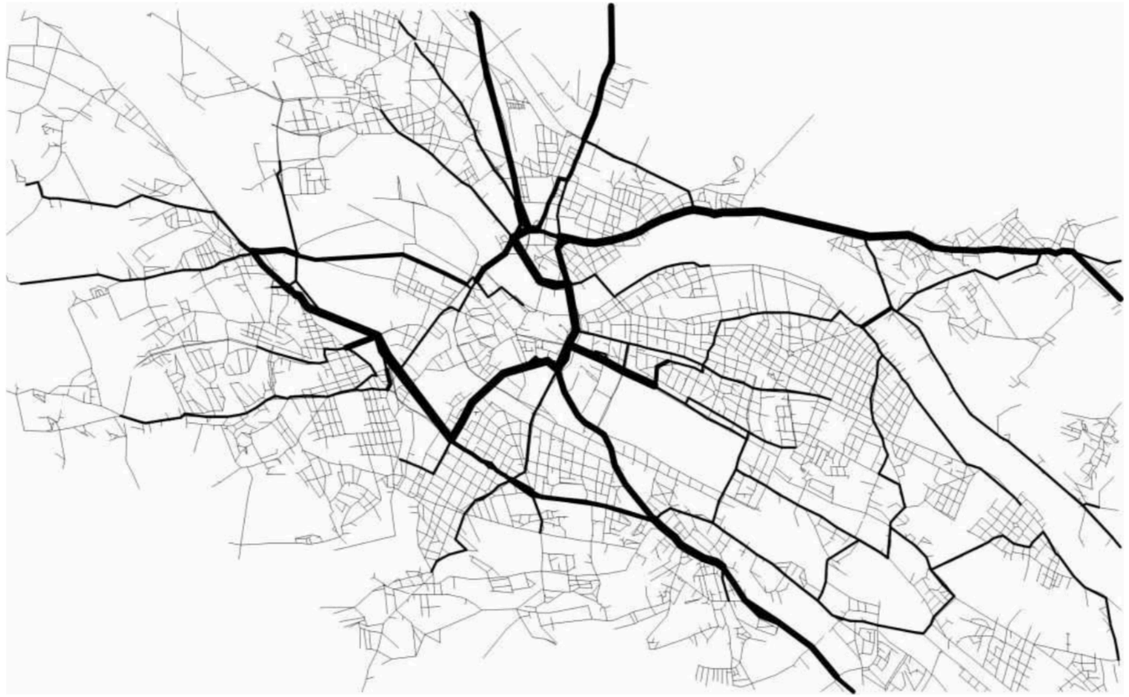
Lecture 10: Spatial Networks

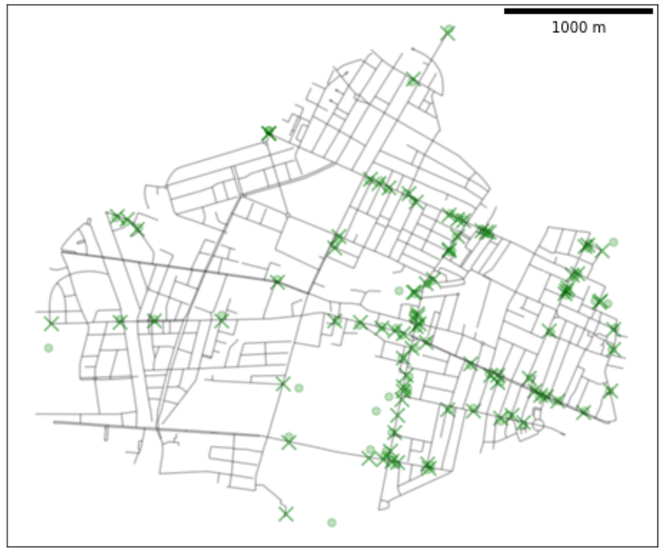
Instructor: Ane Rahbek Vierø
March 27, 2023



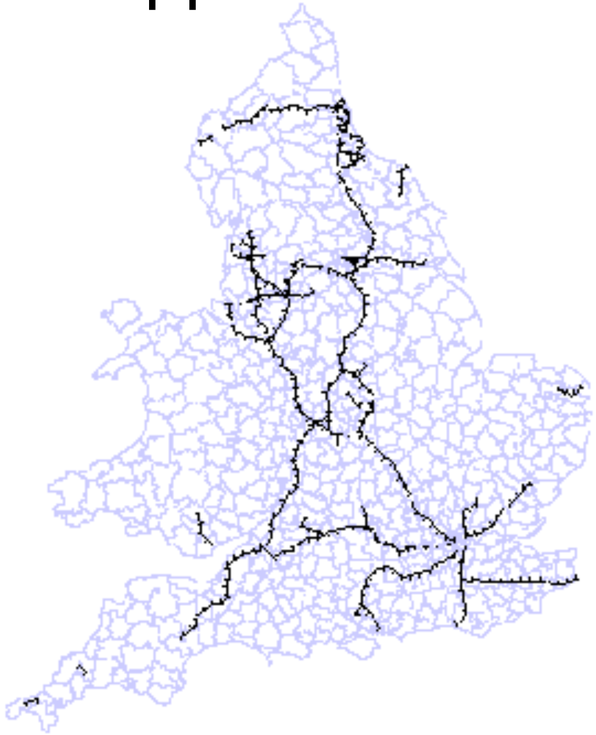
Today you will learn about spatial networks

Network intro + difference to non-spatial networks

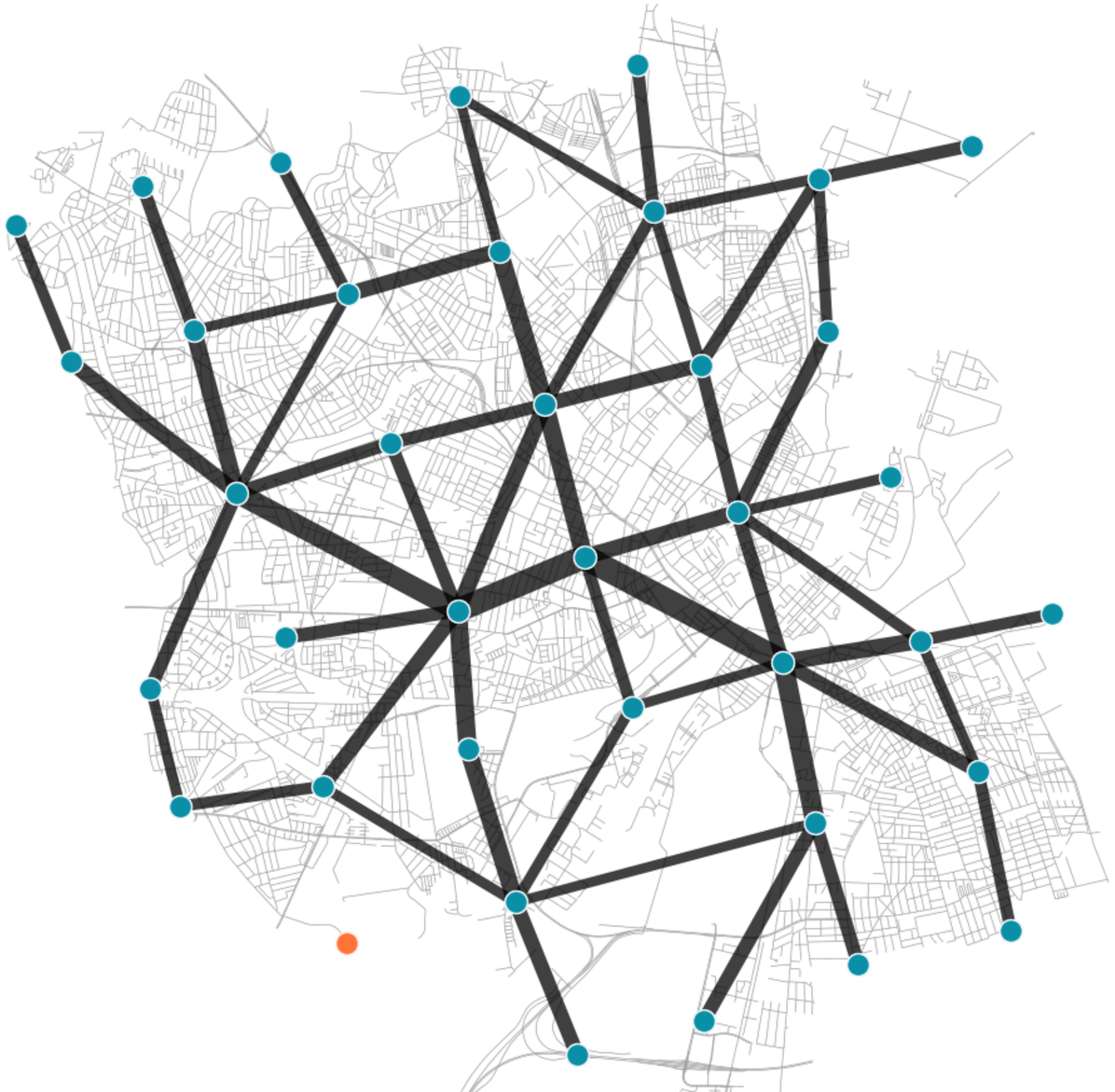
Python libraries for network analysis



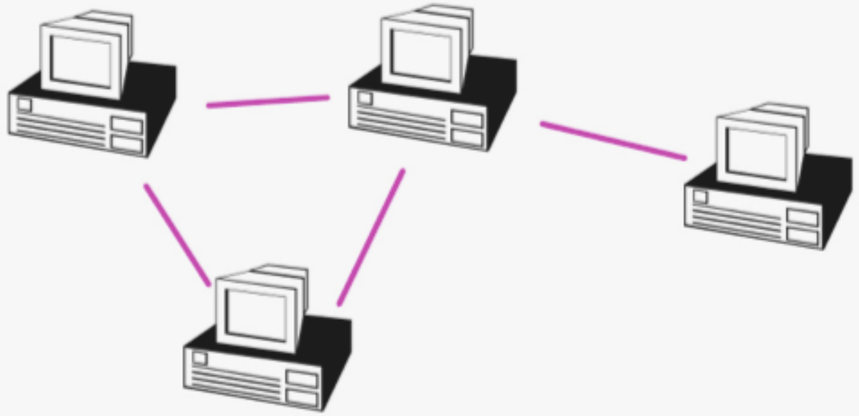
Optimal networks + spatial network applications

