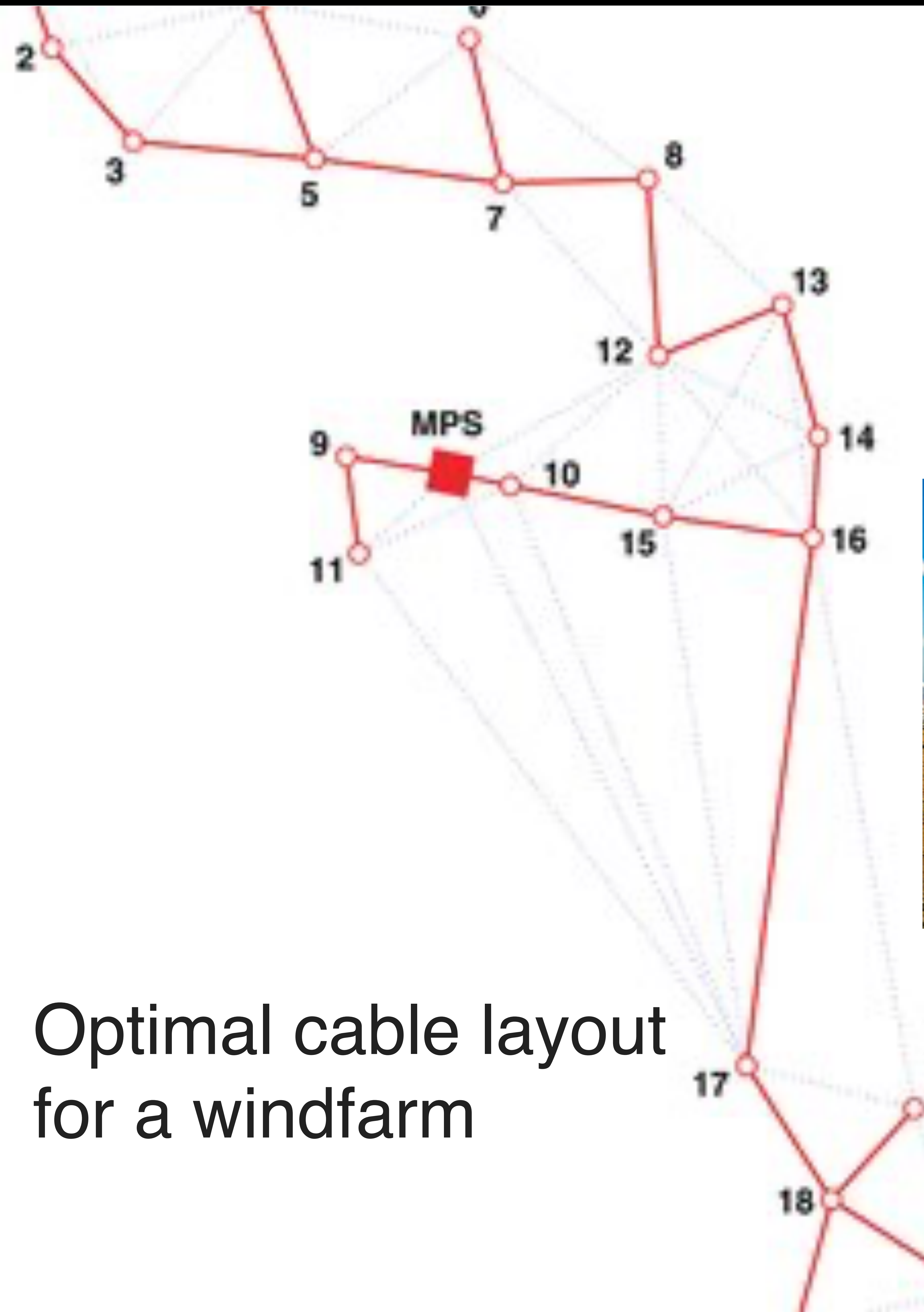
The MST has many applications for spatial networks

Railroad network planning



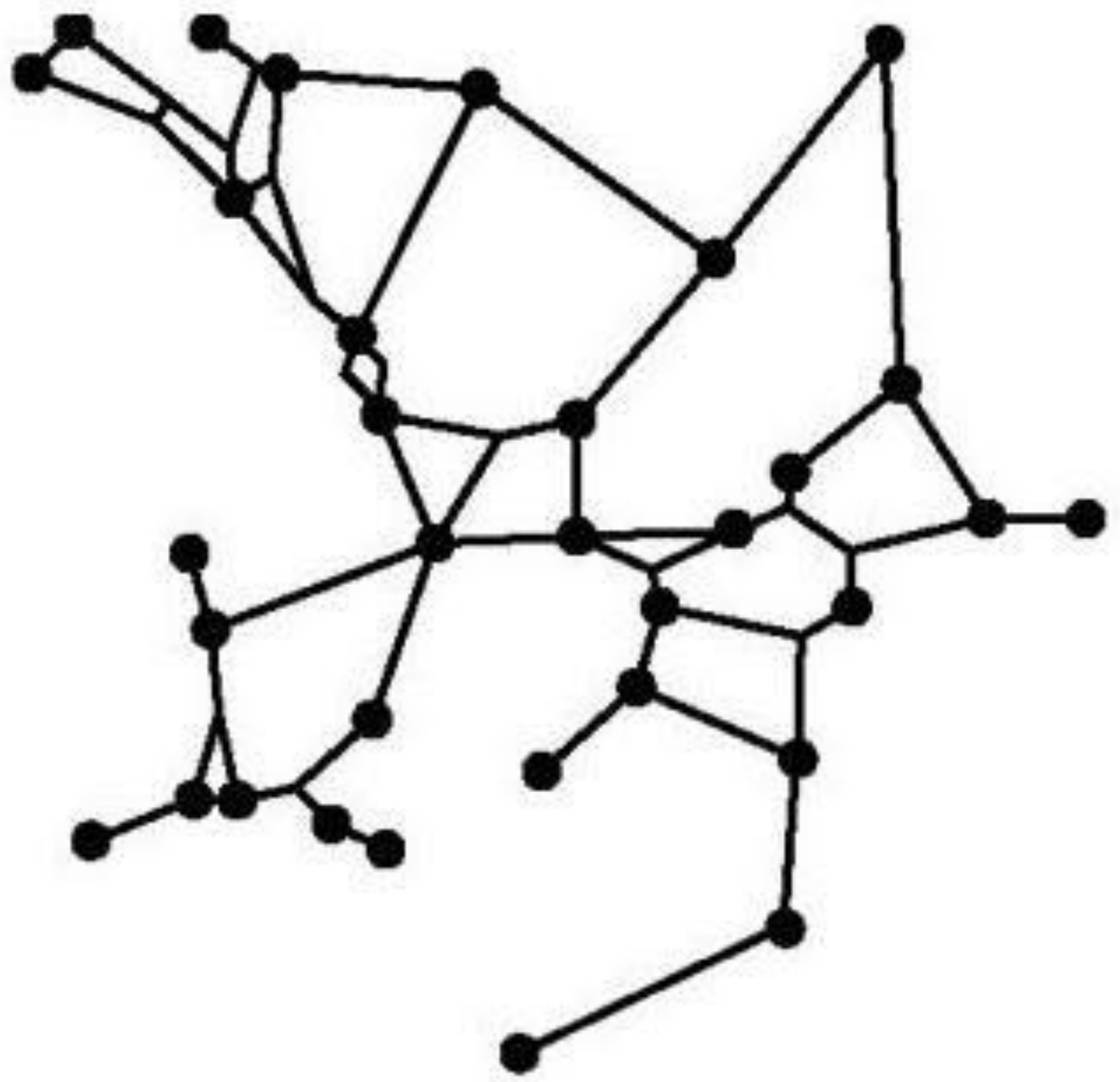
1845



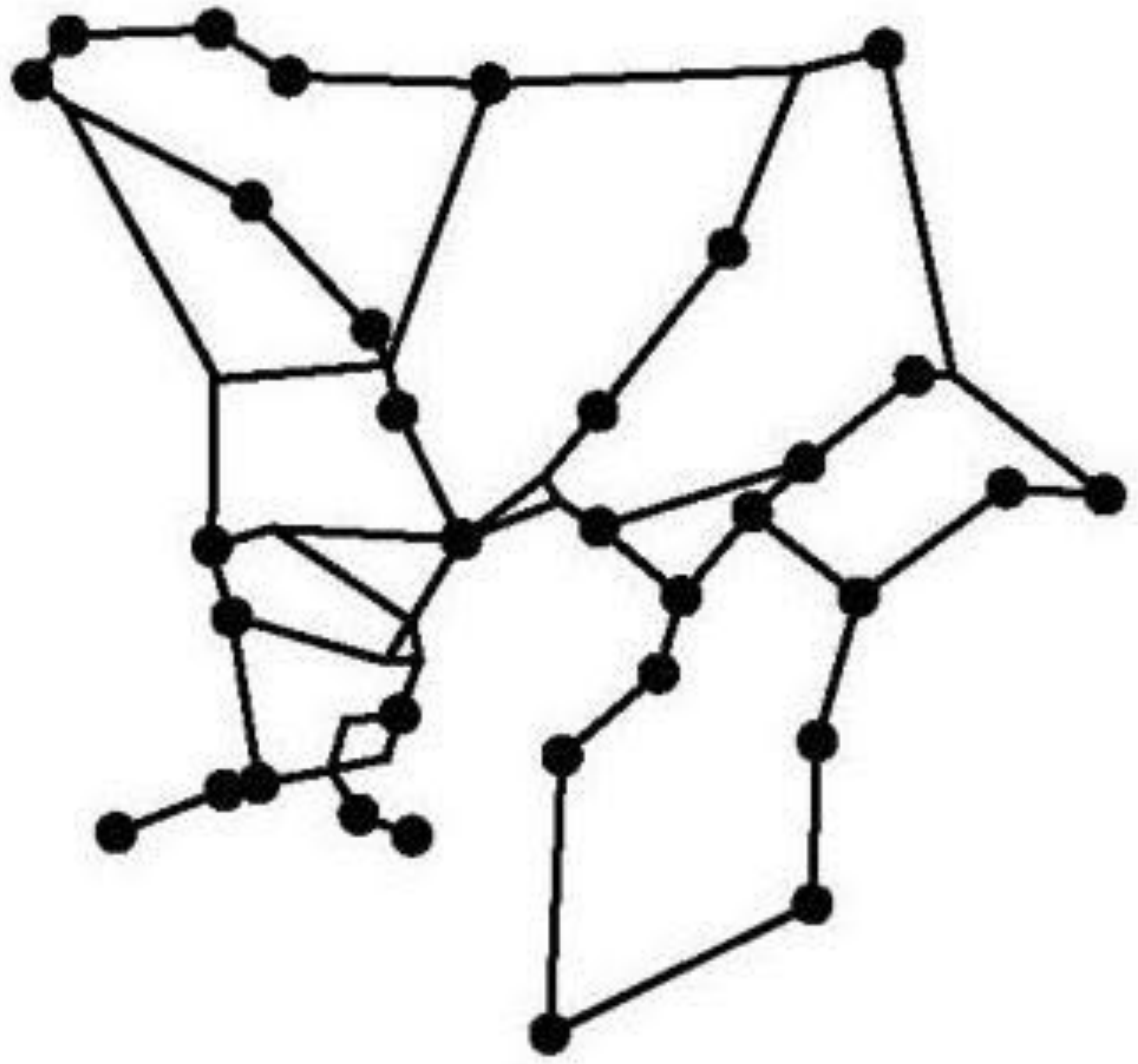


Optimal cable layout for a windfarm

Slime mold network

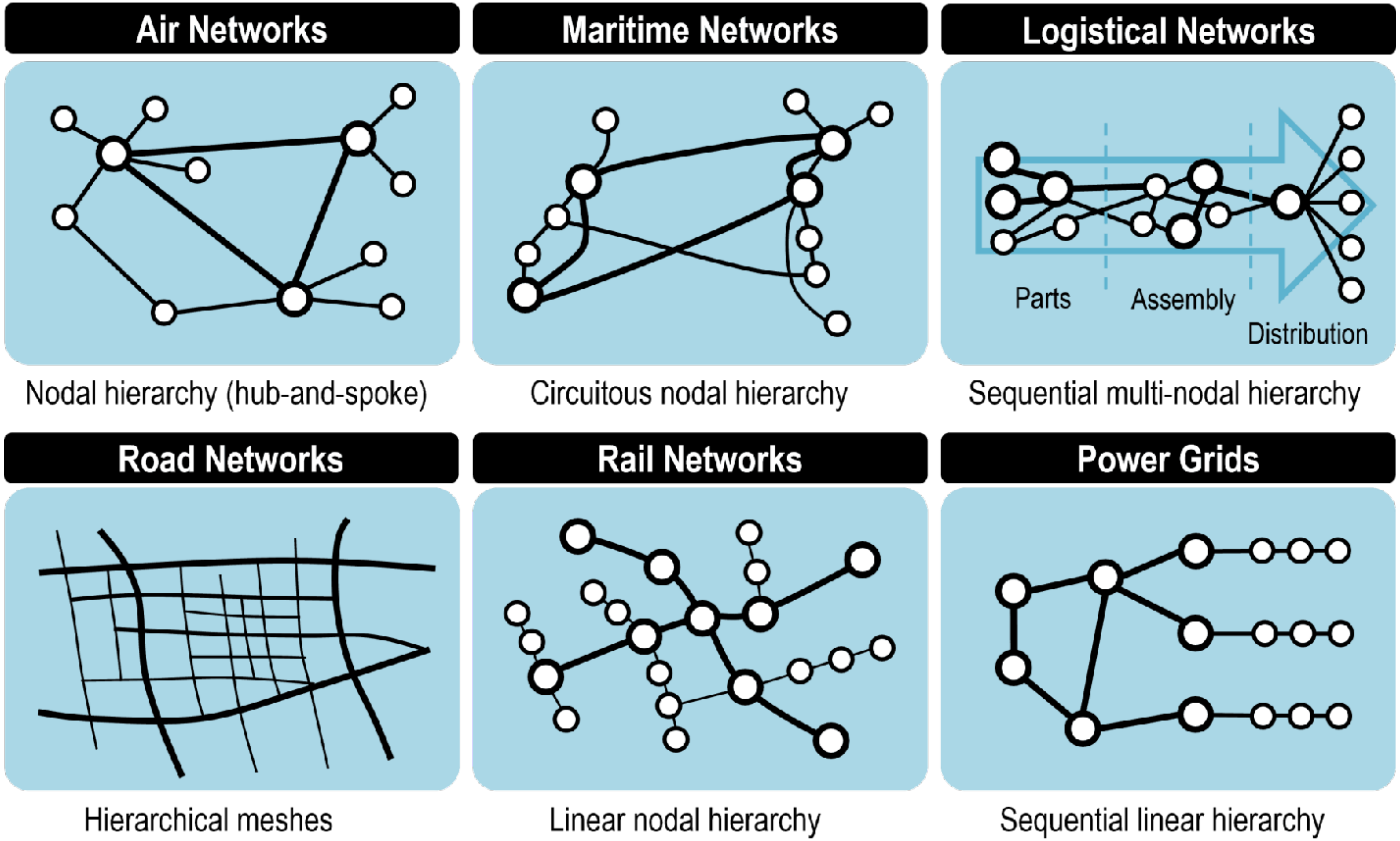


Actual Tokyo railway network



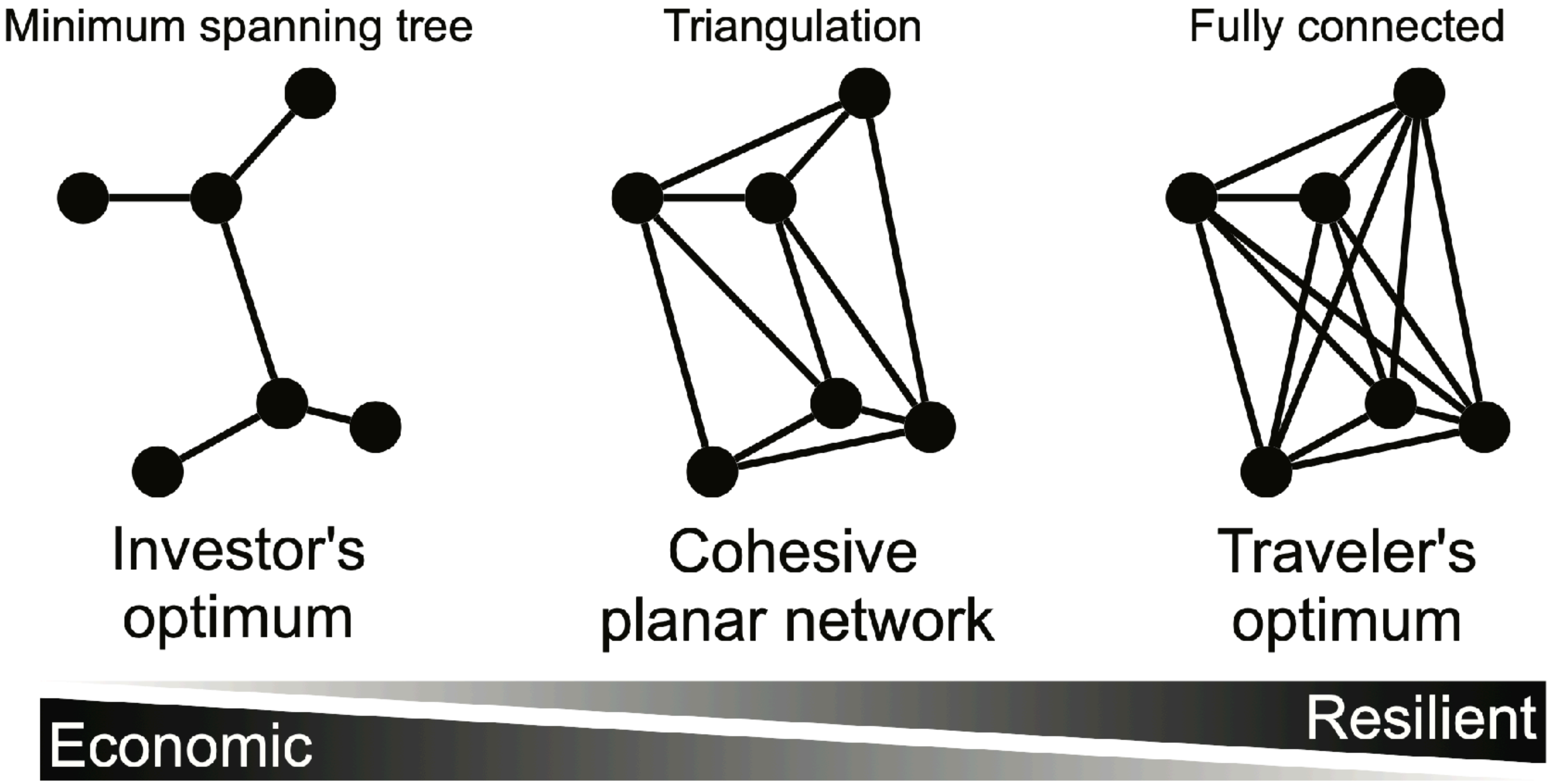
Tero et al. 2010

Spatial networks have different shapes and vulnerabilities



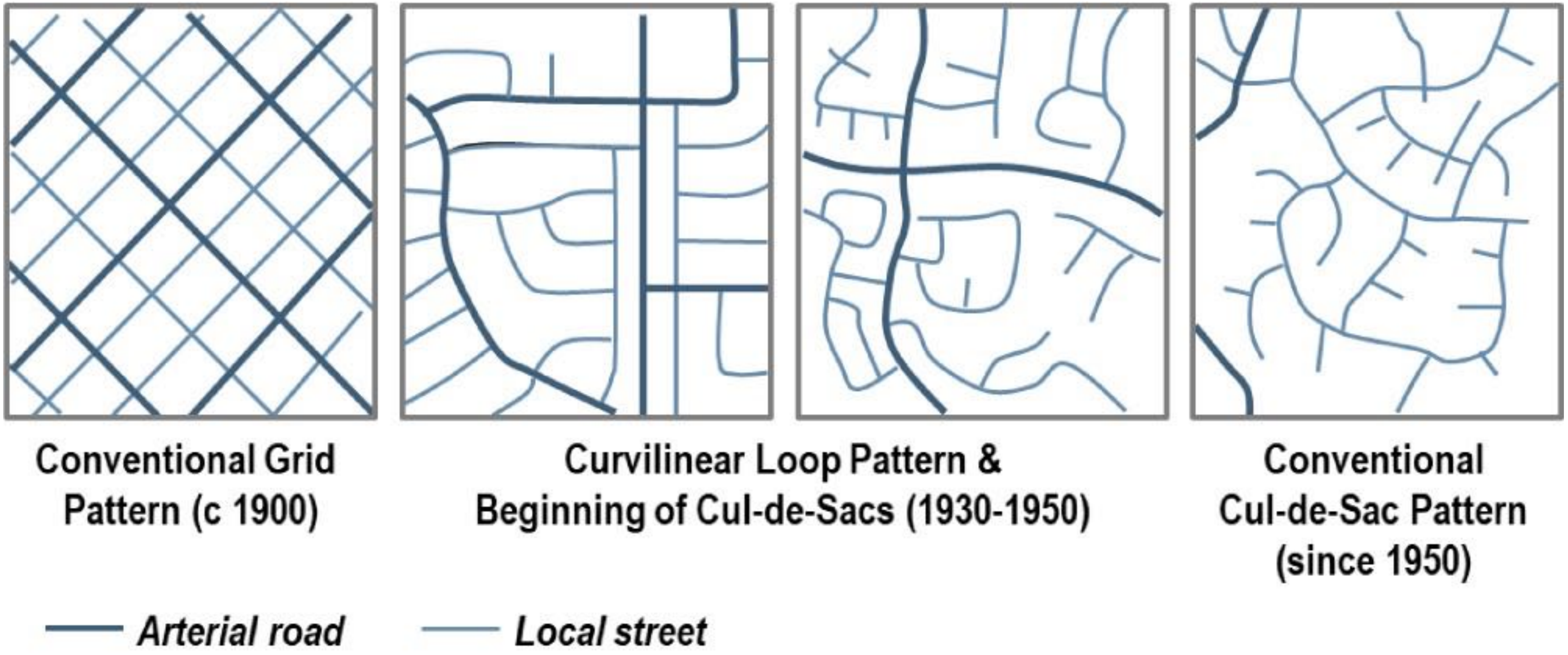
https://transportgeography.org/contents/chapter9/transportation-and-disasters/transportation-networks-vulnerabilities/Attribution

There is a tradeoff between economic investment and resilience



Street networks

Street networks can look very different



https://transportgeography.org/contents/chapter8/transportation-urban-form/street-network-types/

Spatial network measures can quantify these differences

Detour index: Straight distance versus network distance

$$DI = \frac{D(S)}{D(T)}$$

Also: circuity, directness, stretch

https://transportgeography.org/contents/methods/graph-theory-measures-indices/

Spatial network measures can quantify these differences

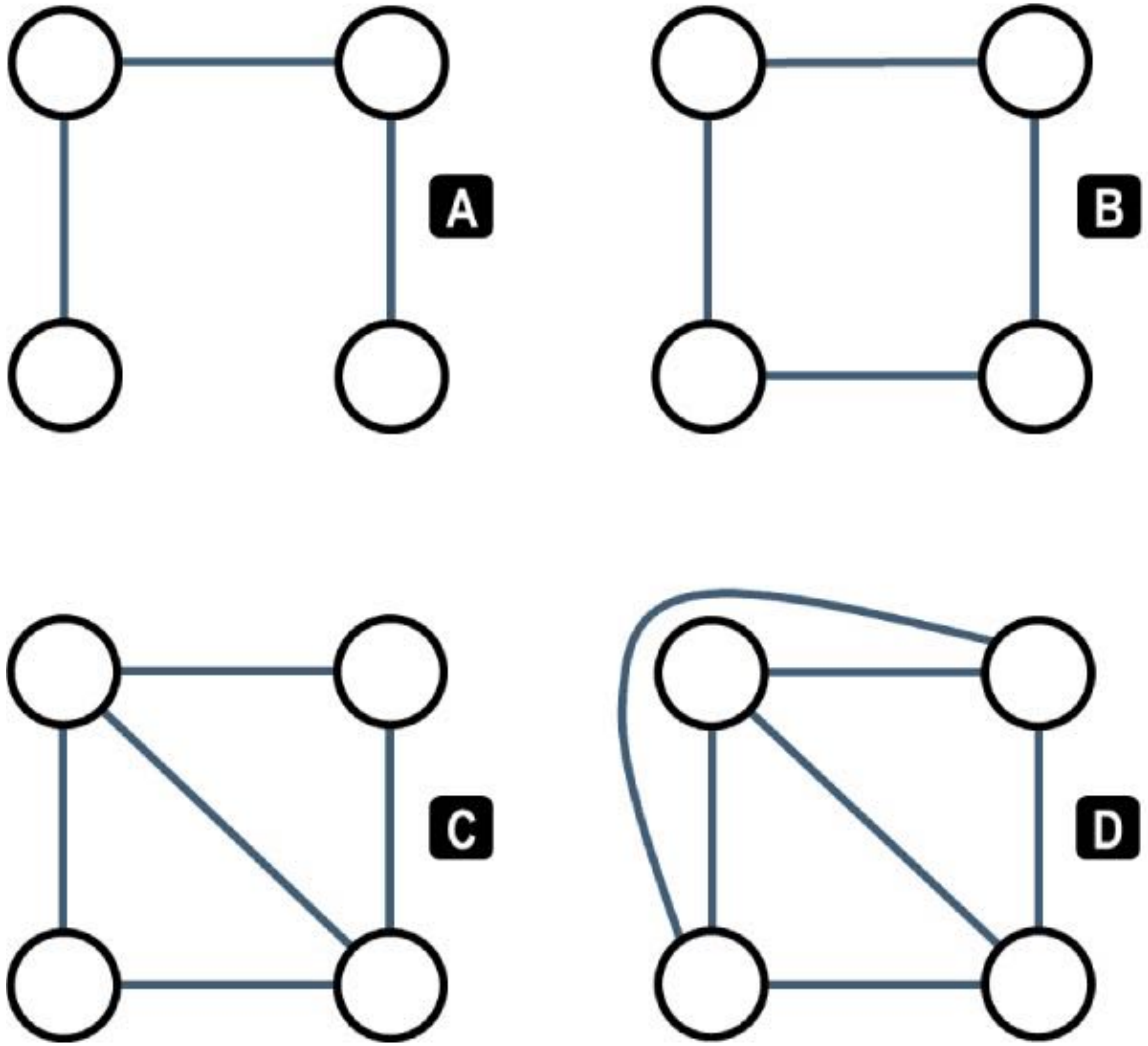
Network density: Links per surface area

$$ND = \frac{L}{S}$$

https://transportgeography.org/contents/methods/graph-theory-measures-indices/

Spatial network measures can quantify these differences

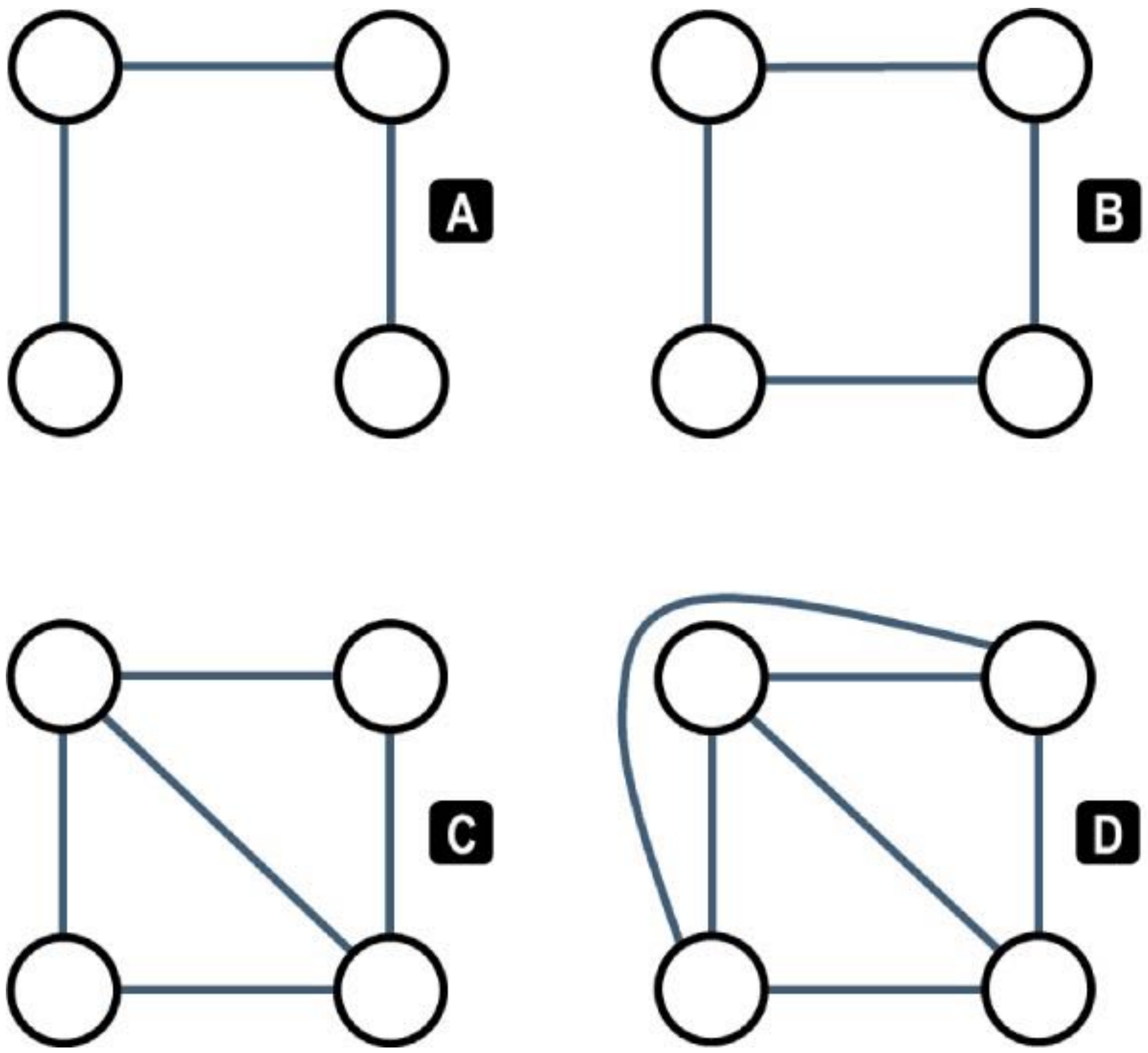
Alpha index: Existing versus possible elementary cycles



https://transportgeography.org/contents/methods/graph-theory-measures-indices/alpha-index-graph/