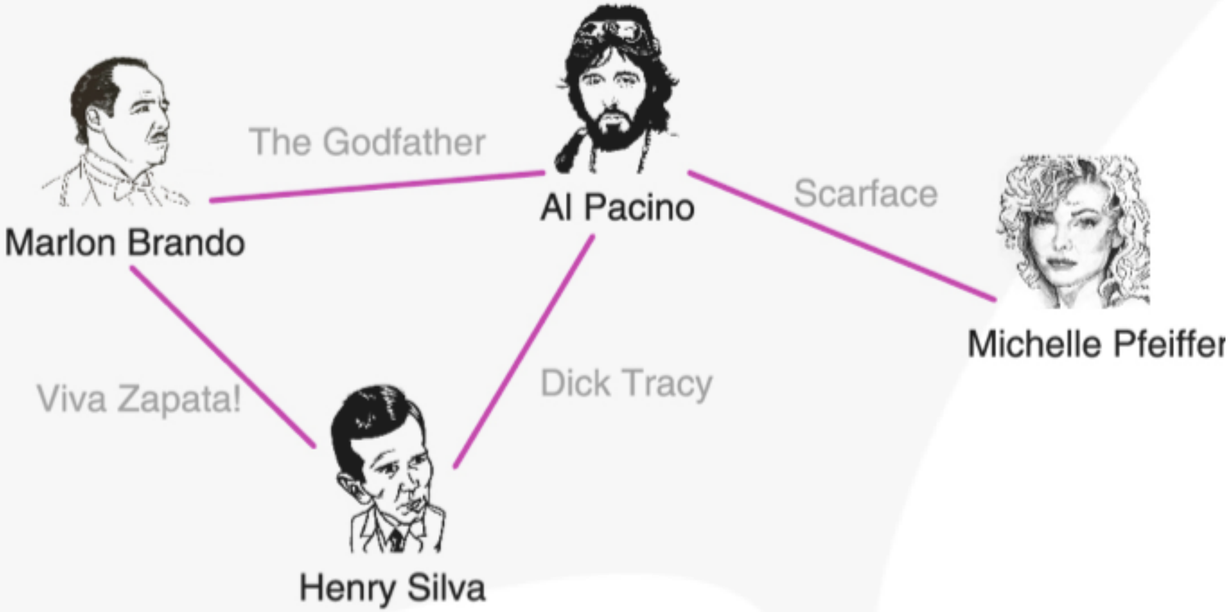


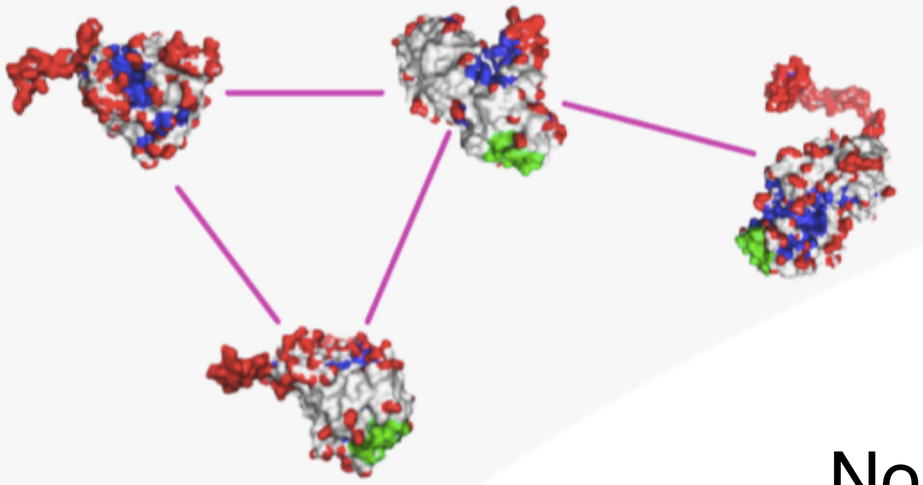
We model spatial data as networks when we care about spatial relationships and connectivity

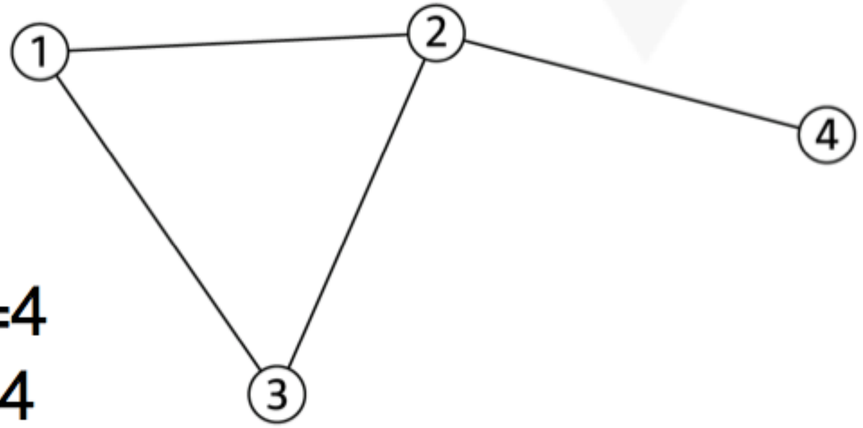


Networks are a common language for different applications







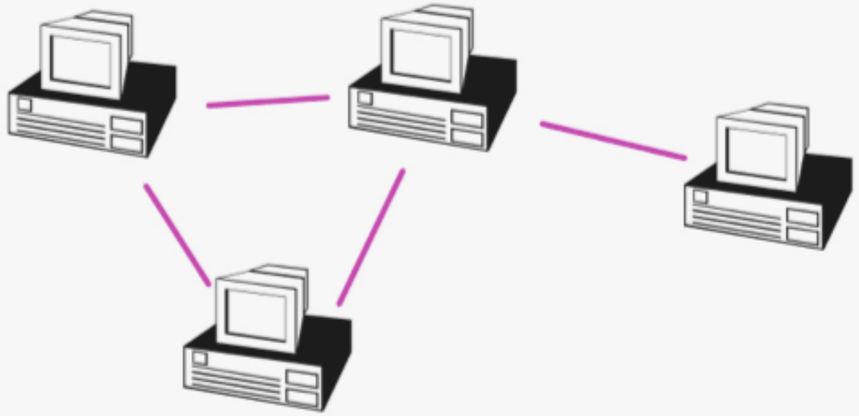


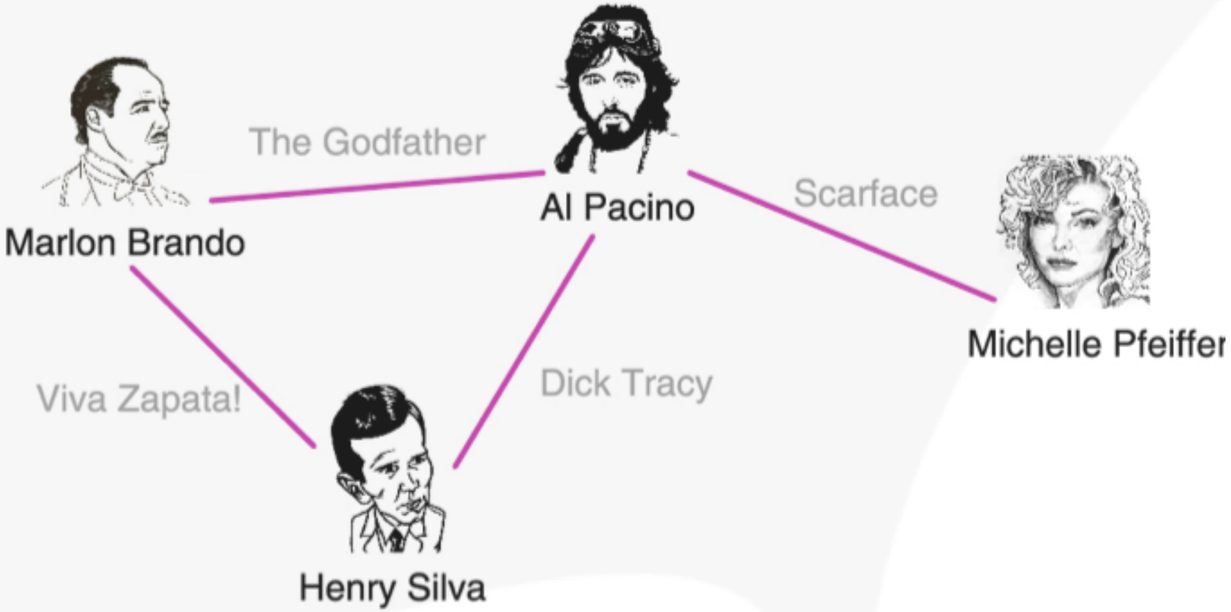
Nodes N=4

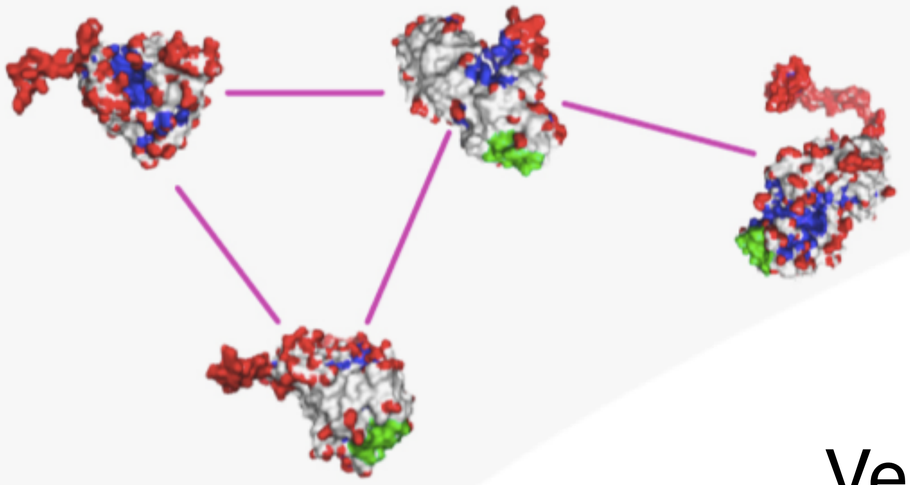
Links L=4

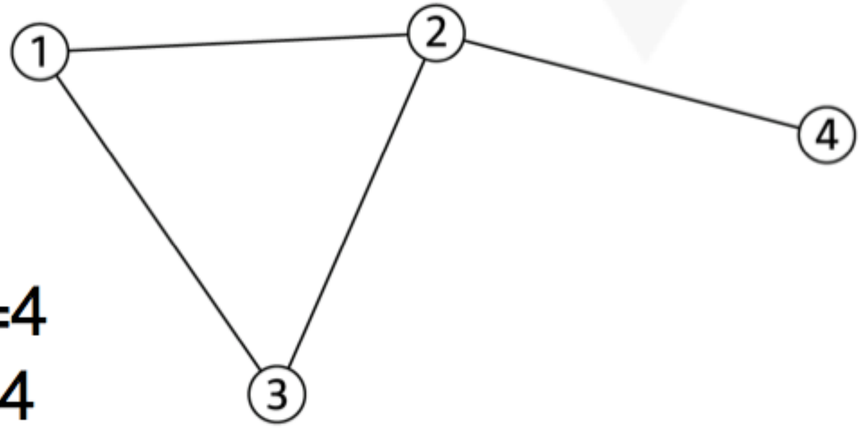
Graph $\mathcal{G} = (\mathcal{N}, \mathcal{L})$