Spatial network measures can quantify these differences

Alpha index: Existing versus possible elementary cycles



$$\alpha = \frac{u}{2v - 5}$$

	u (e-v+p)	2v-5	Alpha
A	0	3	0.0
B	1	3	0.33
C	2	3	0.66
D	3	3	1.0

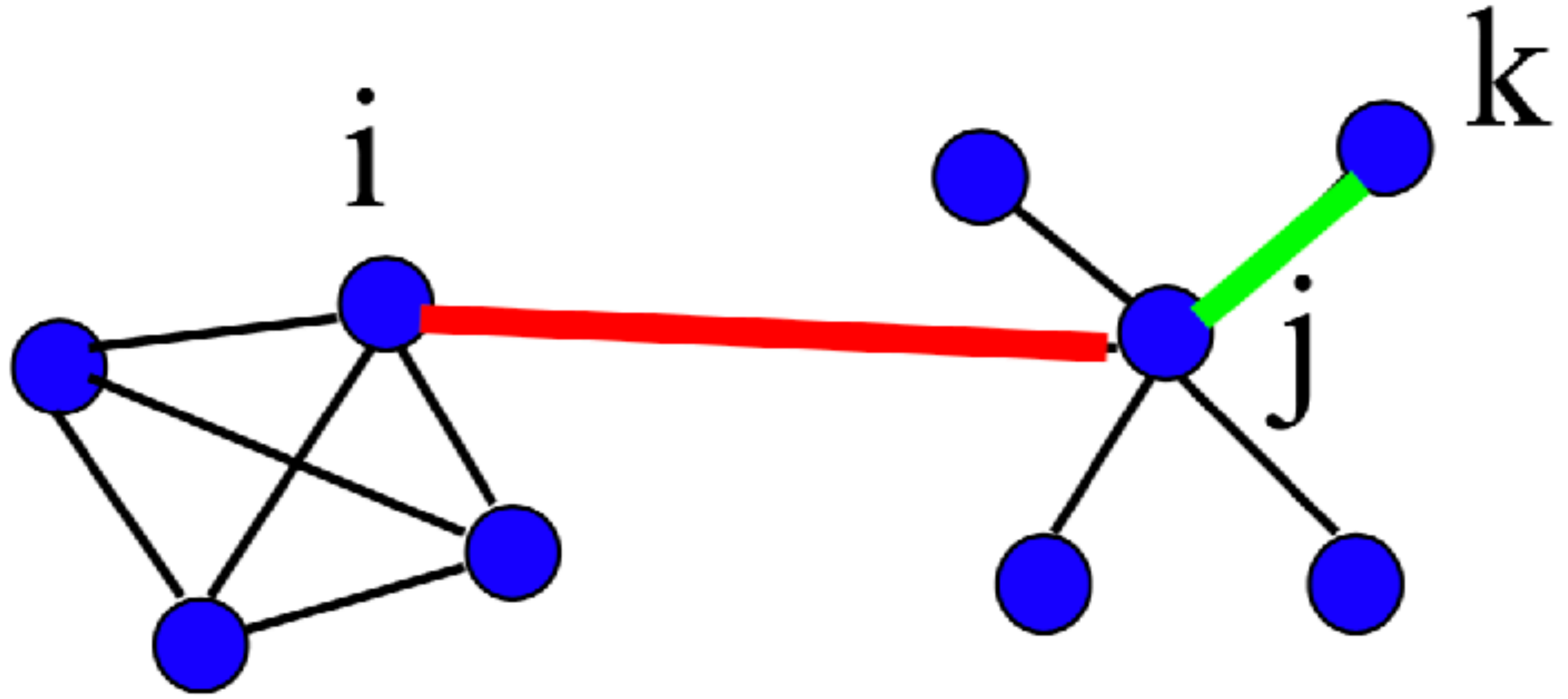
Tree

Max. planar

https://transportgeography.org/contents/methods/graph-theory-measures-indices/alpha-index-graph/

Betweenness centrality measures flow

Freeman (1977): Betweenness centrality measures how many times a link is part of all shortest paths in the network.



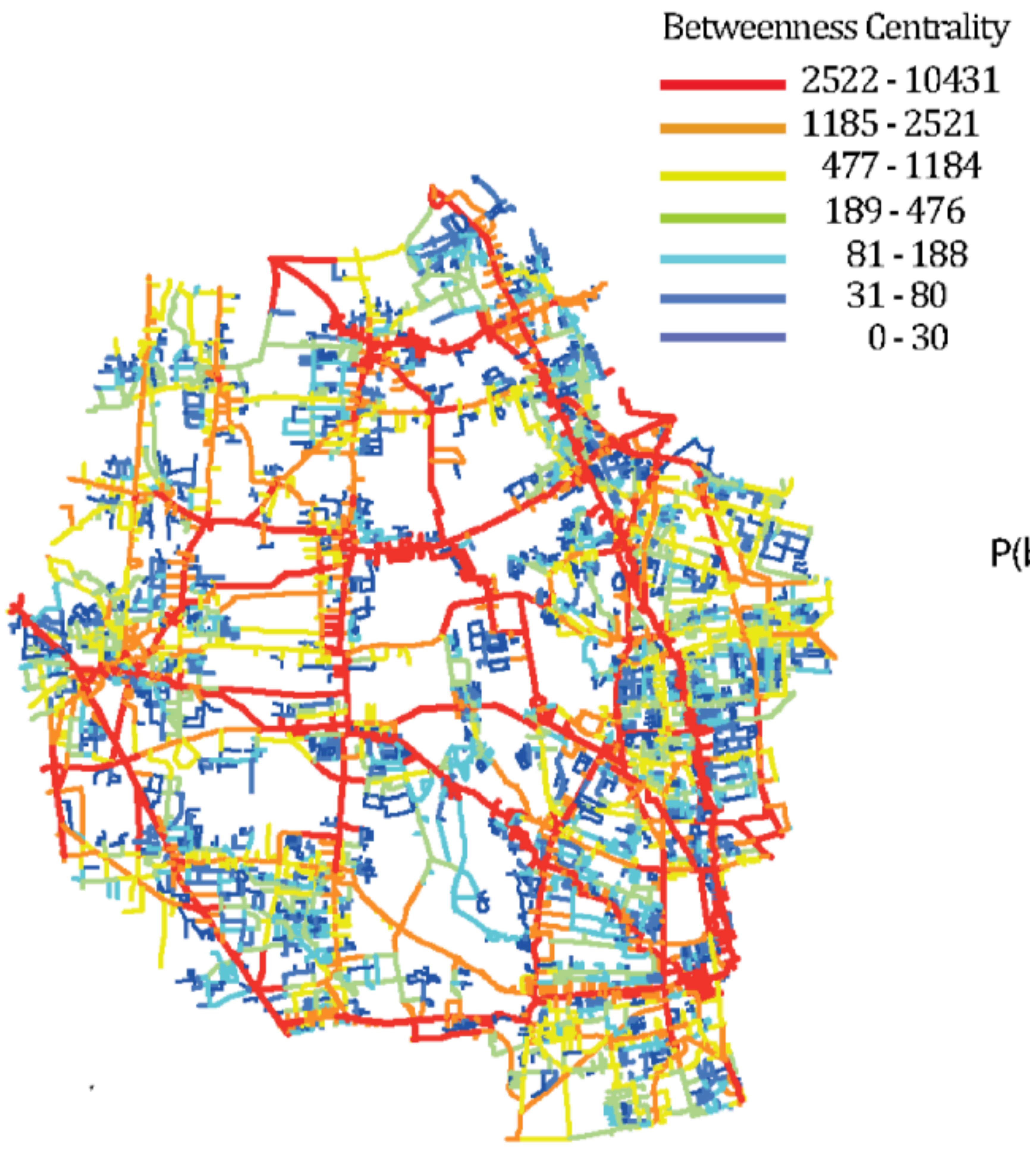
ij: large centrality

jk: small centrality

$$g(ij) = \sum_{s,t} \frac{\sigma_{st}(ij)}{\sigma_{st}}$$

Betweenness centrality measures flow

Freeman (1977): Betweenness centrality measures how many times a link is part of all shortest paths in the network.



Measures the importance of a segment in the shortest paths flow

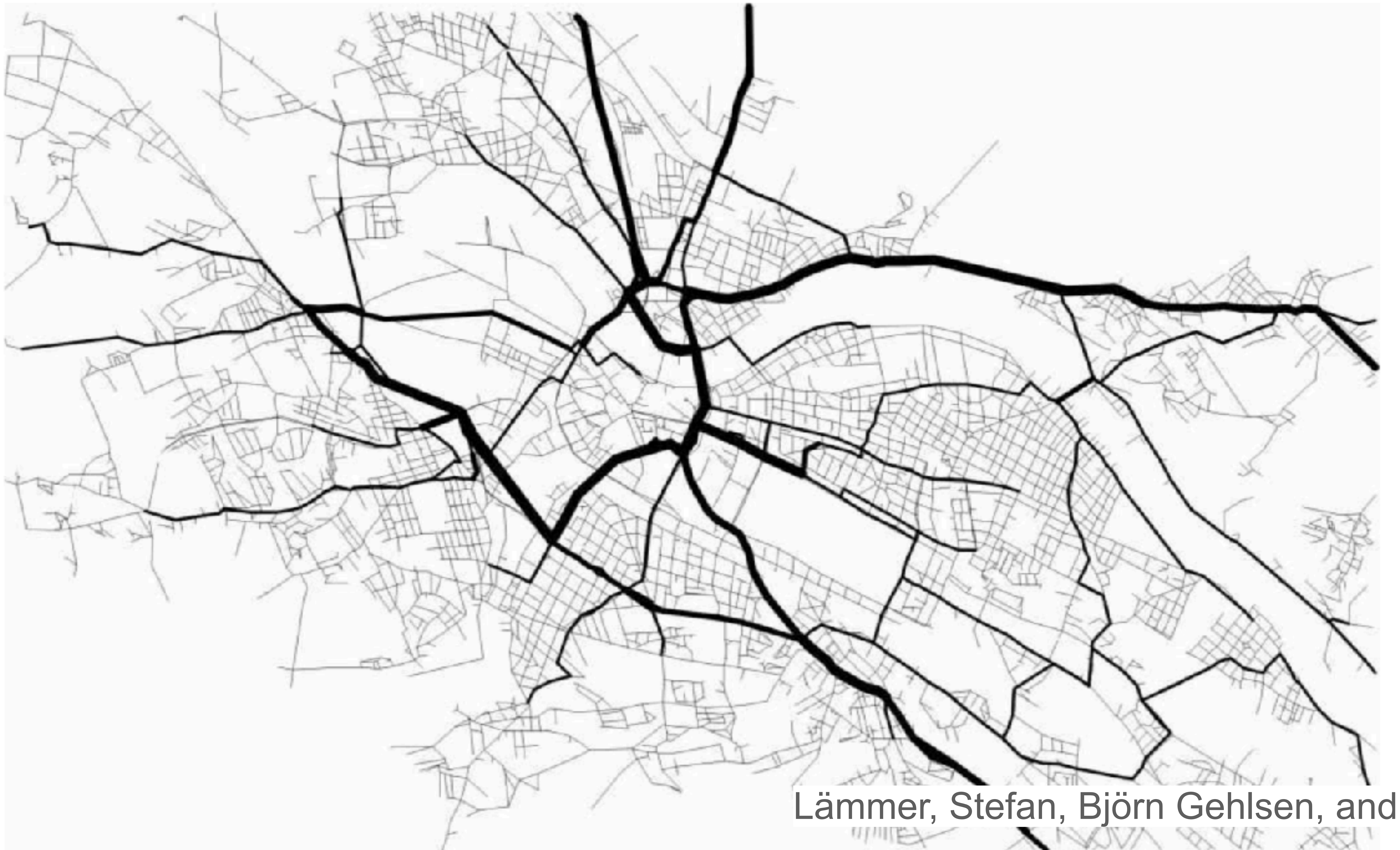
Gives the backbone of stable central roads

Can be broadly distributed

Betweenness centrality measures flow

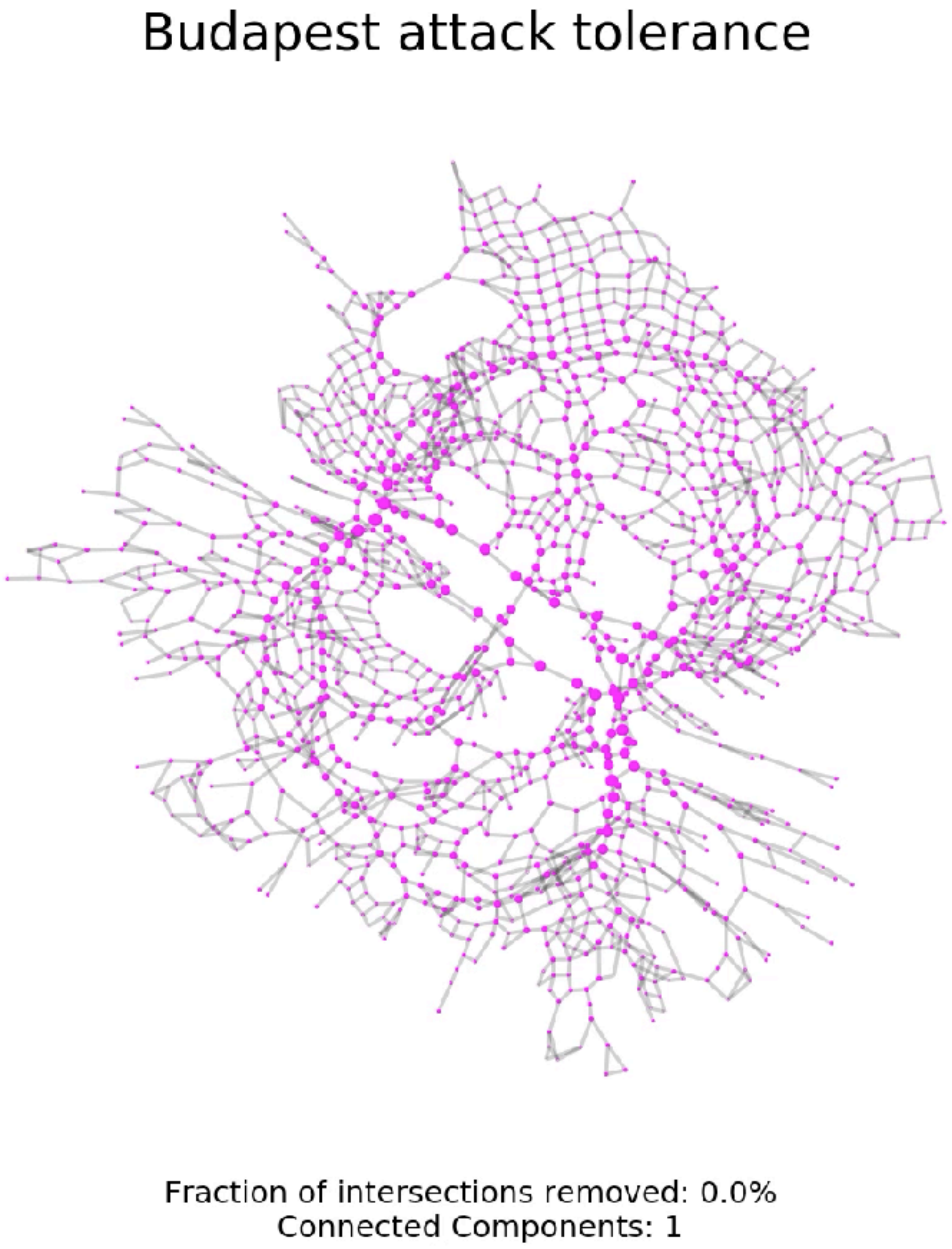
Freeman (1977): Betweenness centrality measures how many times a link is part of all shortest paths in the network.

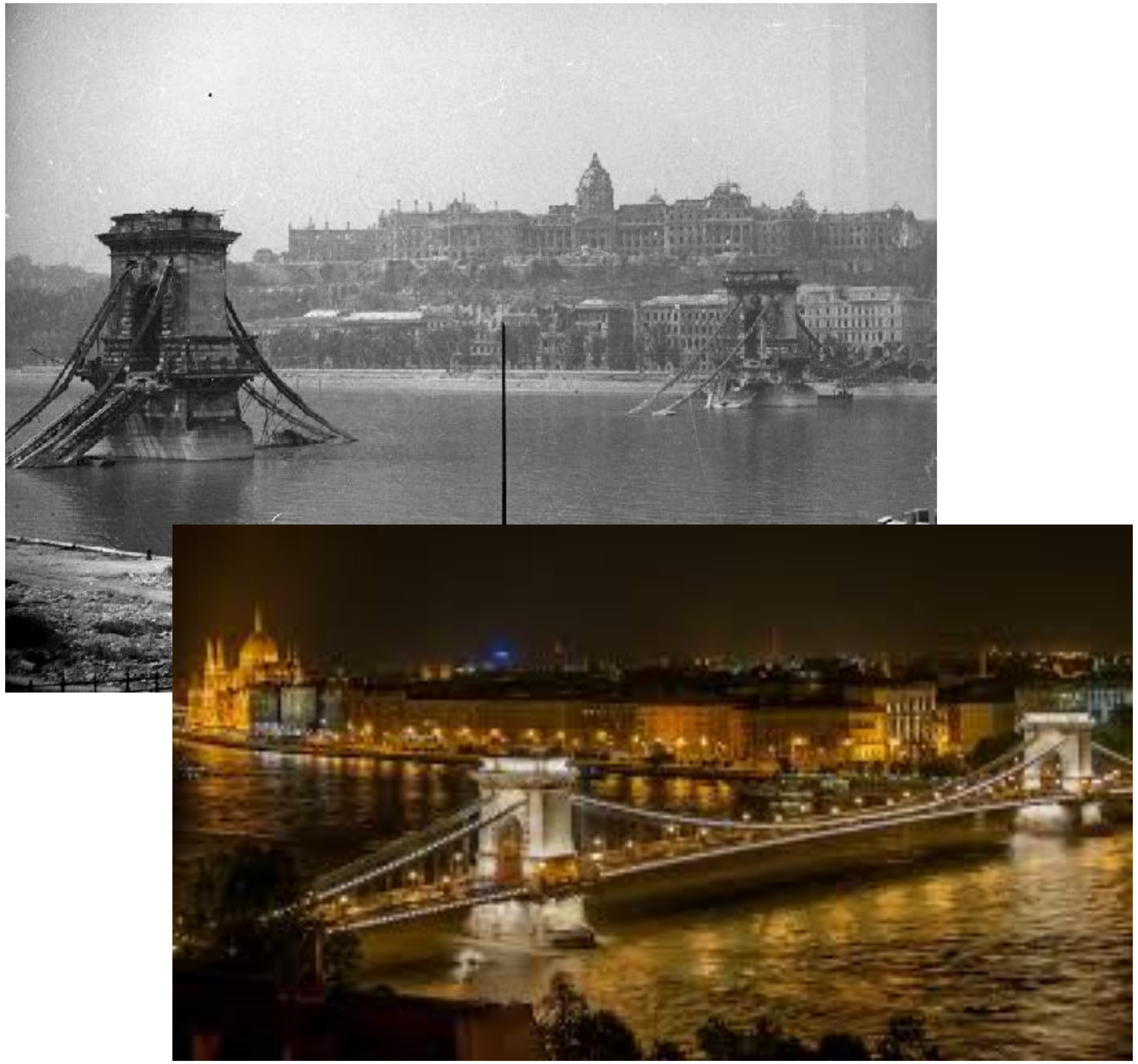
Can point to problems with congestion:



Lämmer, Stefan, Björn Gehlsen, and Dirk Helbing. *Physica A* 363.1 (2006): 89-95.

Betweenness centrality identifies bottlenecks in transport networks





Video by Luis Guillermo Natera Orozco

Street network orientation

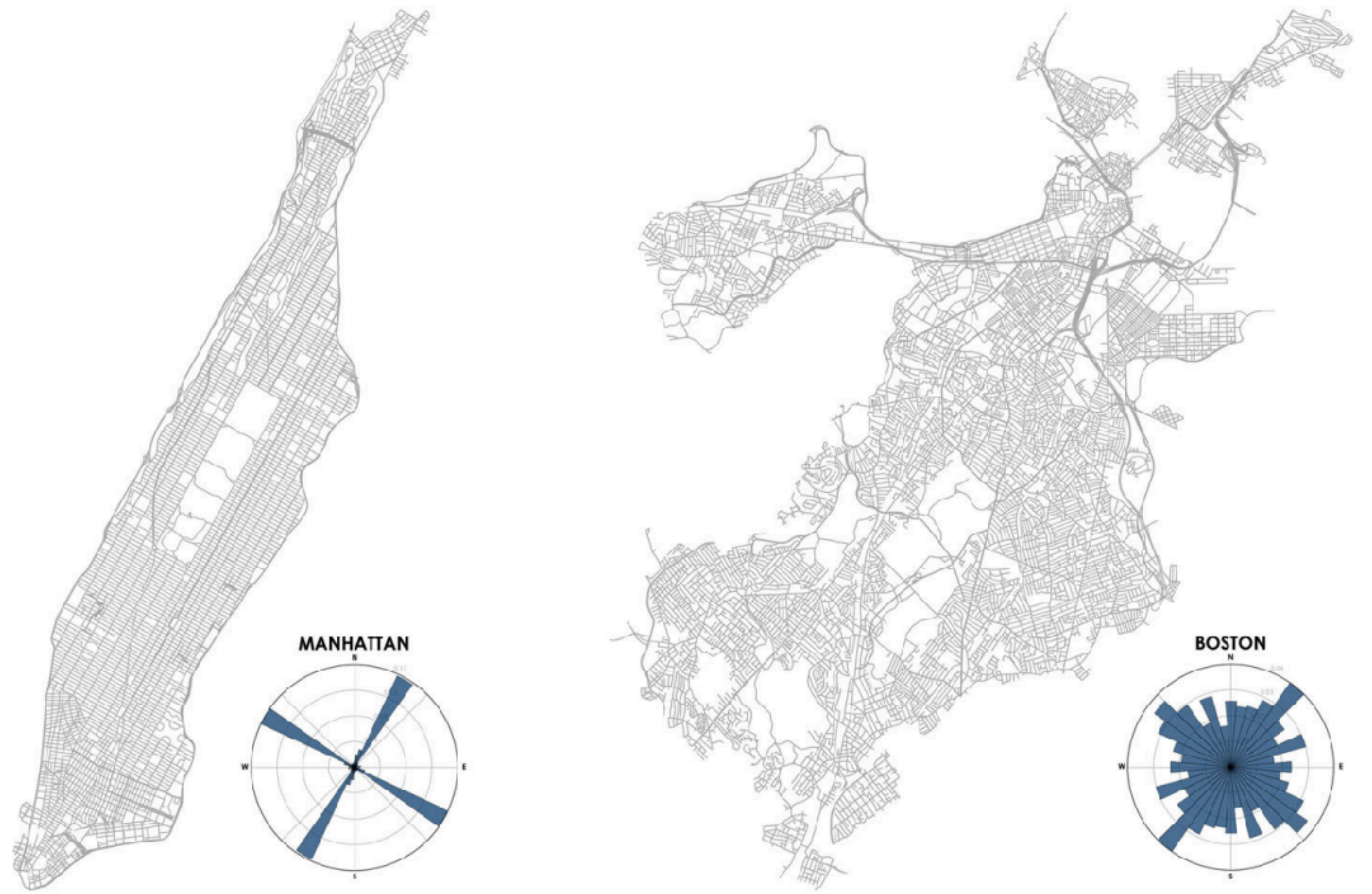


Fig. 3 Street networks and corresponding polar histograms for Manhattan and Boston

Geoff Boeing. Applied Network Science 4:67 (2019)

Street network orientation

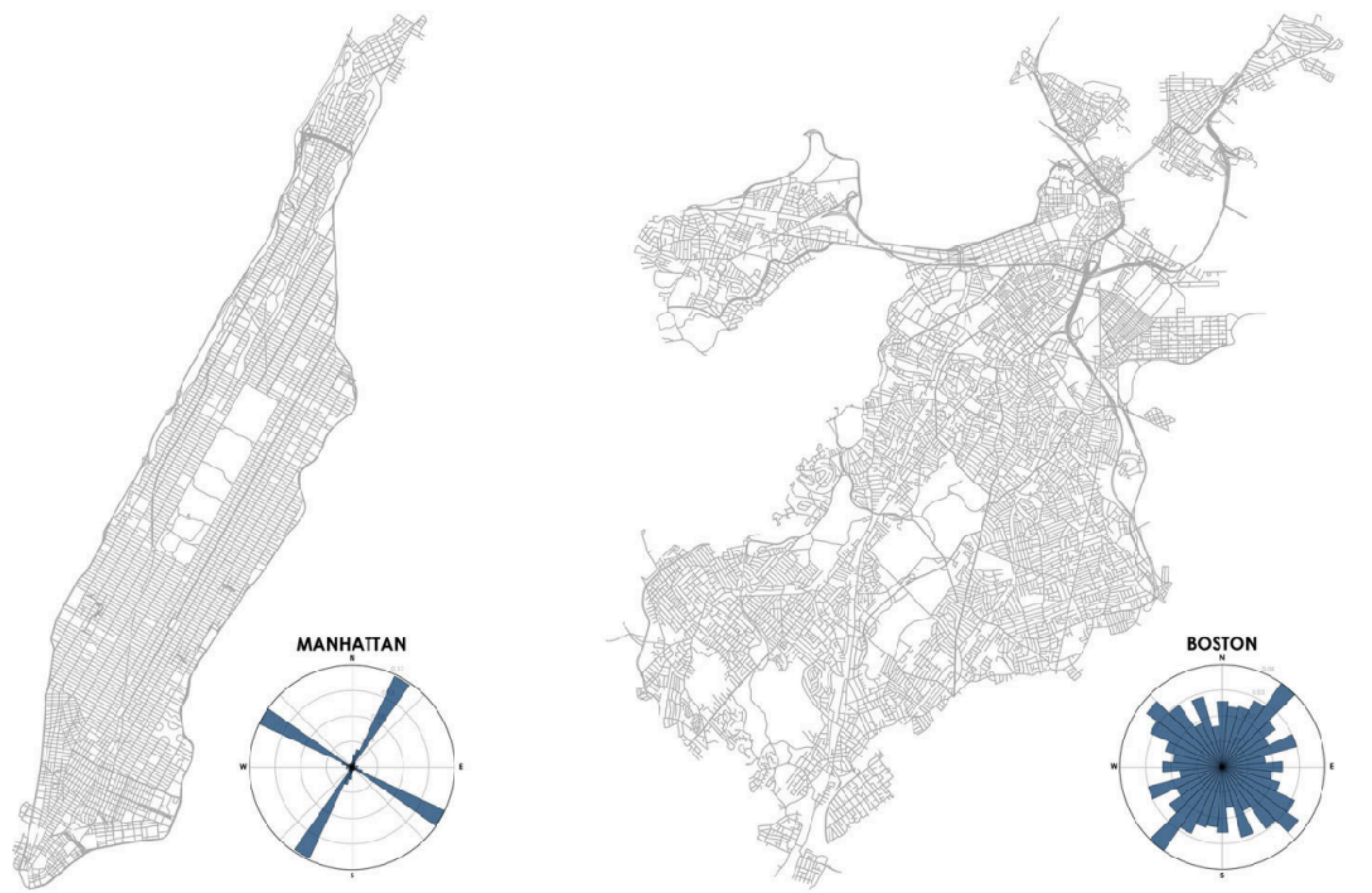


Fig. 3 Street networks and corresponding polar histograms for Manhattan and Boston