Network = Graph + real-world meaning







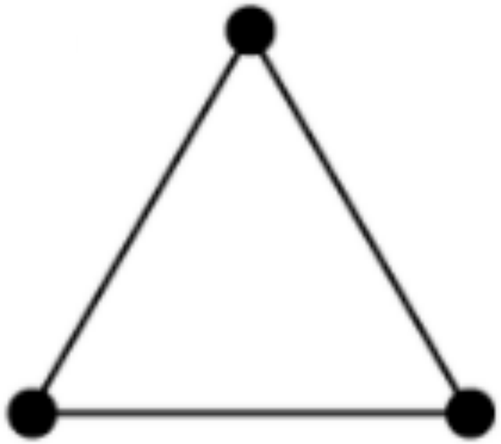


Vertices V=4

Edges E=4

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

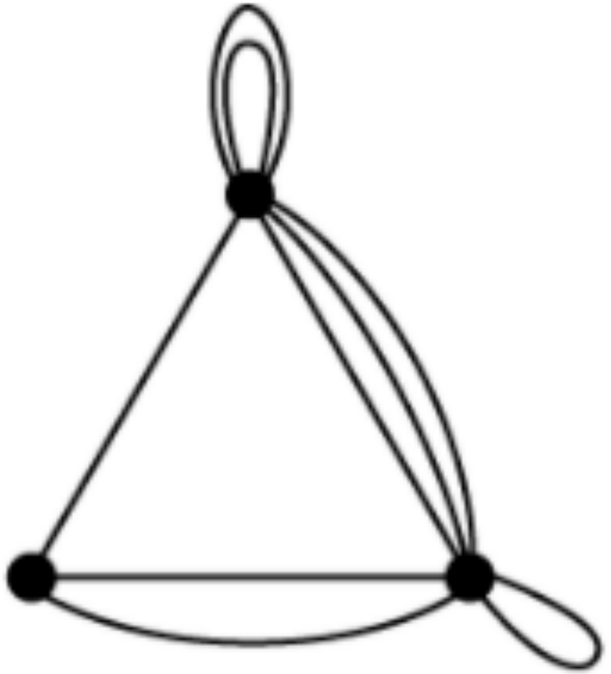
Real spatial networks are usually pseudo-graphs



Simple graph

+ parallel links

= Multigraph



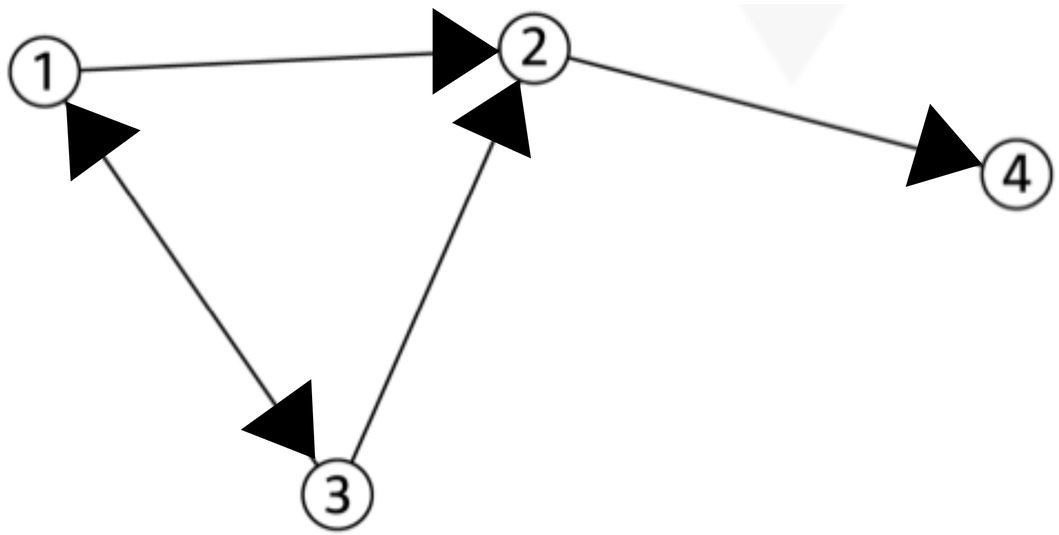
+ self-loops

= Pseudograph

OSMnx calls this MultiGraph

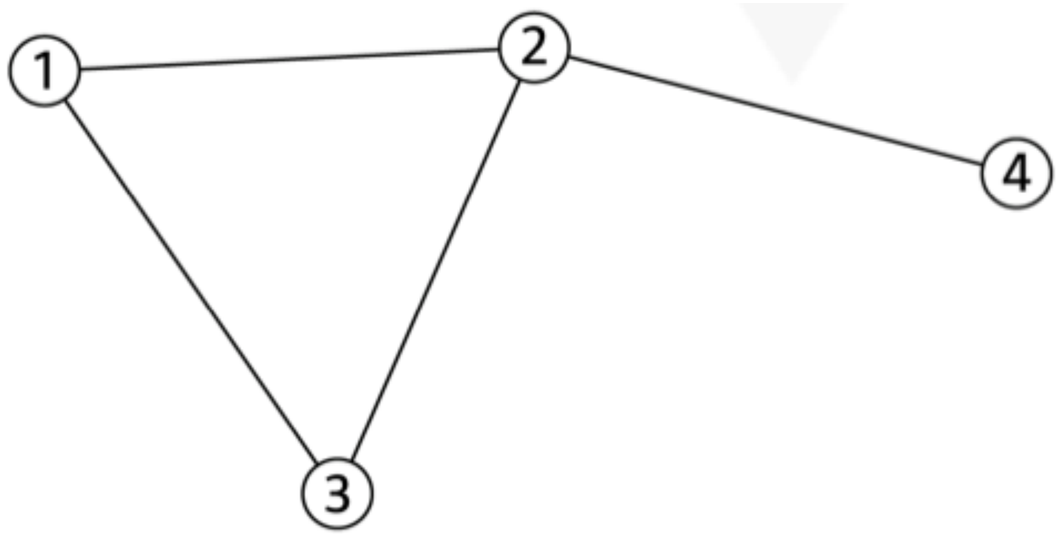
A directed graph (digraph) has links with a direction

Nodes N=4
Links (Arcs) L=5



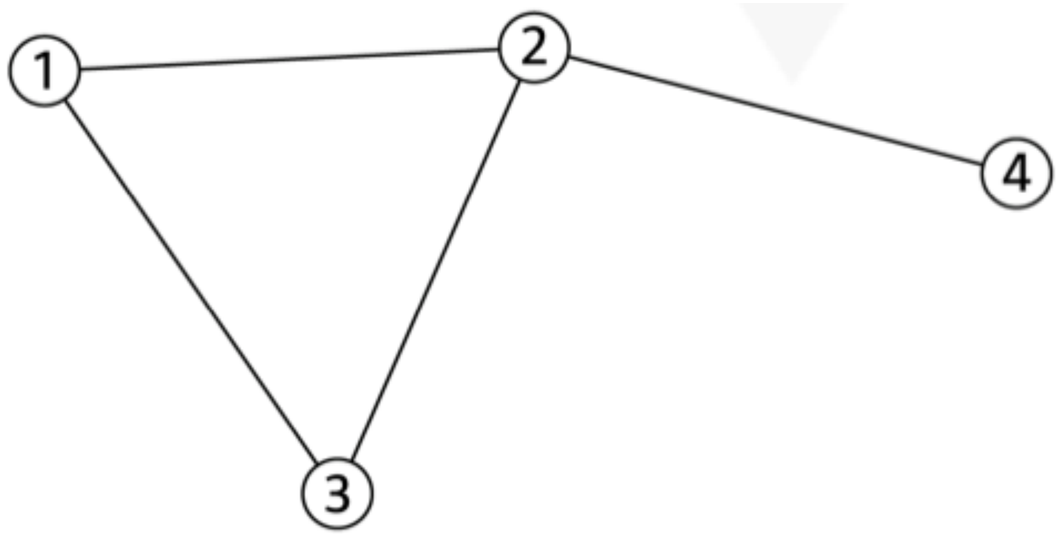
The degree k_i of a node i is the number of incident links

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



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

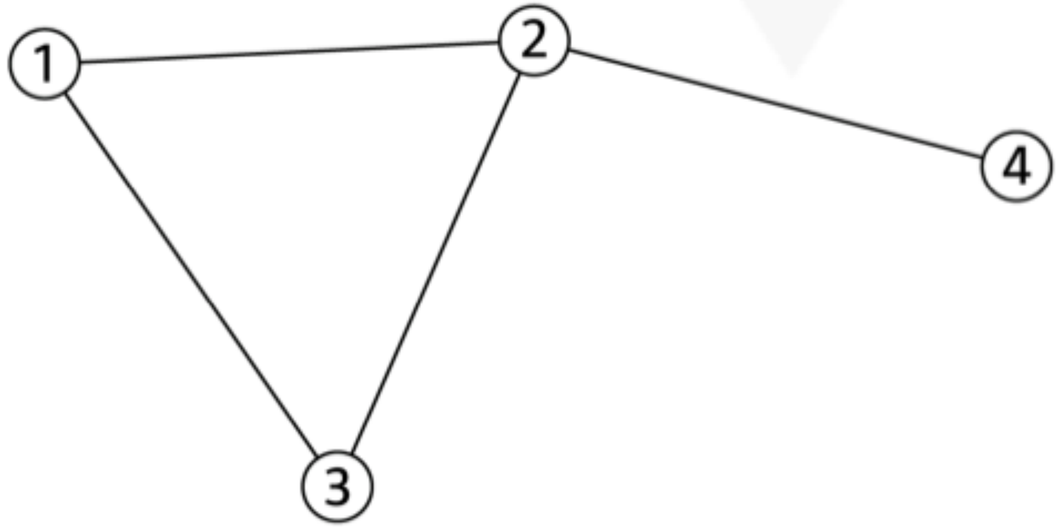
$$k_1 = 2 \qquad k_3 = 2$$
$$k_2 = 3 \qquad k_4 = 1$$