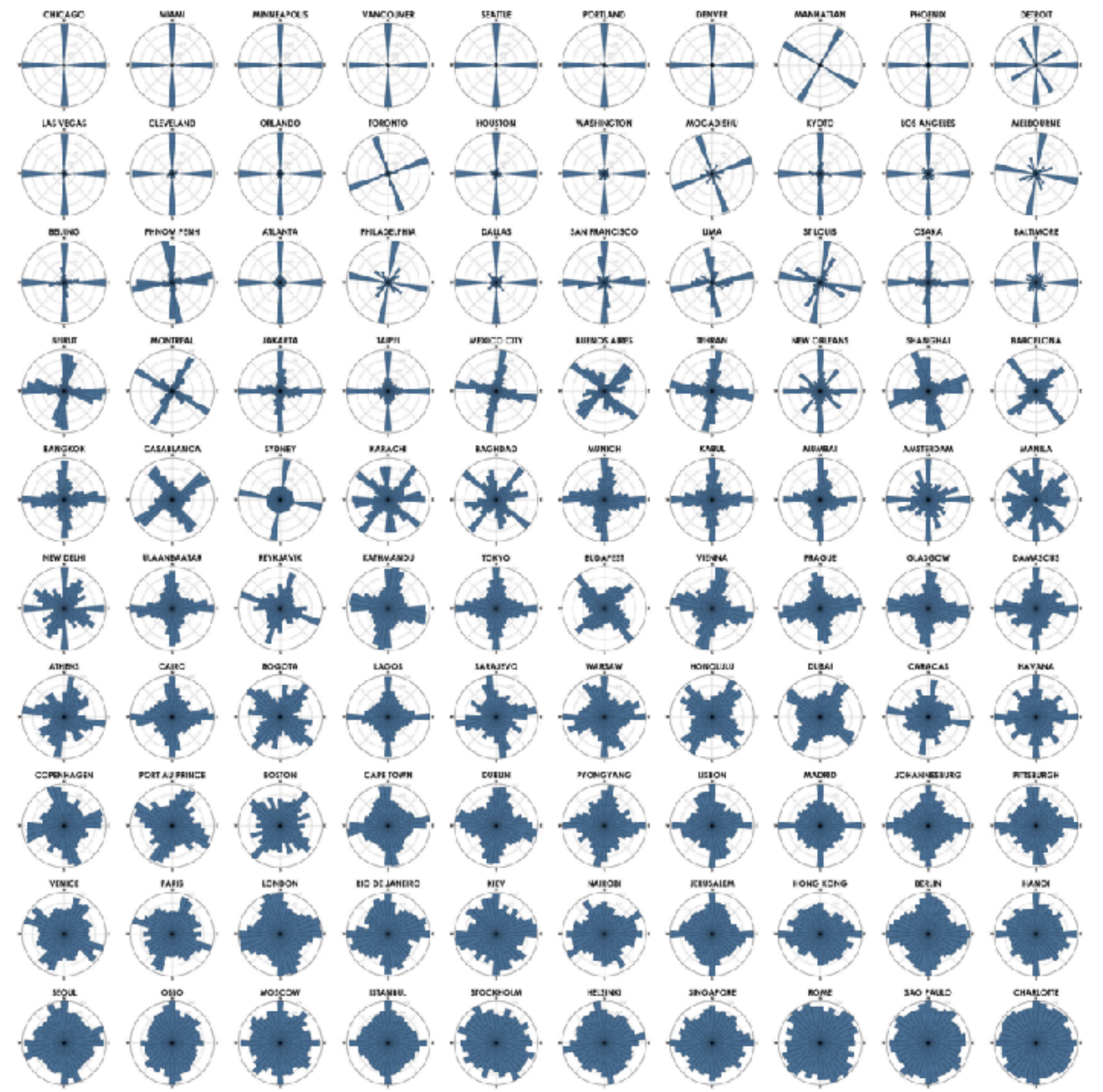
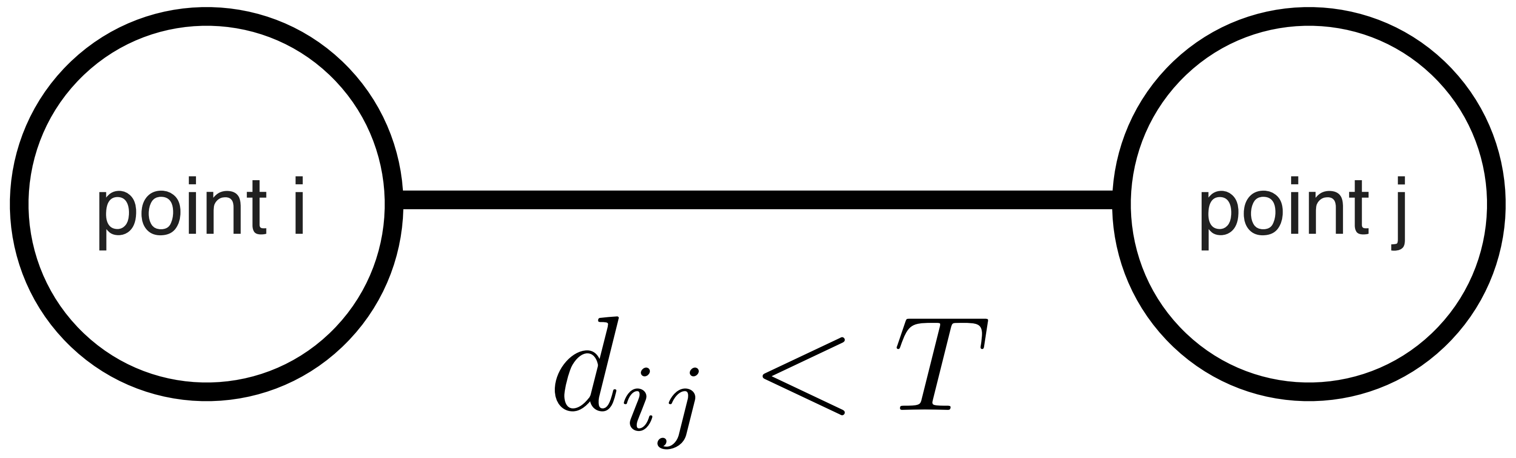


Fig. 5 Polar histograms from Fig. 4, resorted by descending φ from most to least grid-like (equivalent to least to greatest entropy)

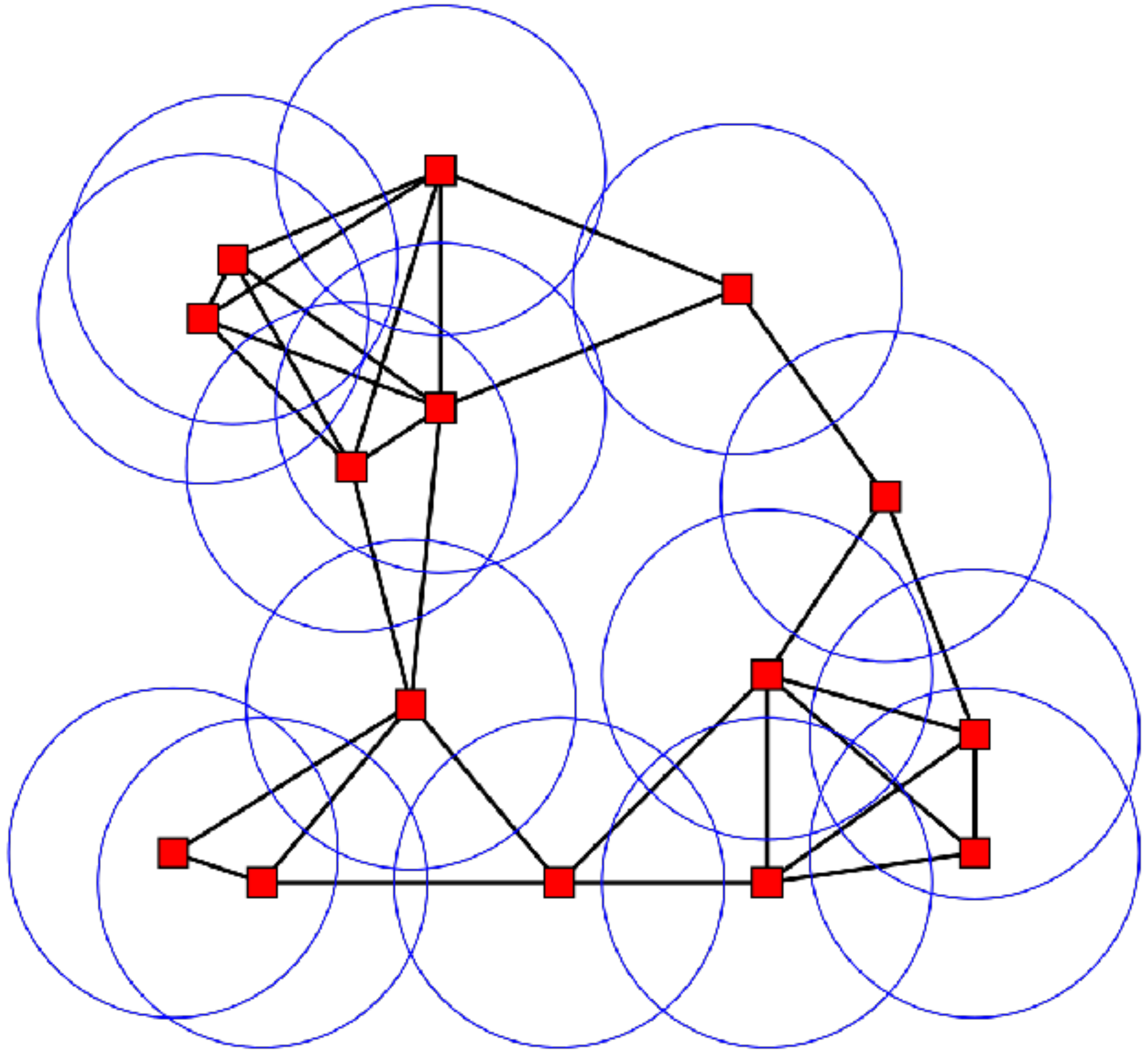
Geoff Boeing. Applied Network Science 4:67 (2019)

Spatial network models

Model: Random geometric graph



distance-based

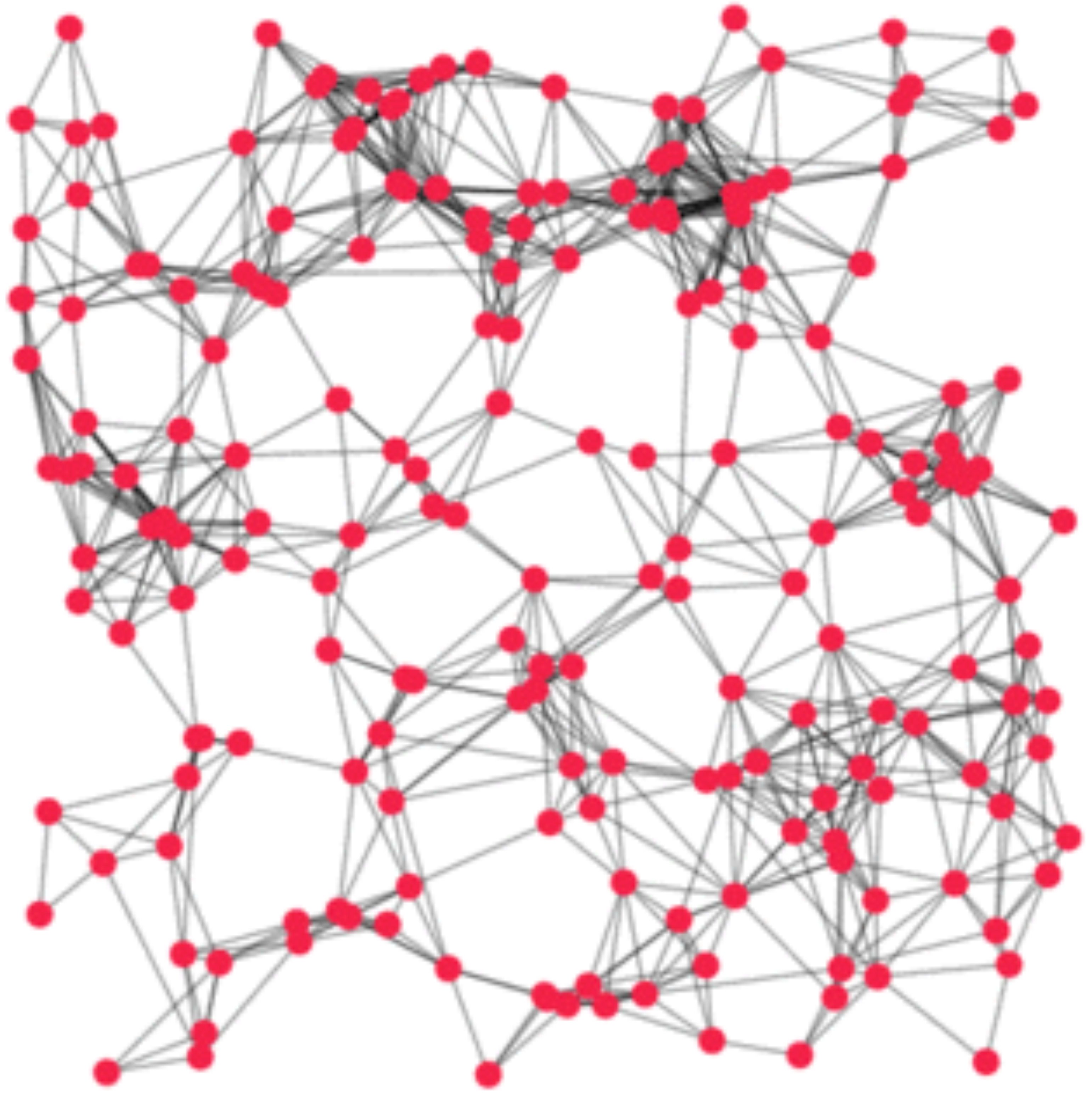


1) Generate random nodes

2) Place links between nodes with distance <T

Model for wireless ad-hoc mobile/vehicle networks

Model: Waxman



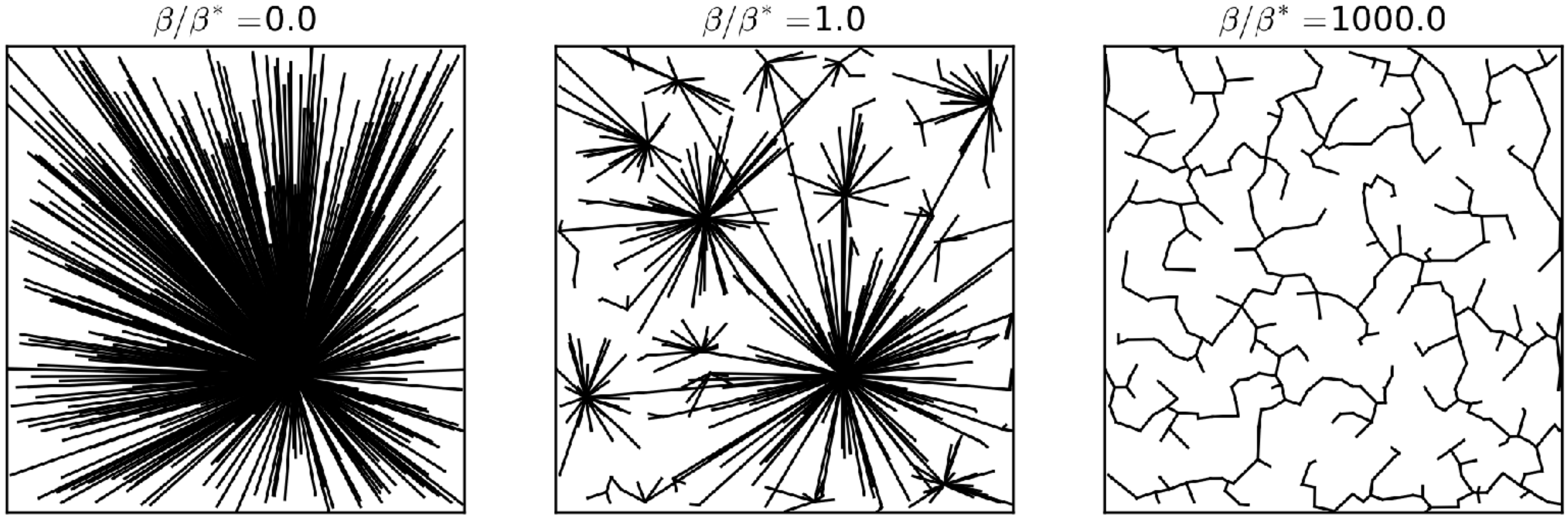
1) Generate random nodes

2) Add links depending on the Euclidian distance

$$P(i,j) = \beta e^{d(i,j)/d_0}$$

Barthélemy, Marc. "Spatial networks." *Physics Reports* 499.1-3 (2011): 1-101.

Model: Hierarchical



Star graph | Spatial hierarchy | Minimum spanning tree

The average detour is minimal here

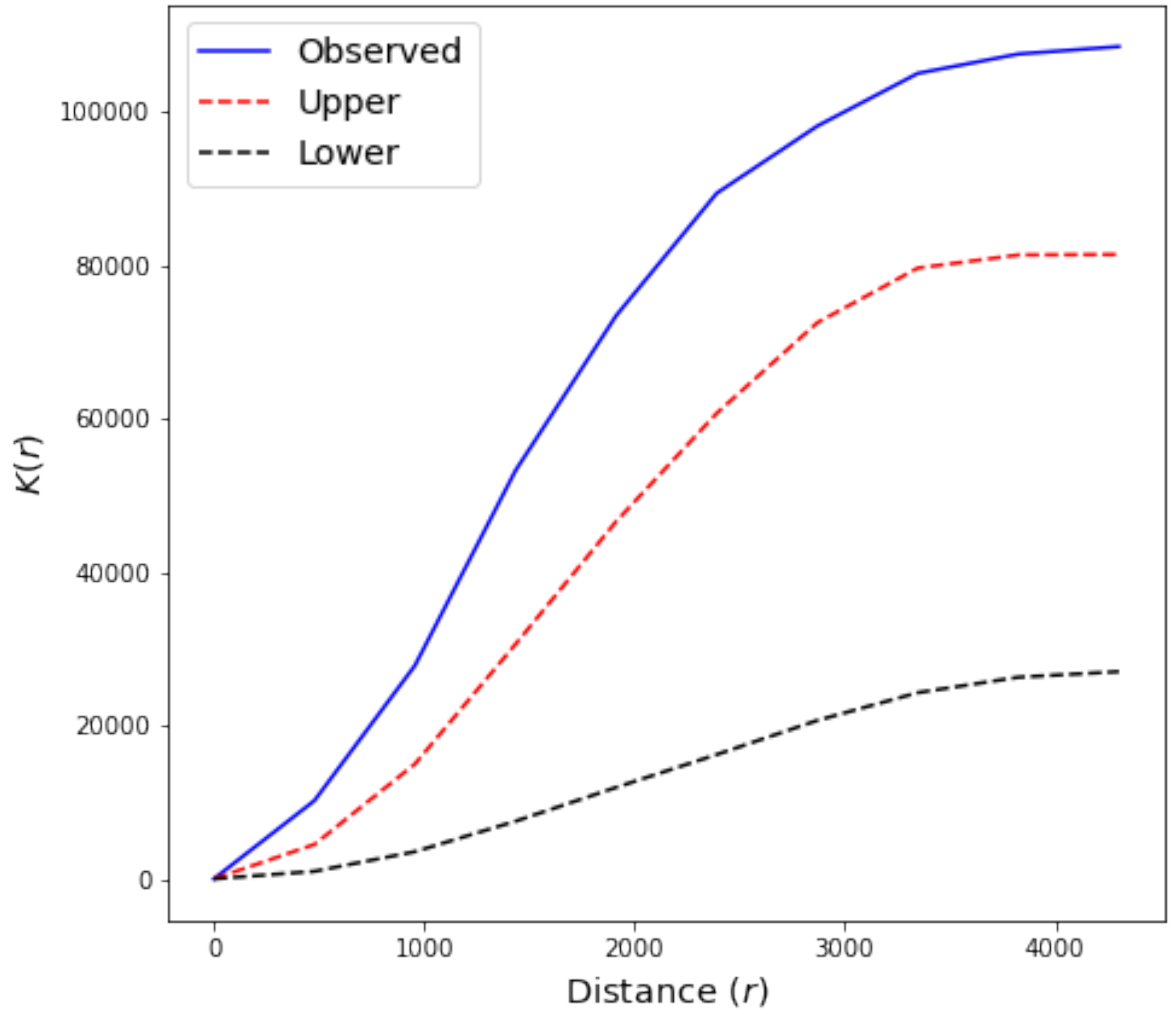
Model railway networks between cities

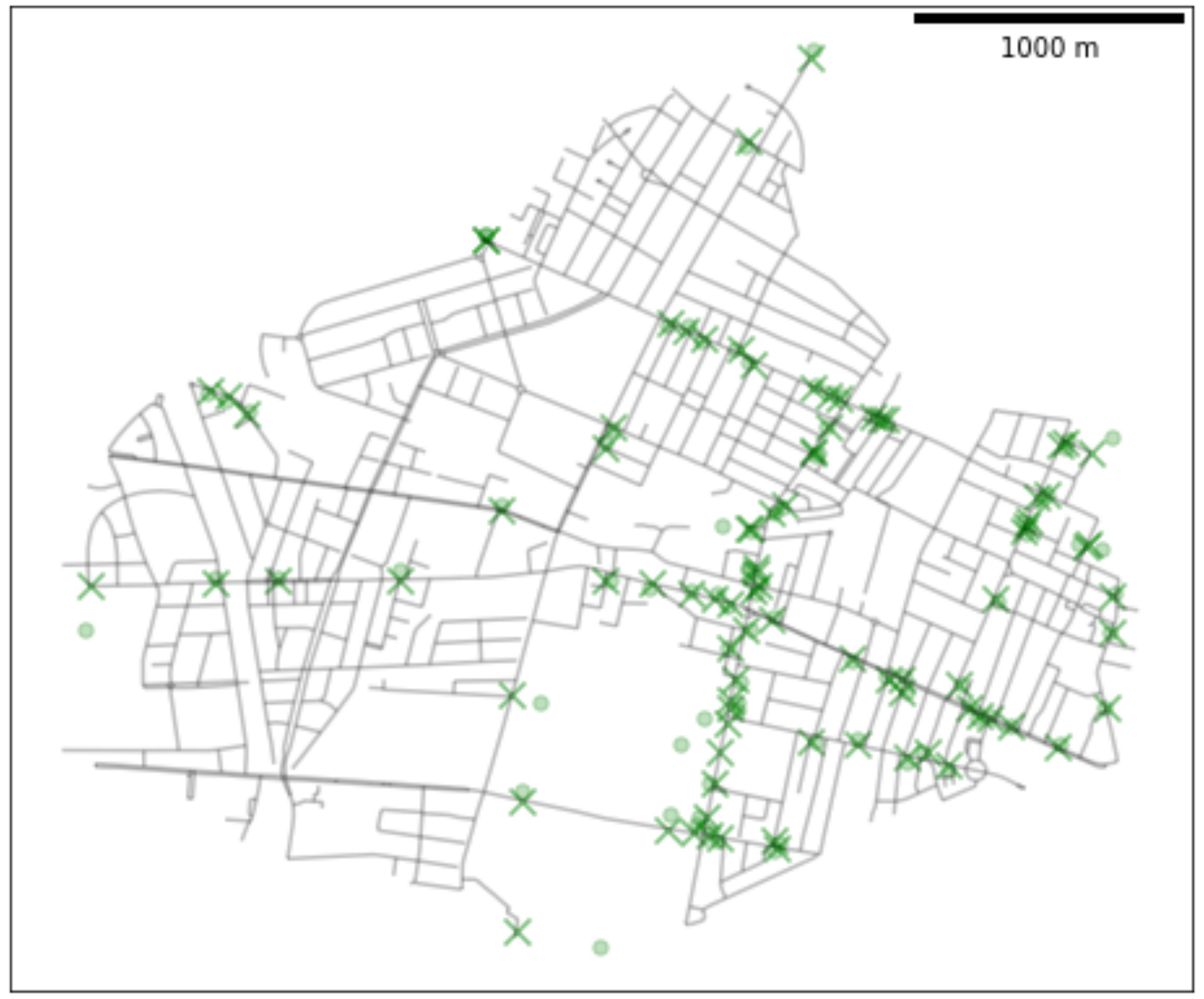
Louf, Rémi, Pablo Jensen, and Marc Barthelemy. *PNAS* 110, 22 (2013)

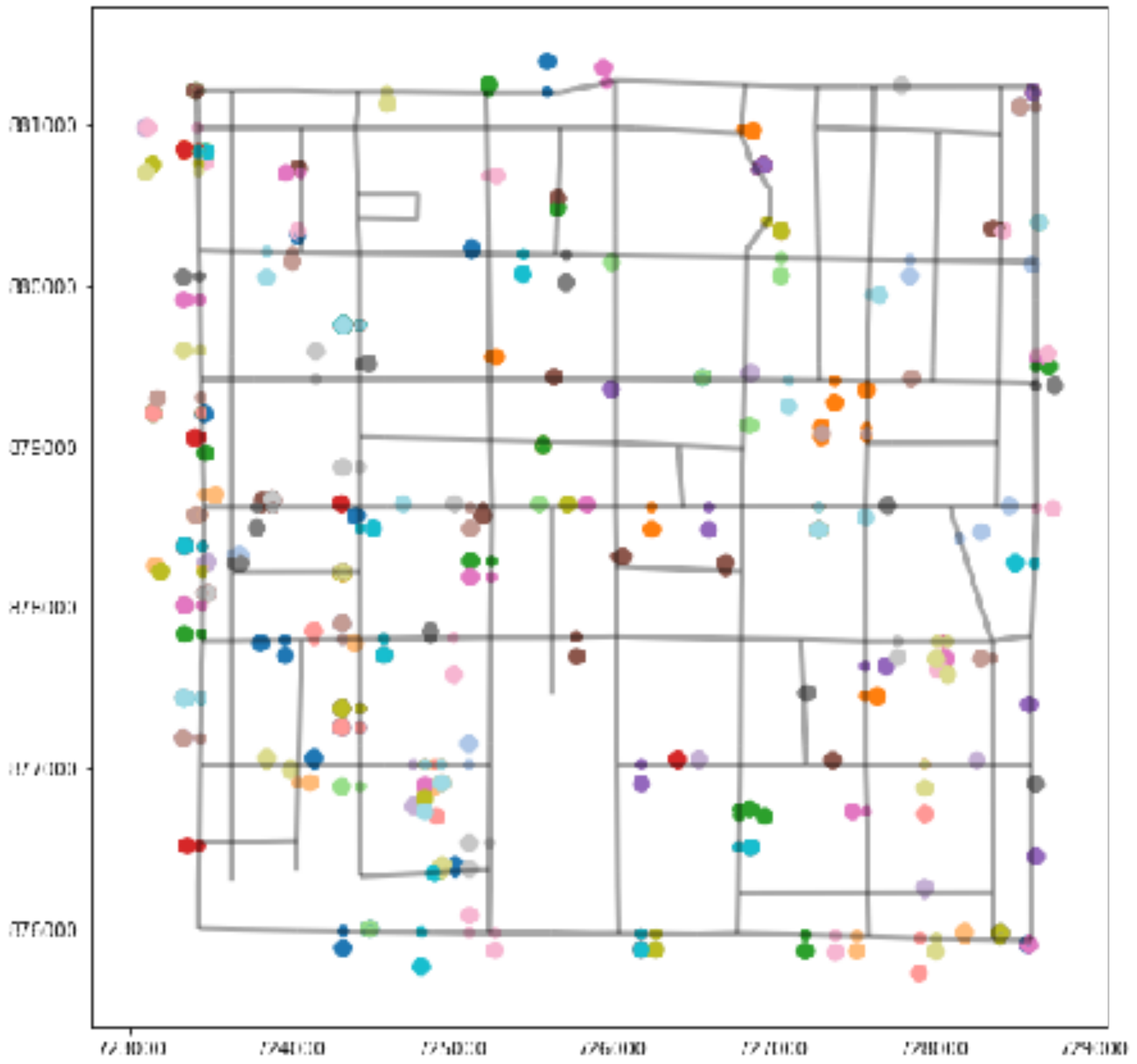
Ripley's $K(d)$ function measures clustering for a distance d