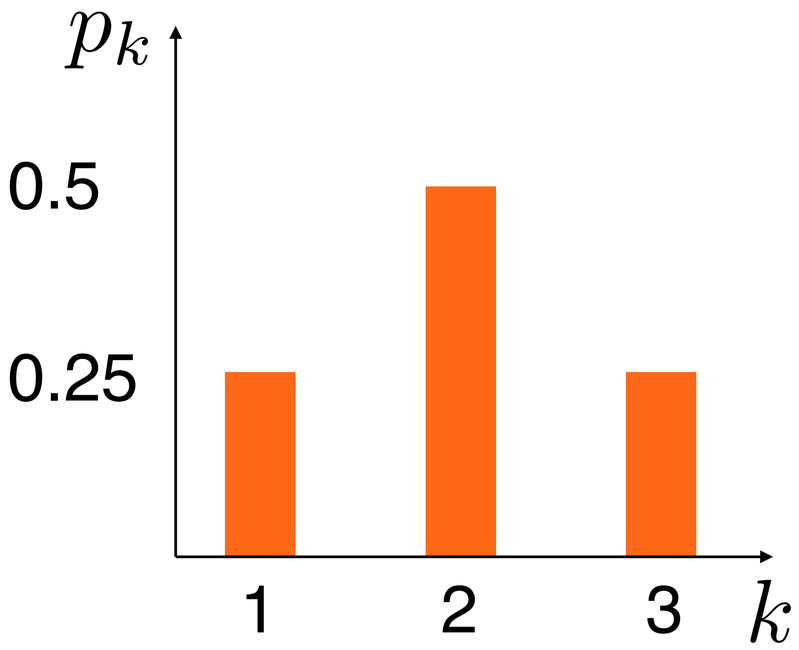


$$\langle k \rangle = \frac{1}{N} \sum_{i=1}^{N} k_i \qquad \langle k \rangle = \frac{2+3+2+1}{4} = 2$$

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

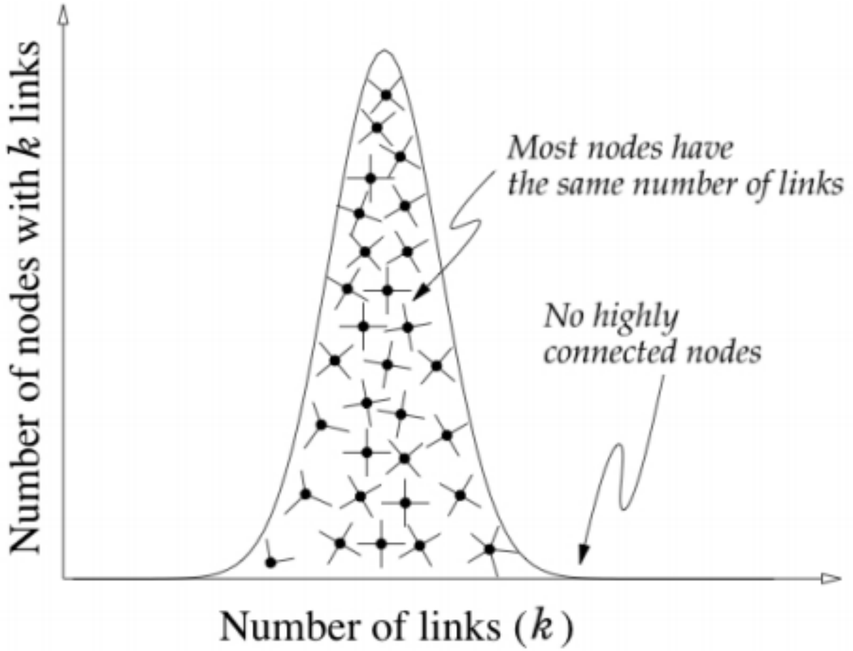
Usually:

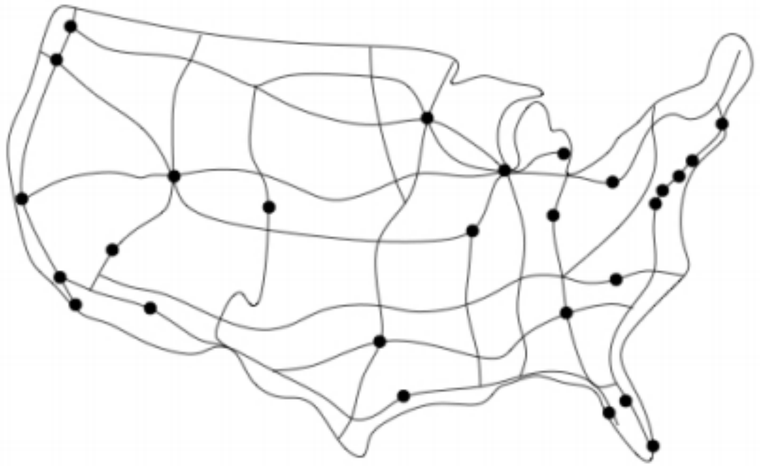
A network's degree distribution tells us something fundamental about how individuals in the system connect

However:

In many spatial networks the degree distribution is far less interesting

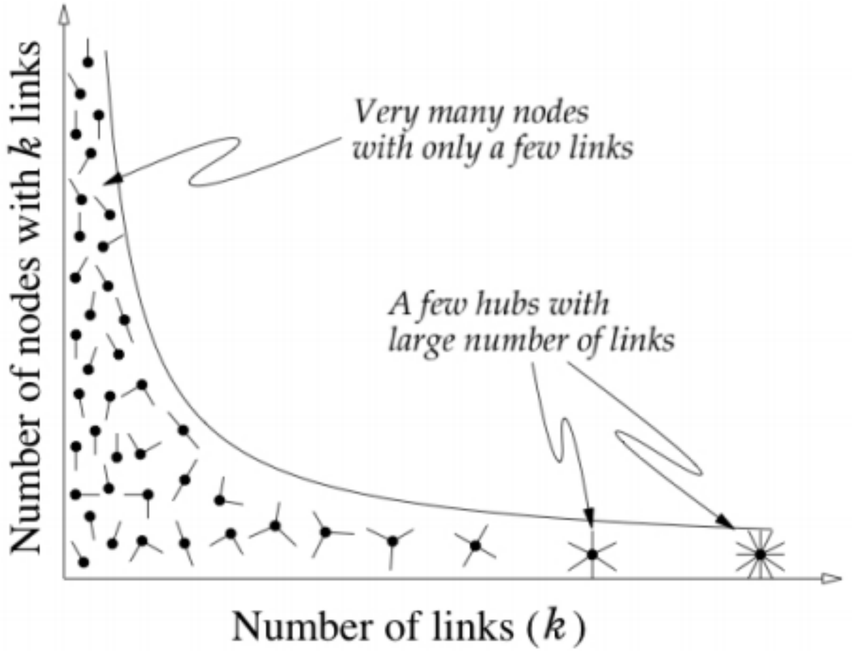
Thin-tailed network

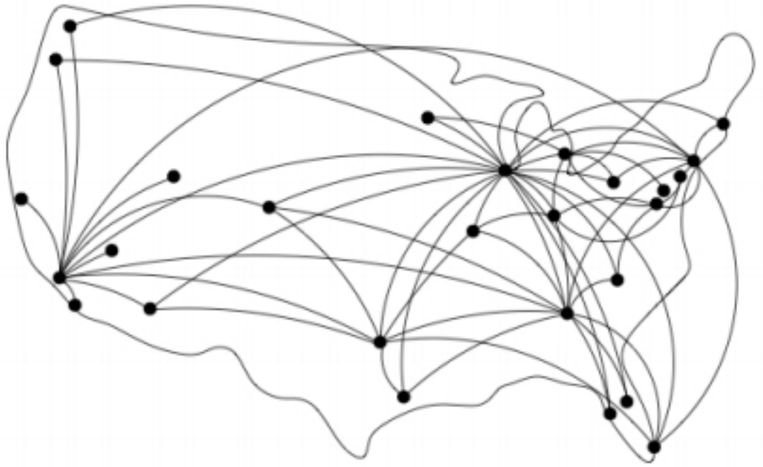




Streets

Heavy-tailed network

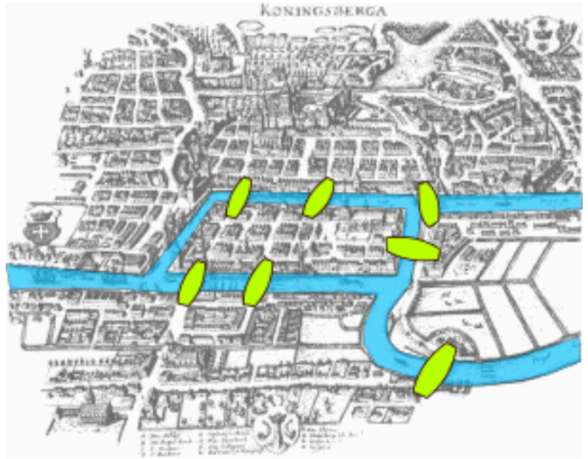


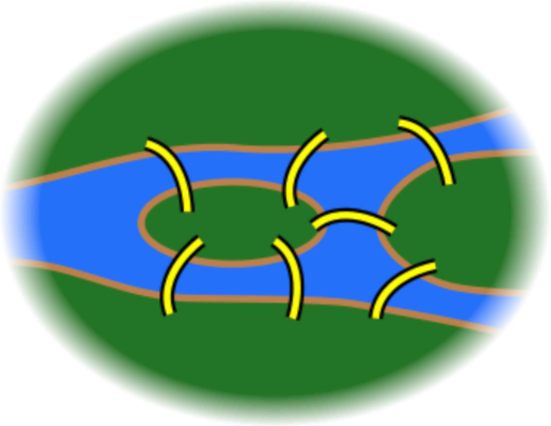


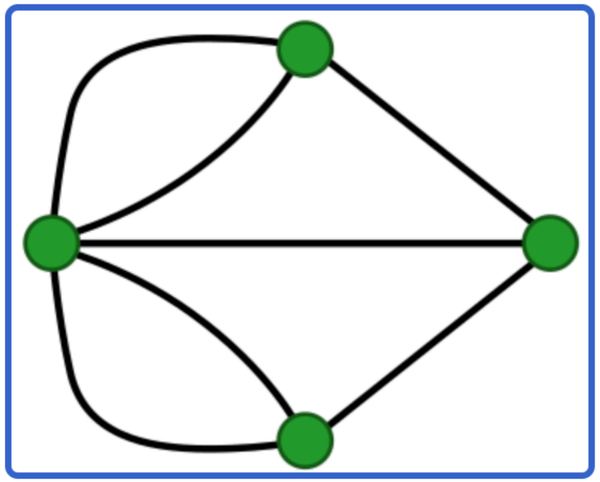
Airlines

Spatial problems central in early graph history

Euler’s Seven Bridges of Königsberg



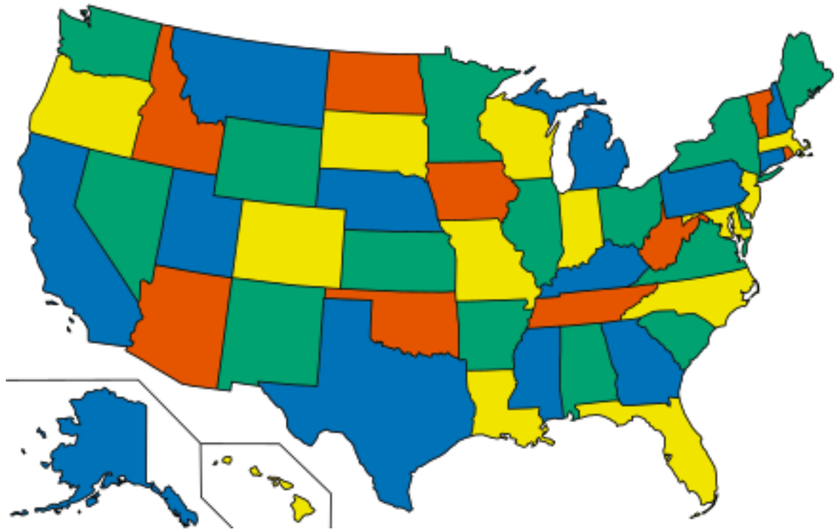


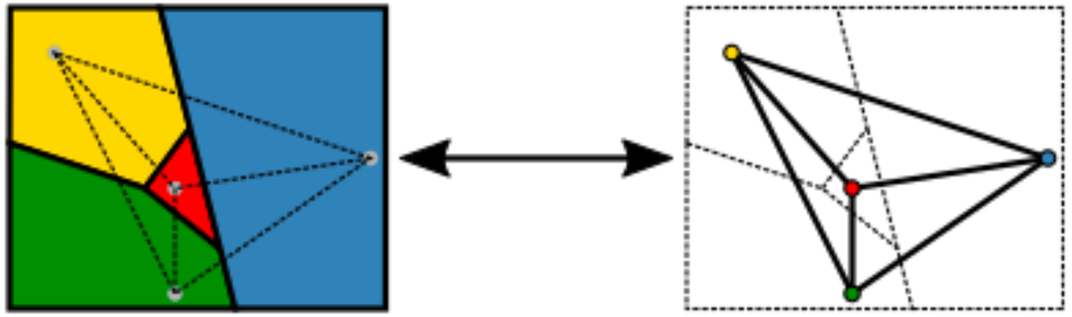


https://spatial-analytics.readthedocs.io/en/latest/

Spatial problems central in early graph history

The ‘Four color theorem’

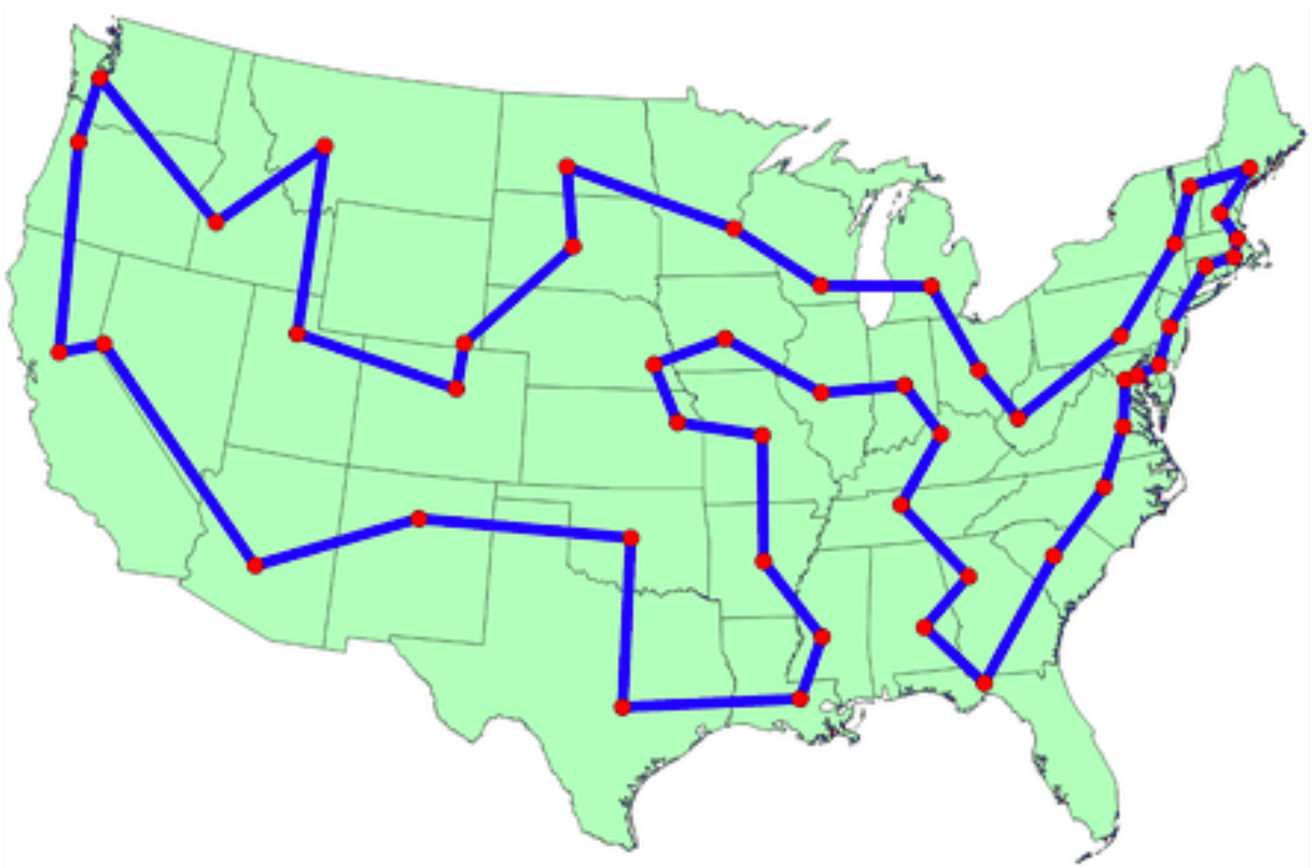




https://spatial-analytics.readthedocs.io/en/latest/

Spatial problems central in early graph history

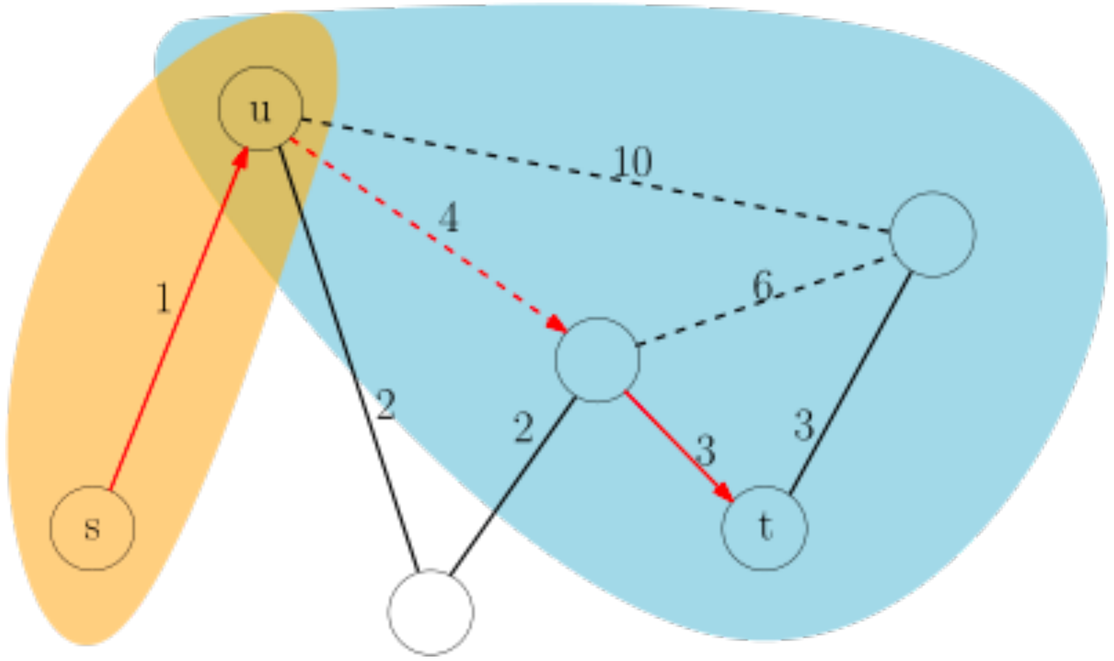
Traveling salesman problem



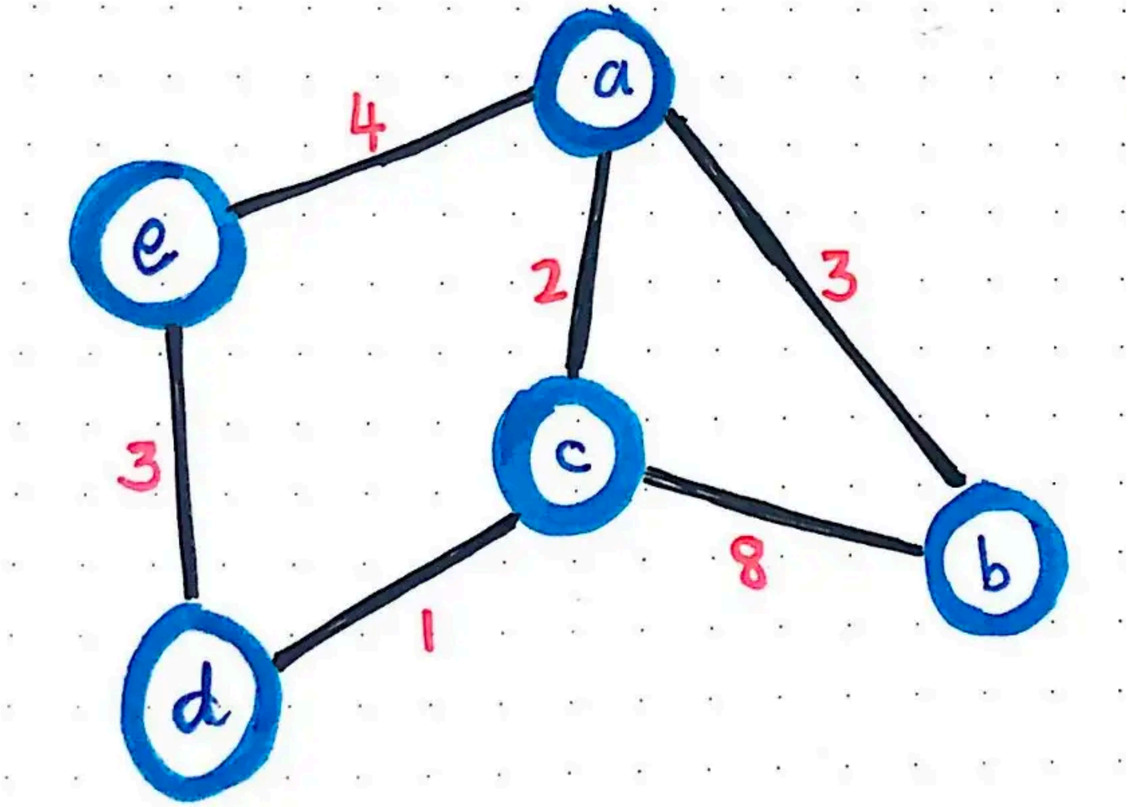
https://spatial-analytics.readthedocs.io/en/latest/