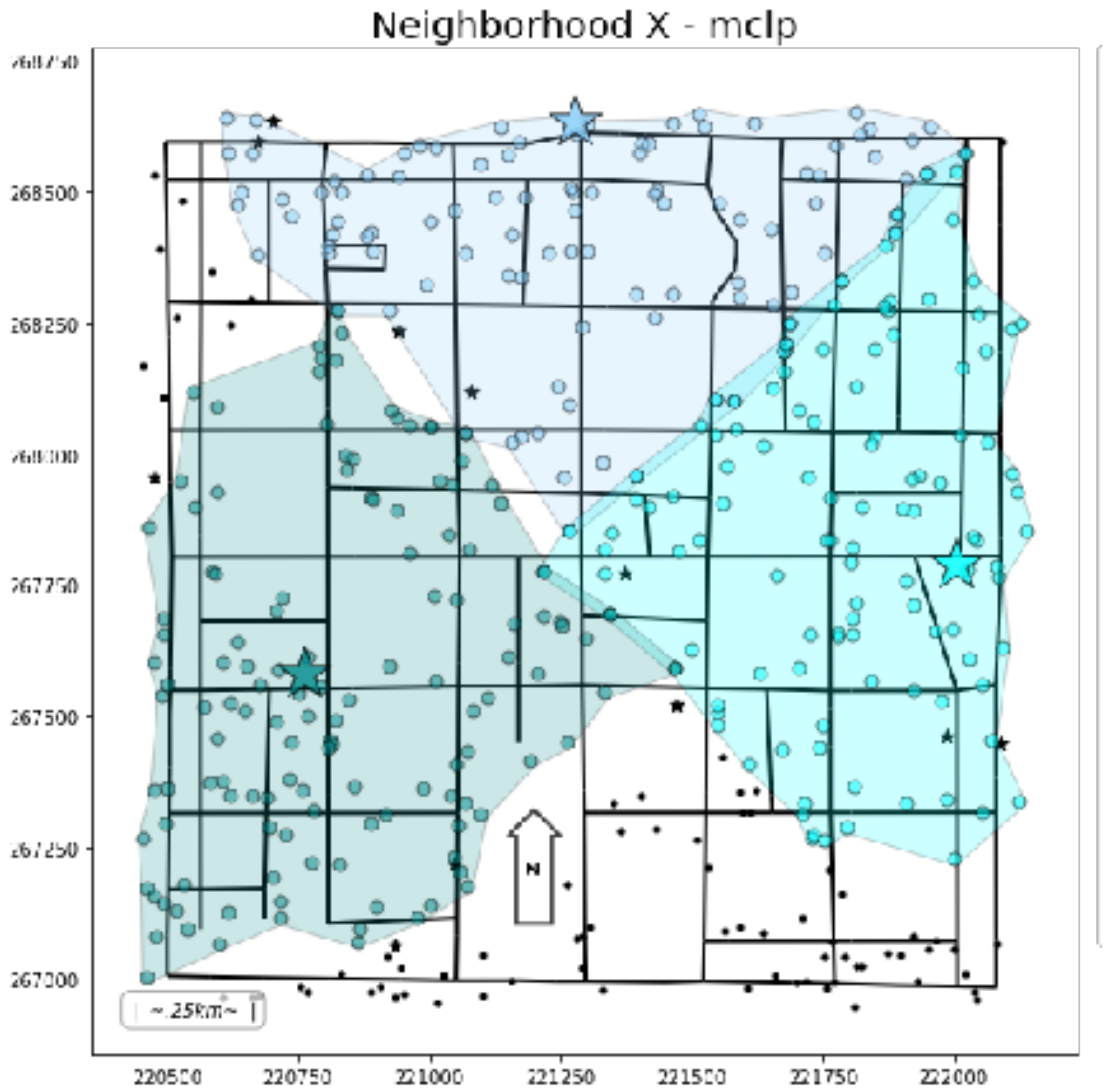
Ripley's $K(d)$ function is the expected number of points within distance d from any randomly sampled point, corrected for density.

Useful Python libraries: spaghetti and snkit





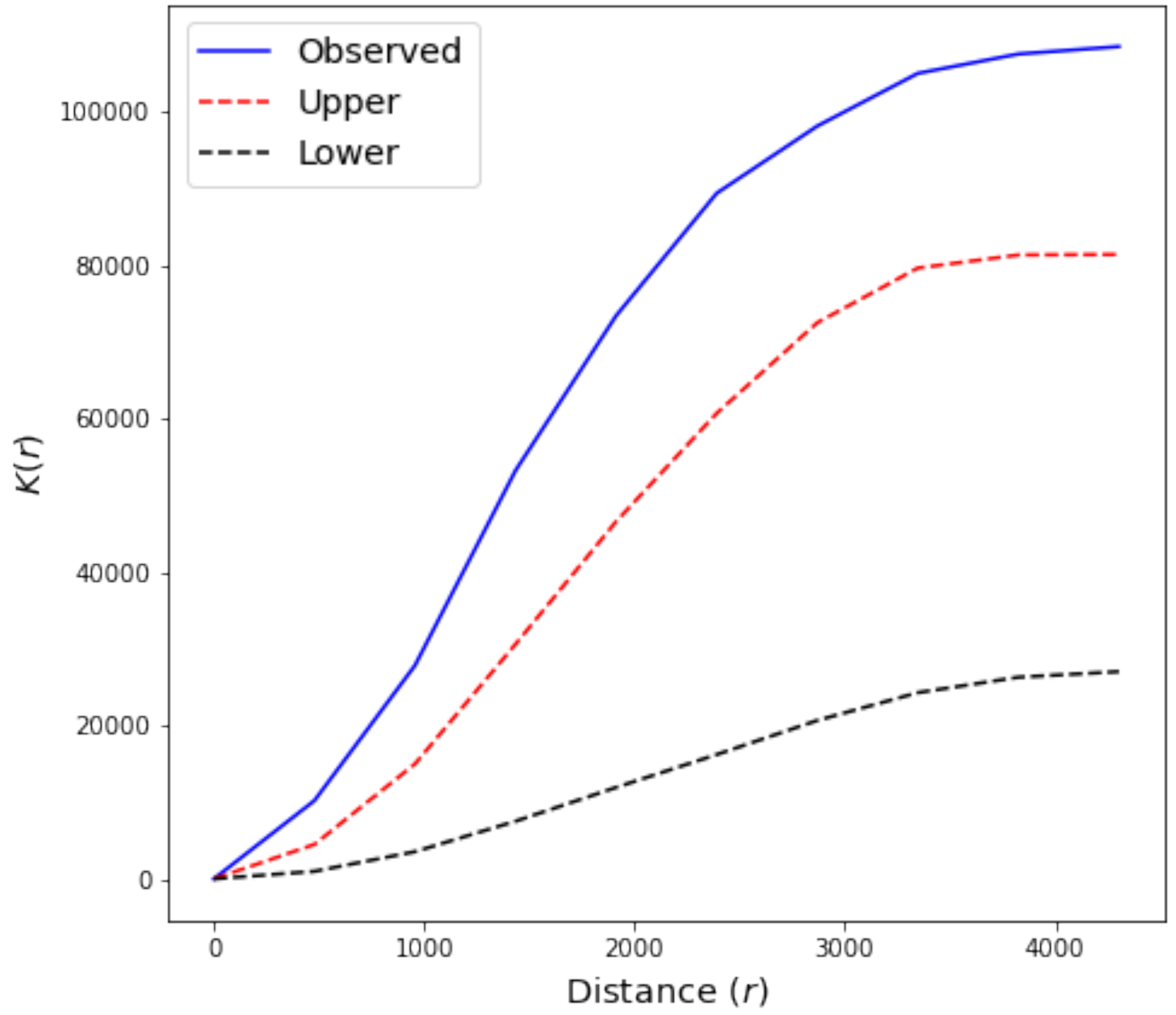


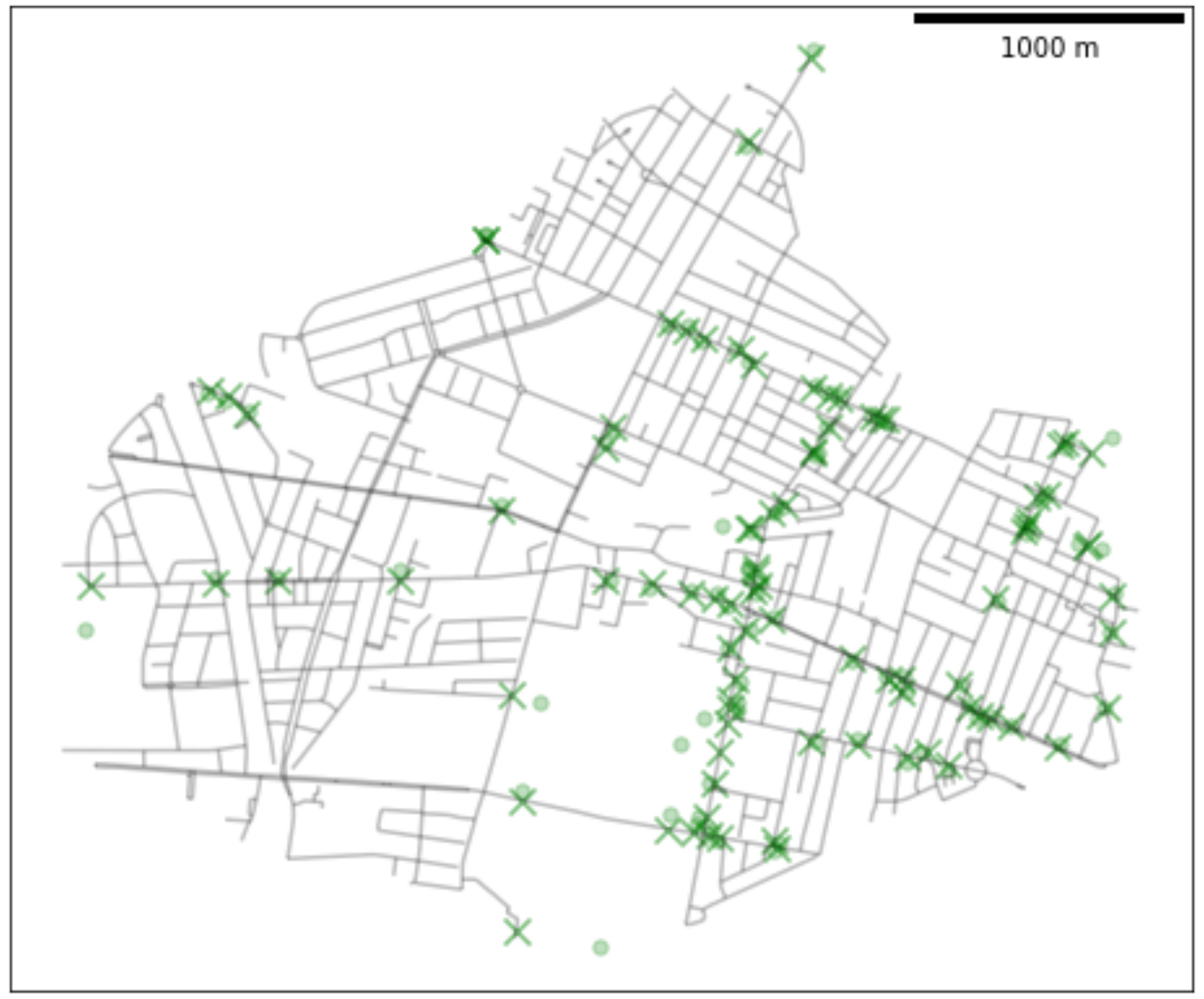


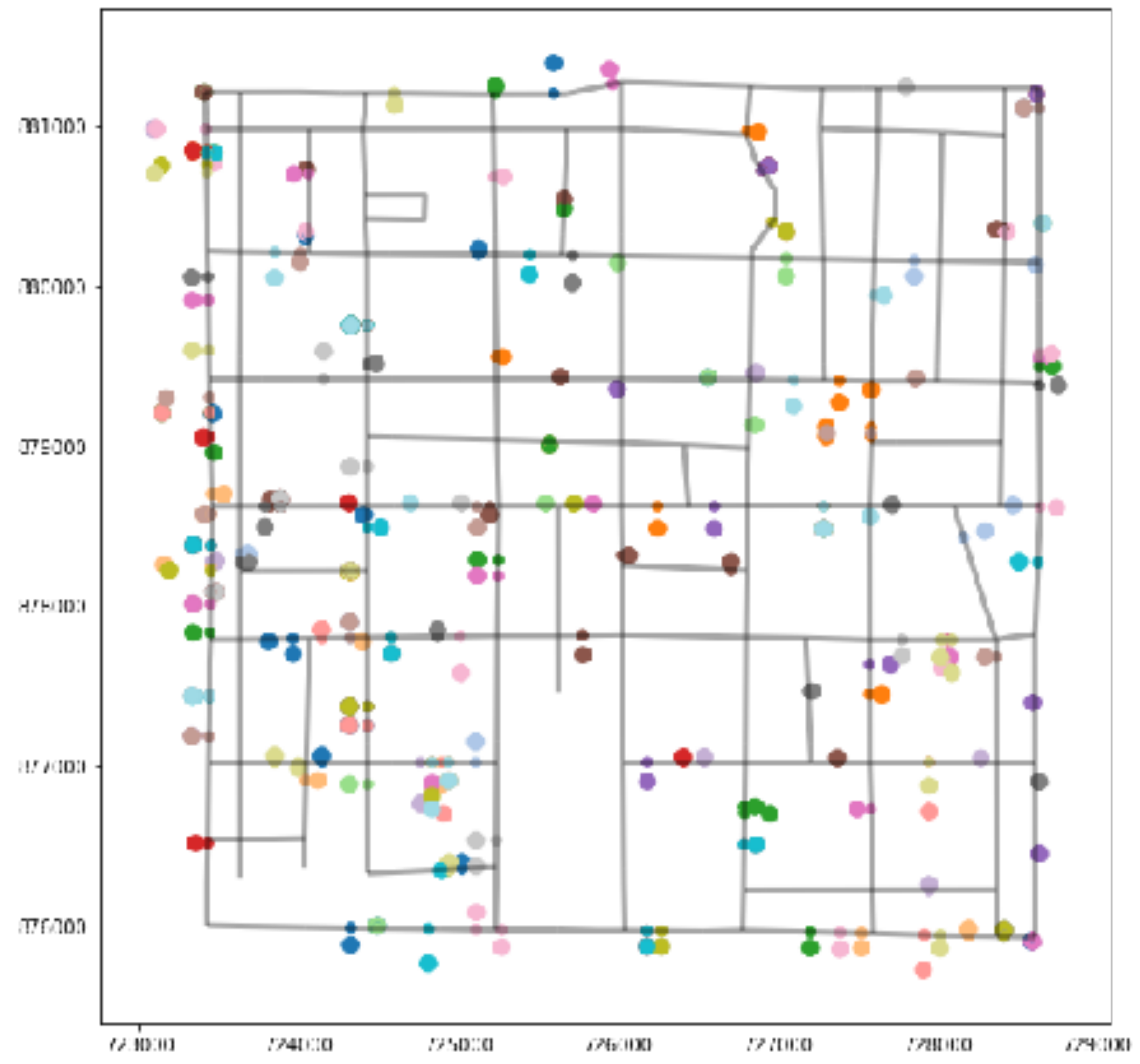
spaghetti allows spatial analysis on networks