We use familiar algorithms for traversal & shortest paths

Contraction hierarchies



Dijkstra’s shortest path



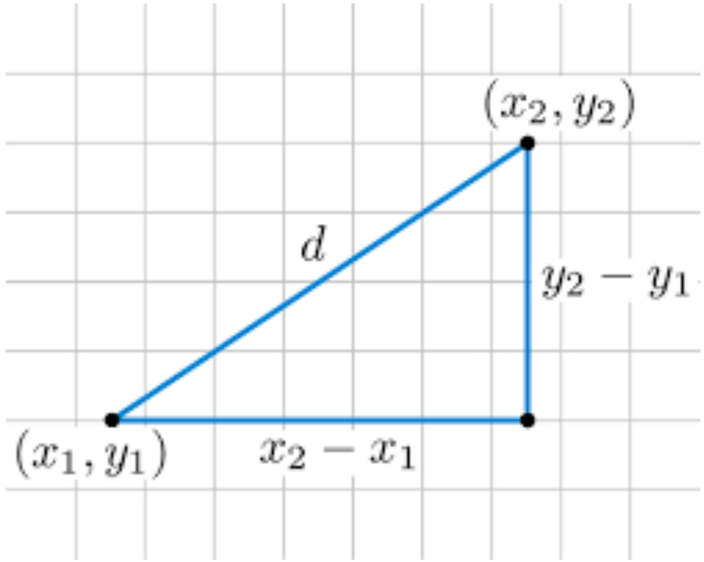
“Spatial is special”

Spatial networks have nodes and (potentially) links that are spatially embedded

A spatial network has nodes in a metric space

You can measure a distance between nodes

In practice: 2D-space with Euclidian distance



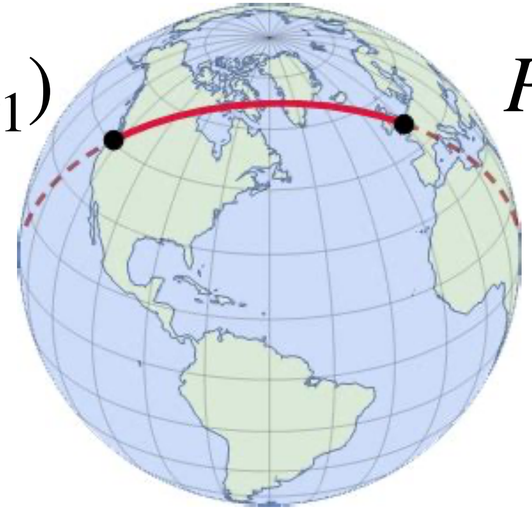
$$d(P_1, P_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

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

$P_1 = (\varphi_1, \lambda_1)$



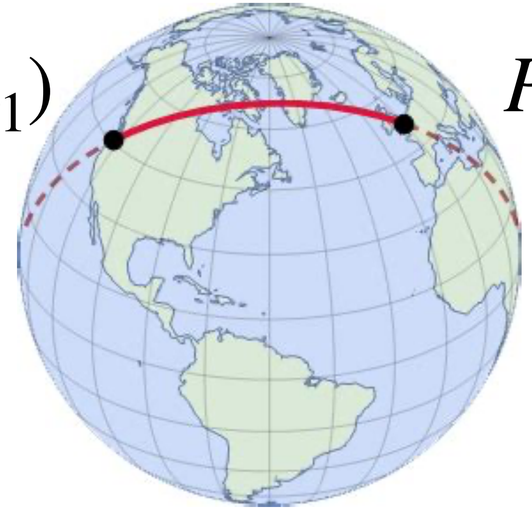
$P_2 = (\varphi_2, \lambda_2)$

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

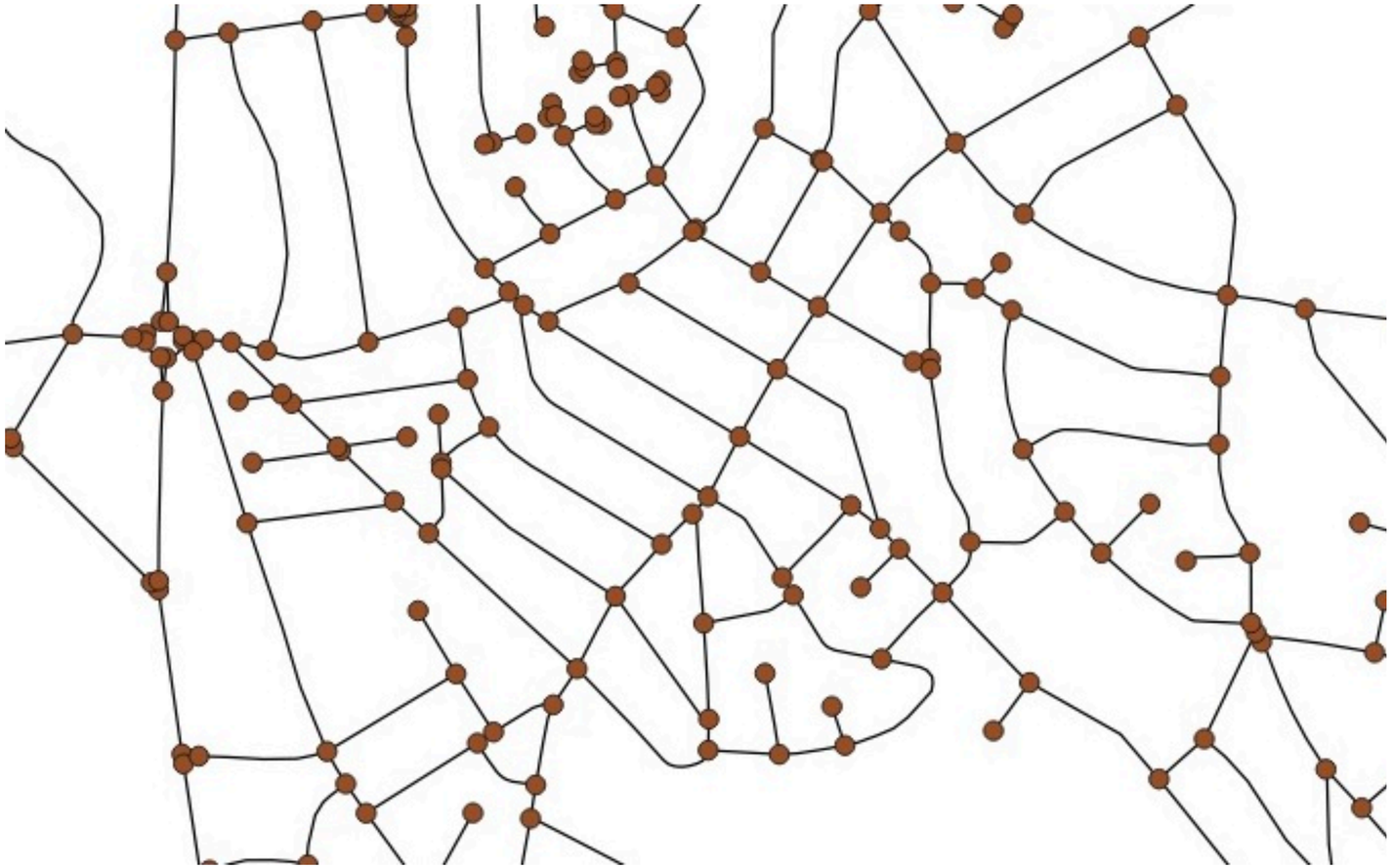
$P_1 = (\varphi_1, \lambda_1)$

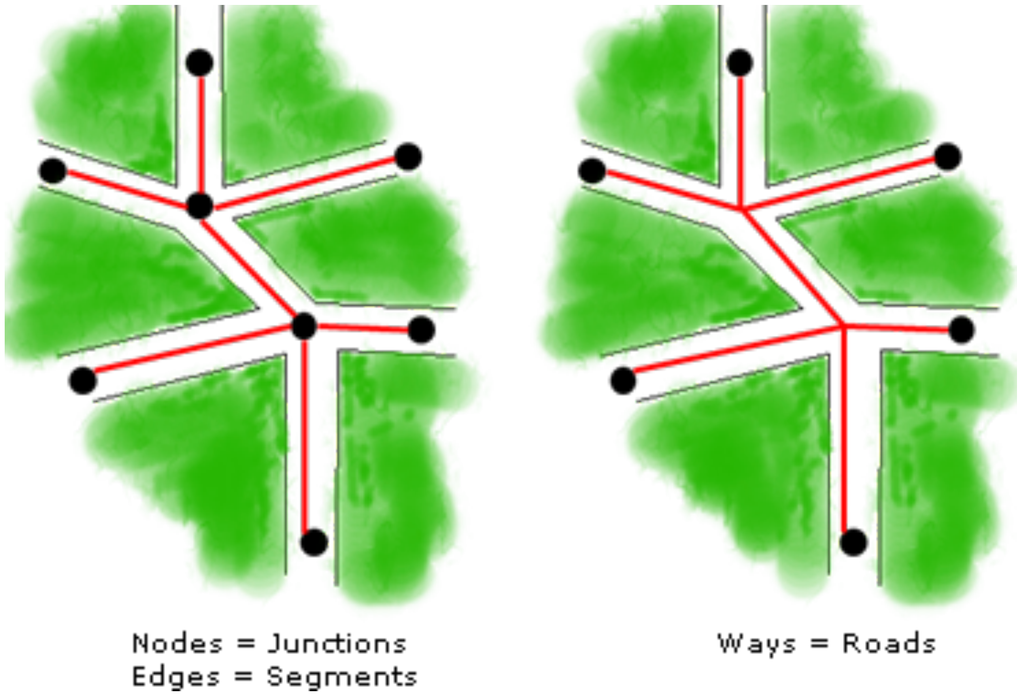


$P_2 = (\varphi_2, \lambda_2)$

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

A spatial network often has edges in a metric space





https://gicycle.wordpress.com/tag/osm/

Network of situated nodes (airports)