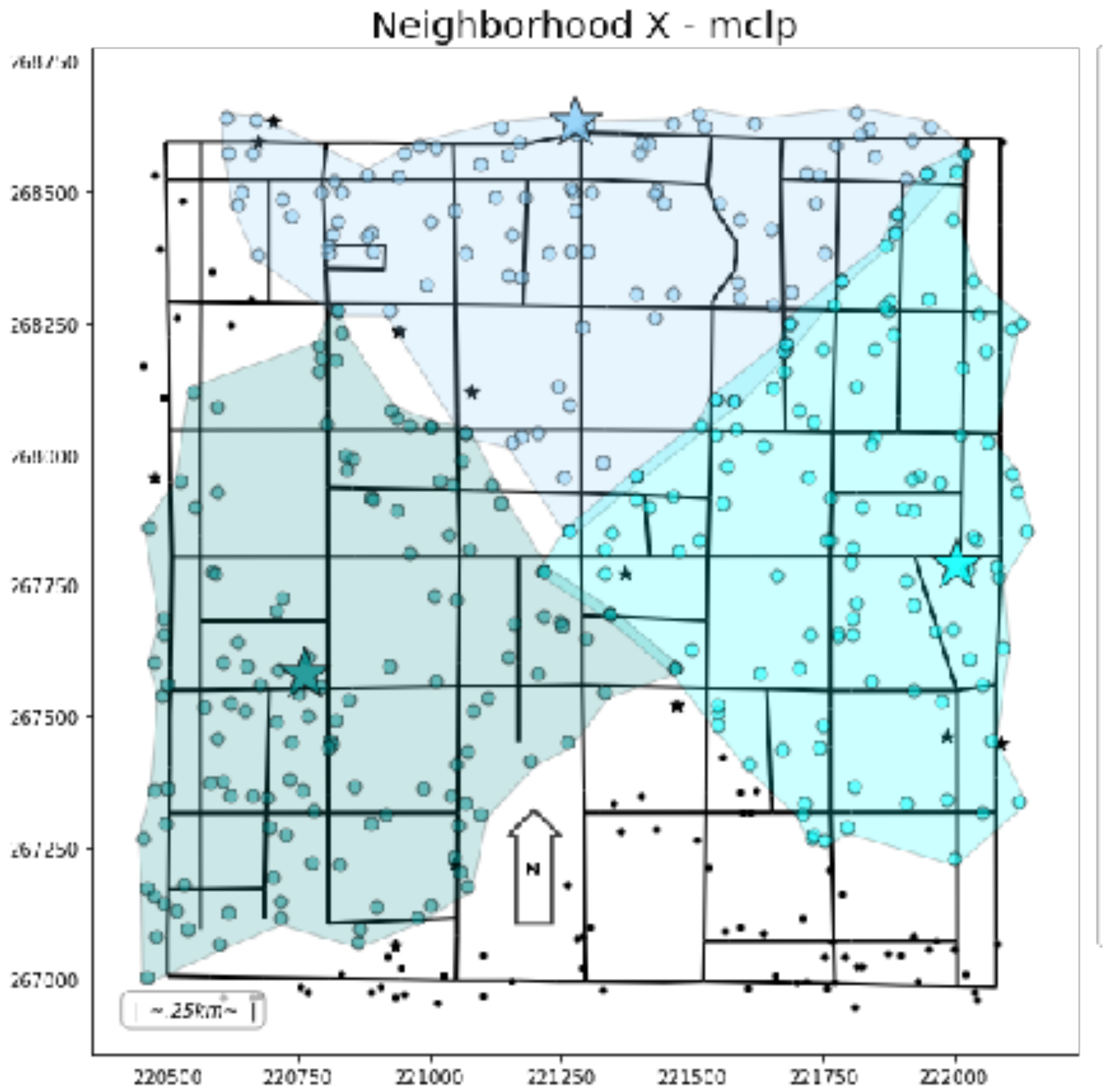
https://pysal-spaghetti.readthedocs.io/en/latest/index.html

Useful Python libraries: spaghetti and snkit



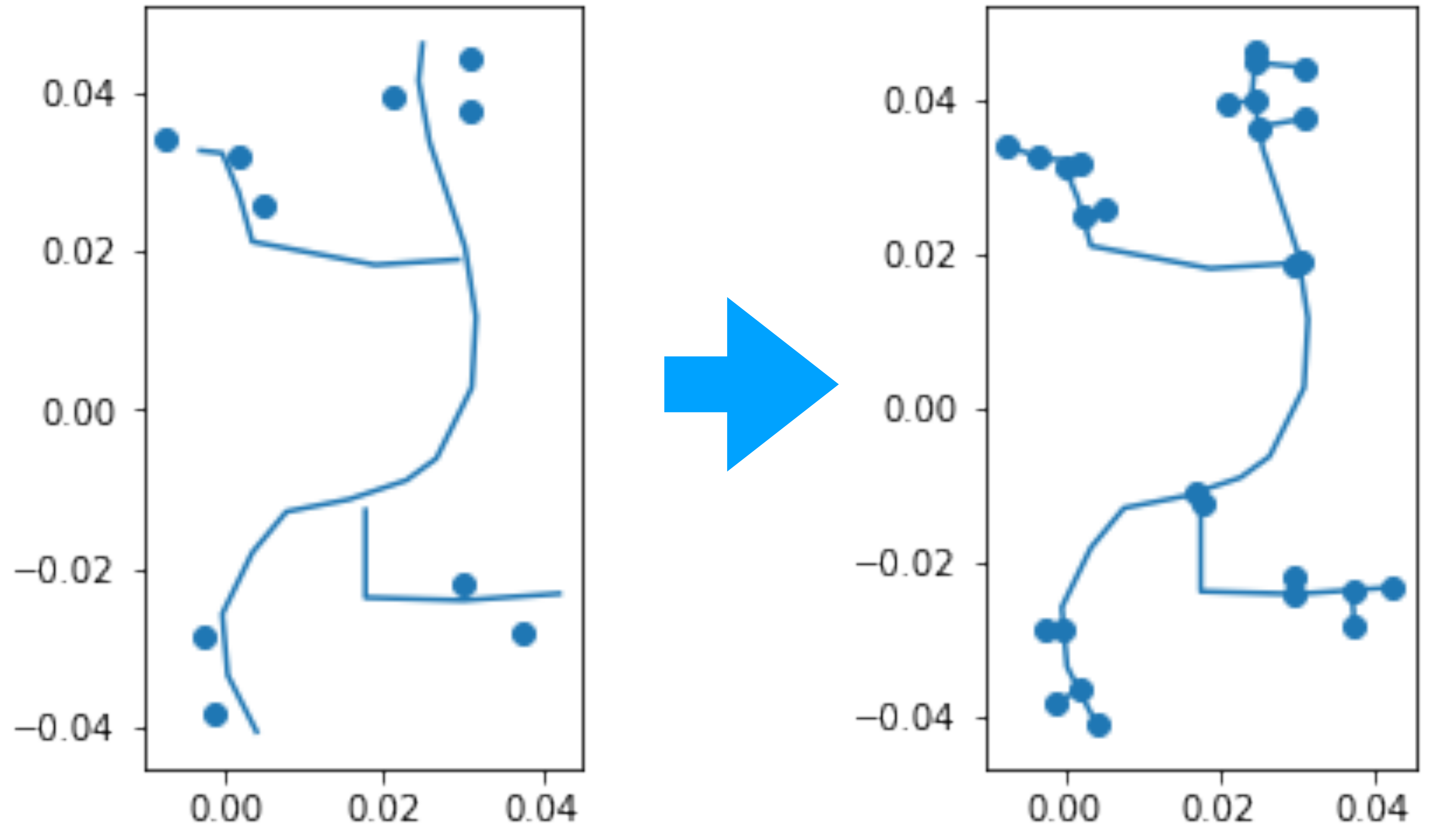






spaghetti allows spatial analysis on networks

snkit helps tidy spatial network data

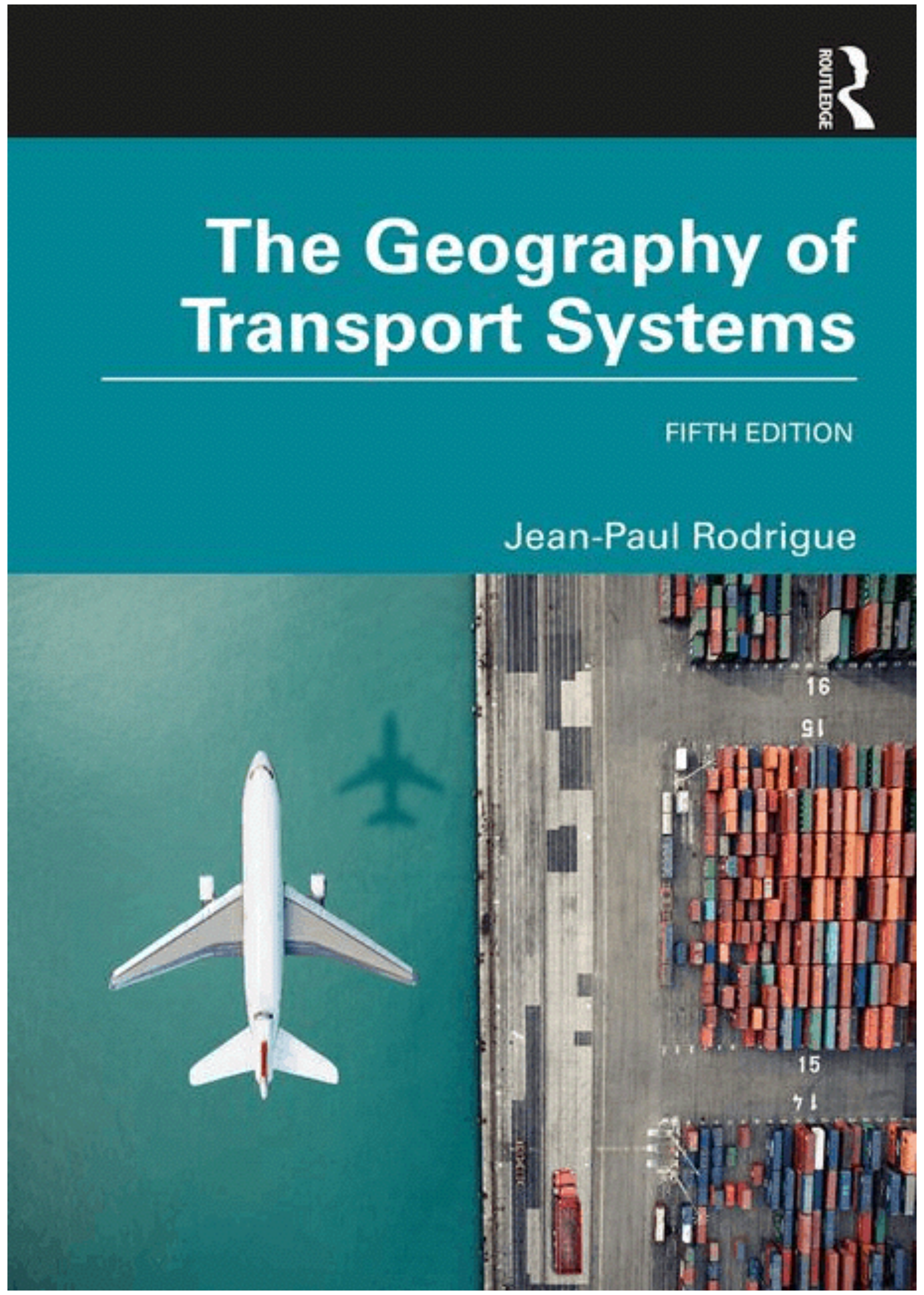


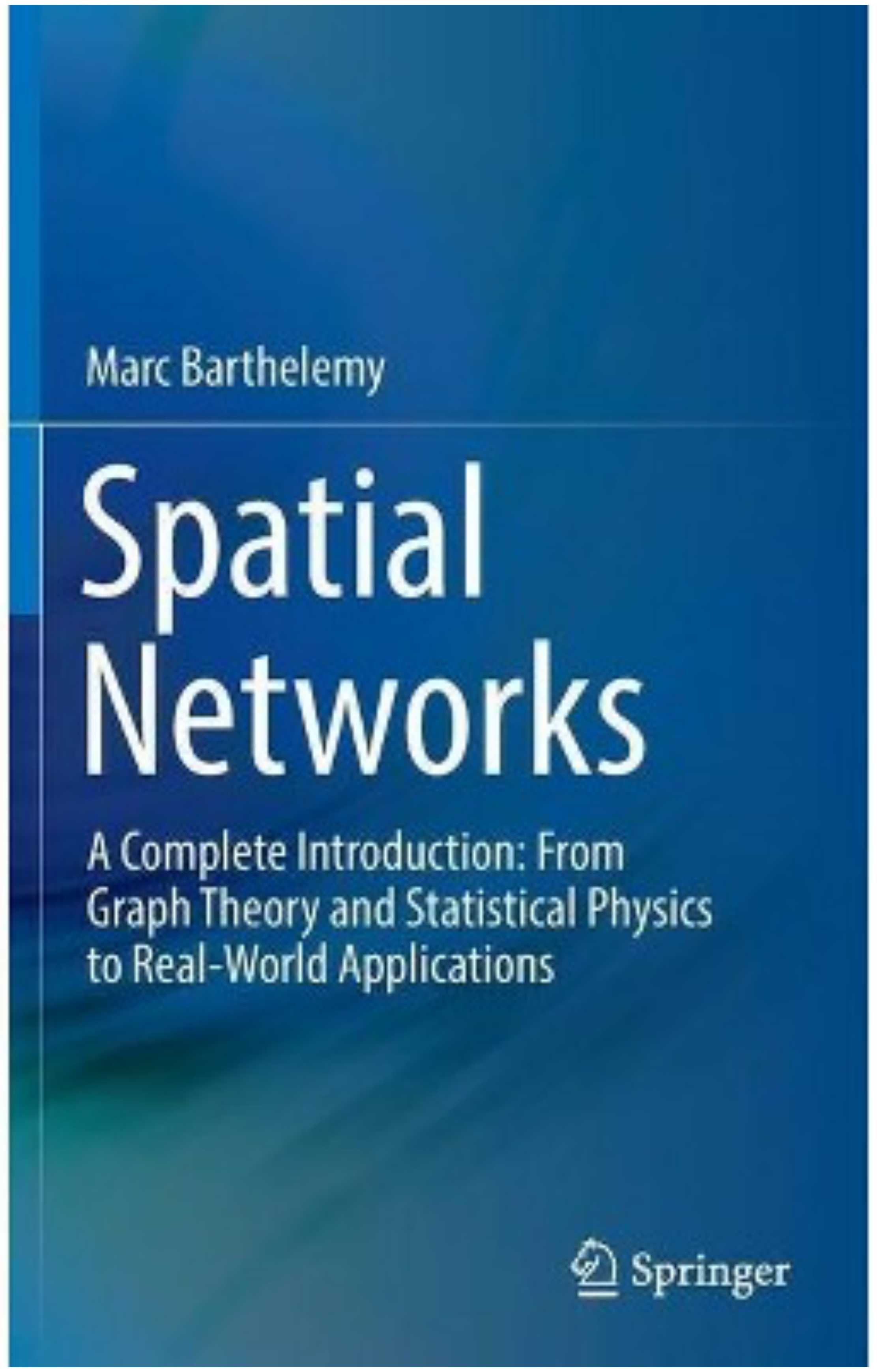
https://github.com/tomalrussell/snkit

https://pysal-spaghetti.readthedocs.io/en/latest/index.html

Jupyter

Sources and further materials for today's class





https://transportgeography.org/