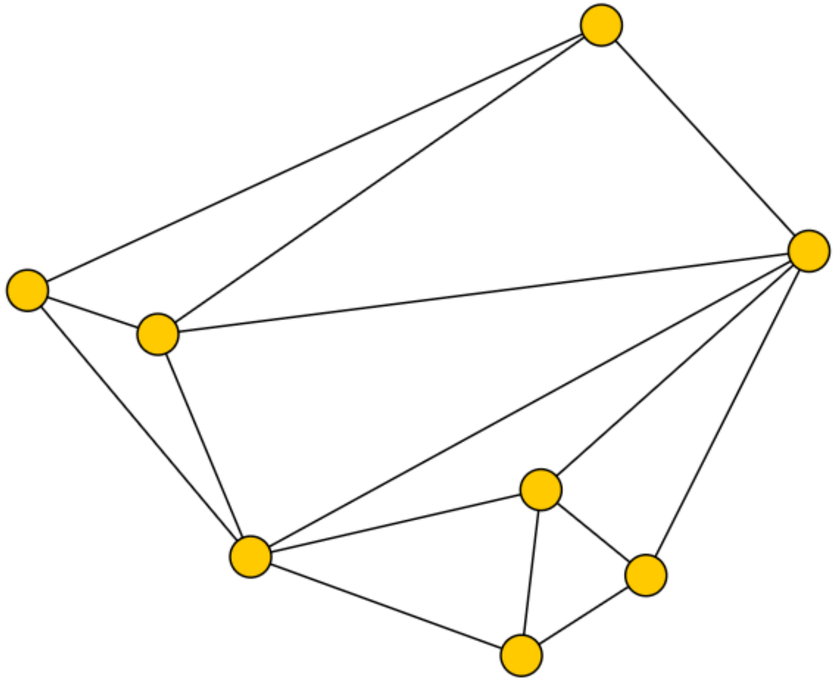
https://studentwork.prattsi.org/blog/2018/05/11/visualizations-of-the-global-flights-network/

(Geo)social network

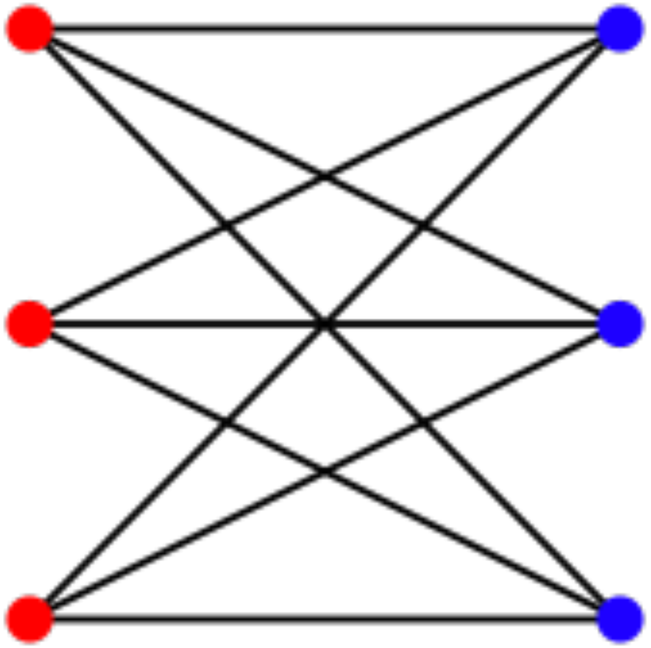
https://spatial-analytics.readthedocs.io/en/latest/

Spatial networks are often (approximately) planar

Planar



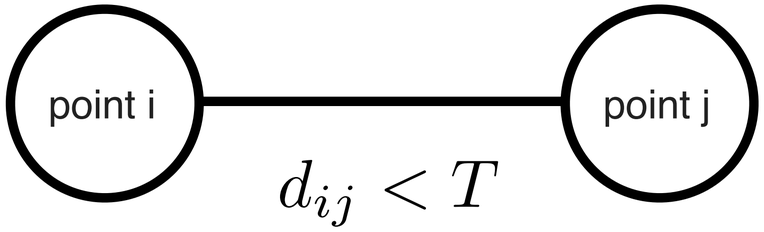
Not planar

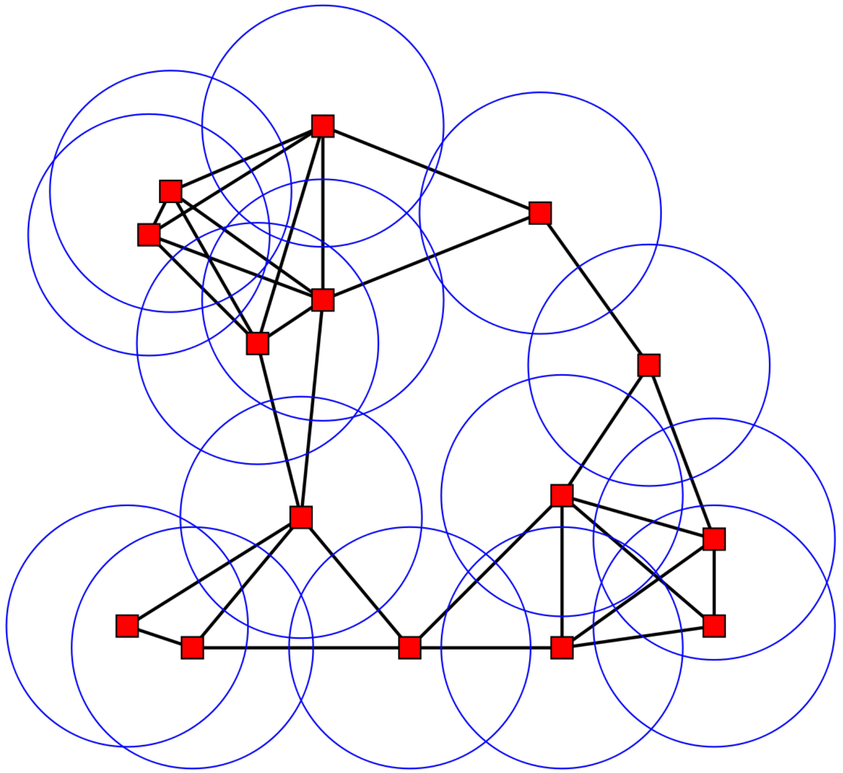


https://www.jasondavies.com/planarity/

Spatial network models

Model: Random geometric graph





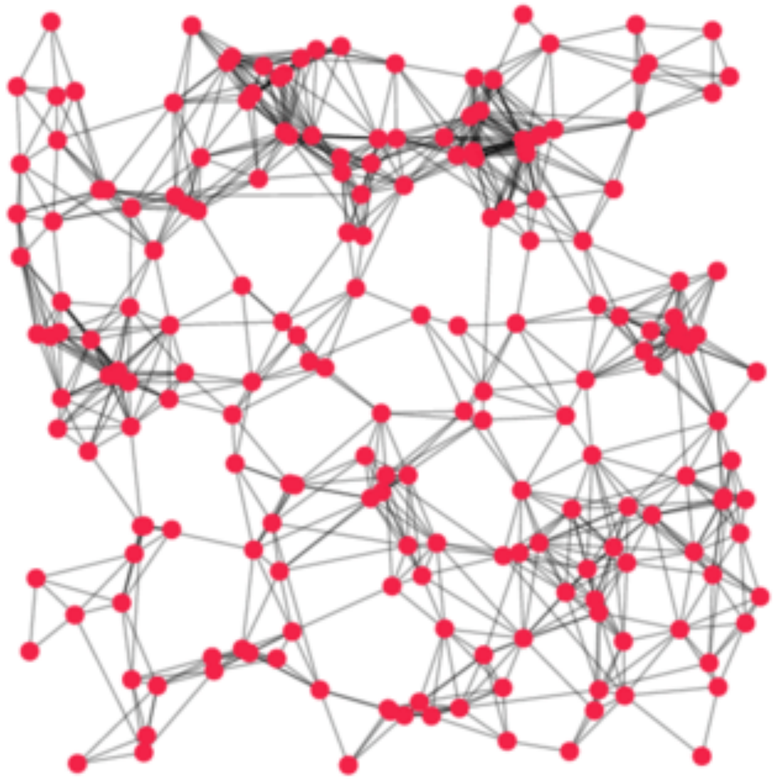
distance-based

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

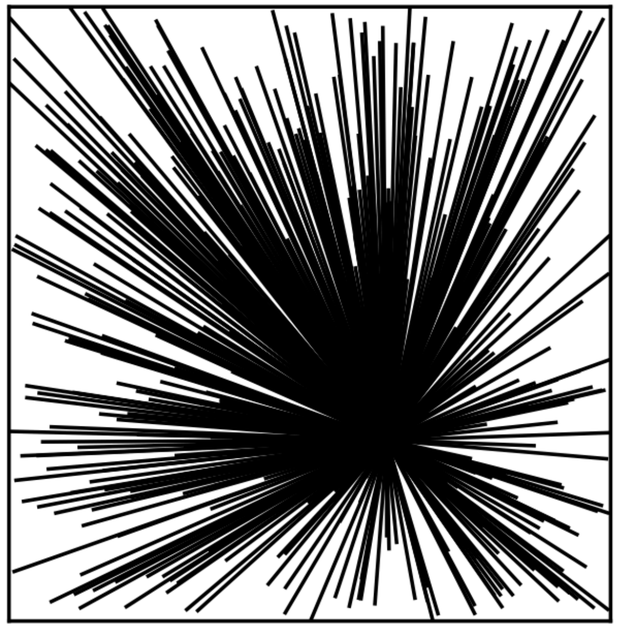
$$P(i,j) = \beta e^{d(i,j)/d_0}$$

Used to model the topology of the internet.

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

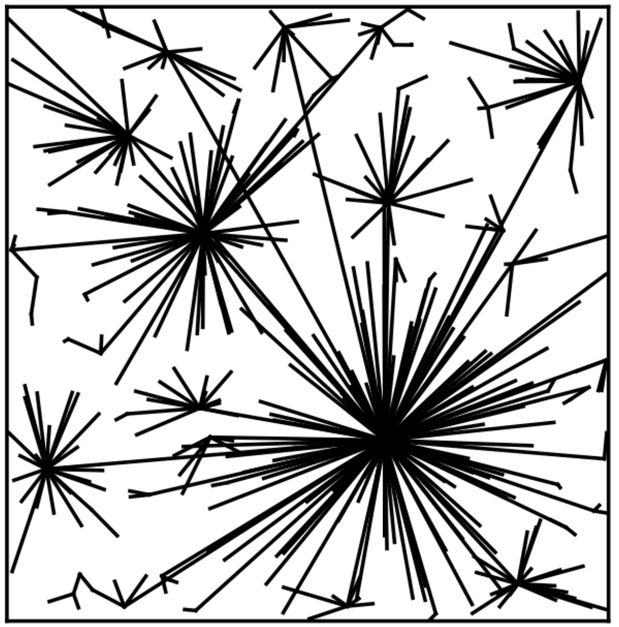
Model: Hierarchical

$\beta/\beta^* = 0.0$



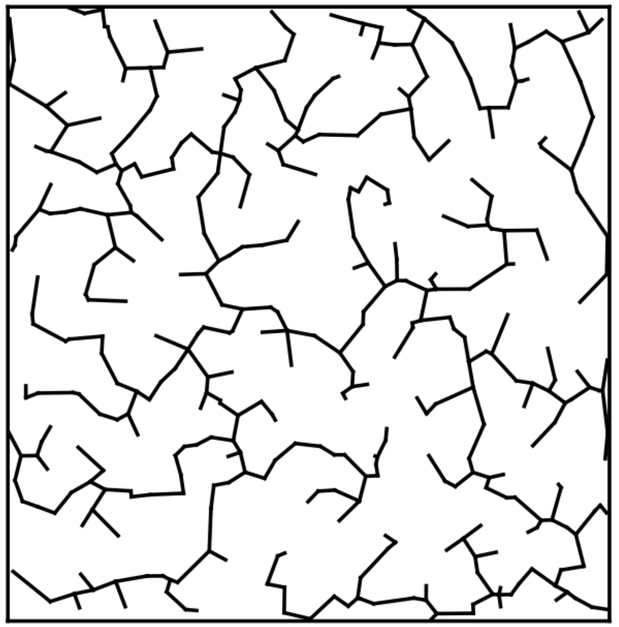
Star graph

$\beta/\beta^* = 1.0$



Spatial hierarchy

$\beta/\beta^* = 1000.0$



Minimum spanning tree

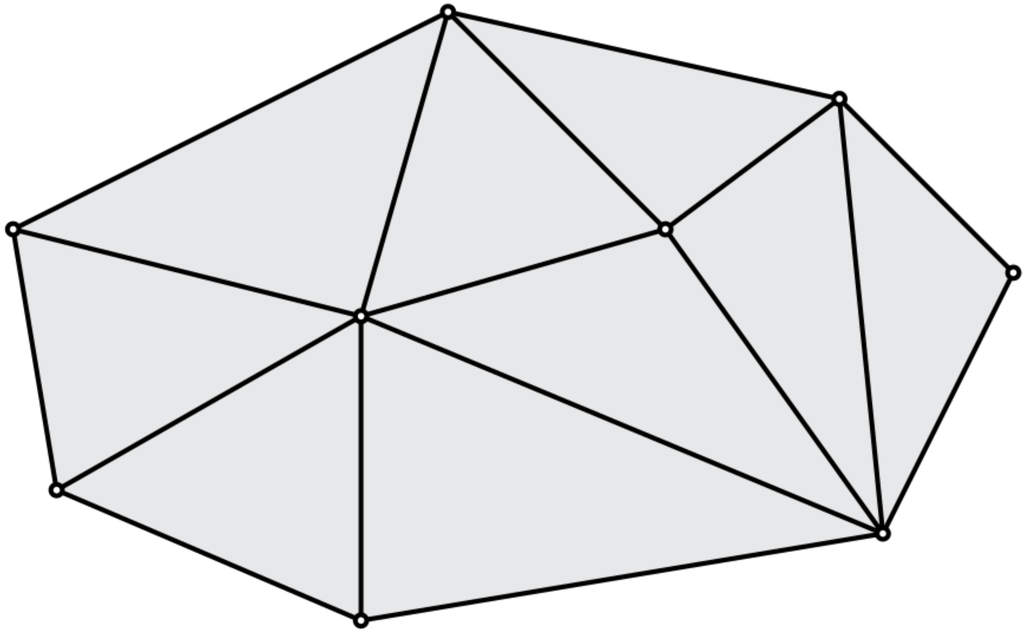
The average detour
is minimal here

Model railway networks
between cities

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

Optimal networks

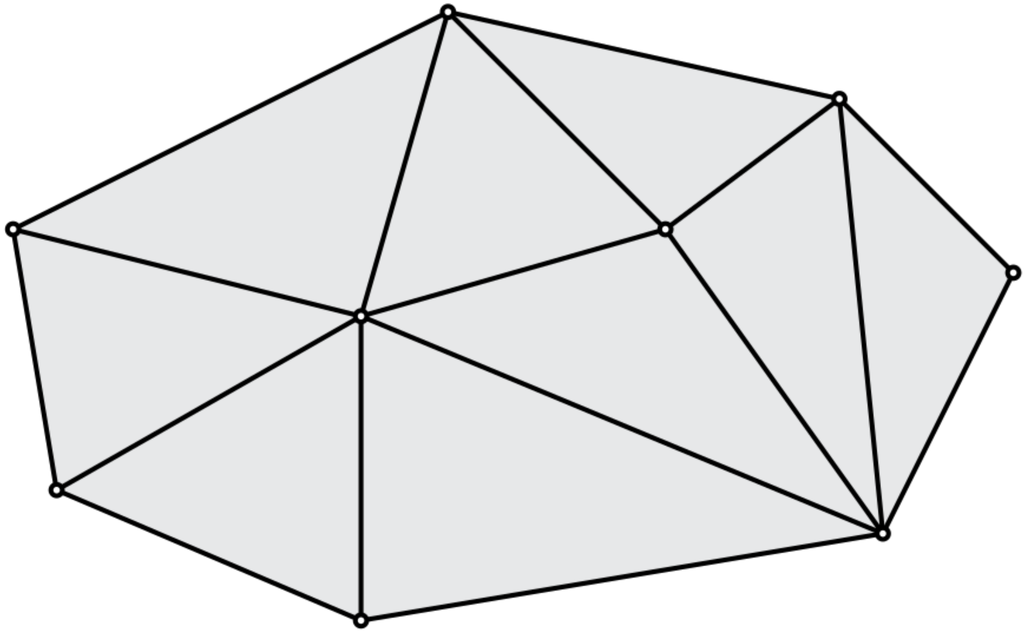
A triangulation of 2D points is a maximal set of non-crossing edges



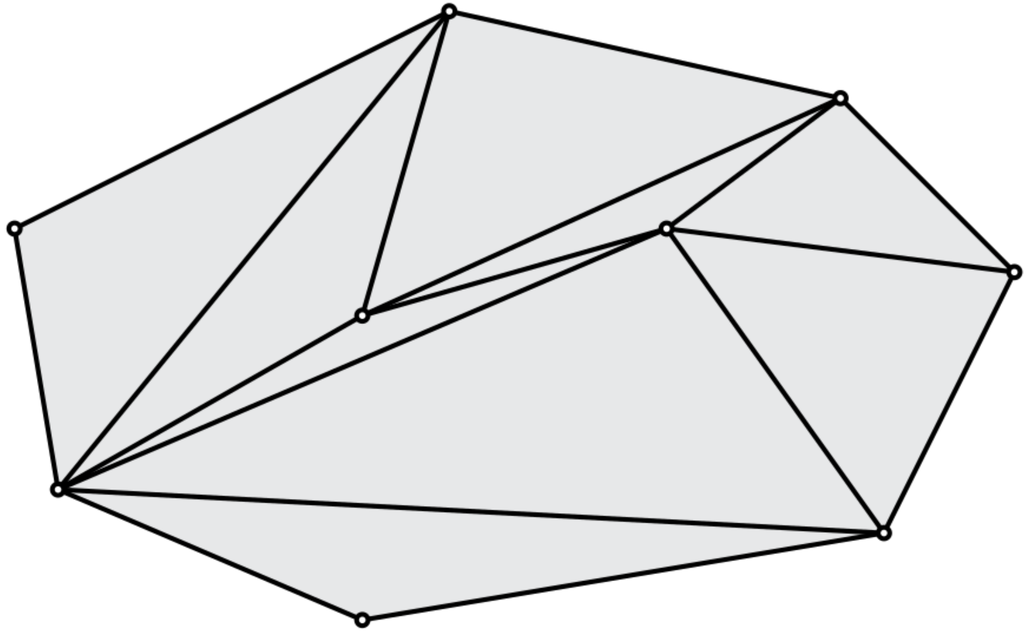
It is planar

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

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



It is planar

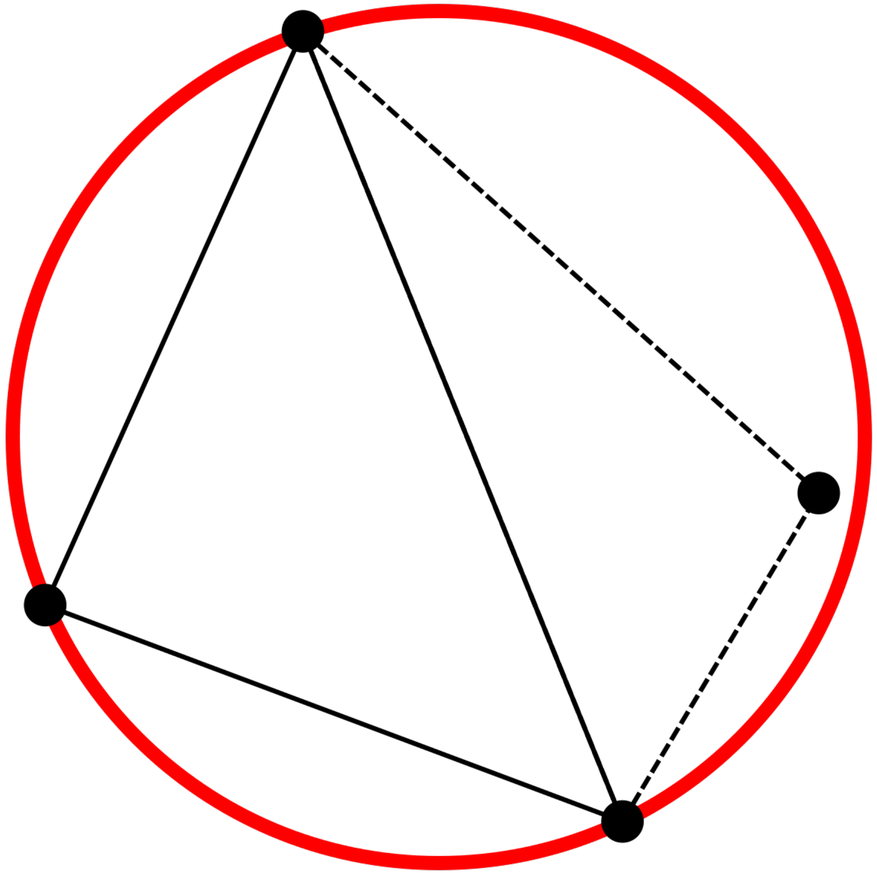


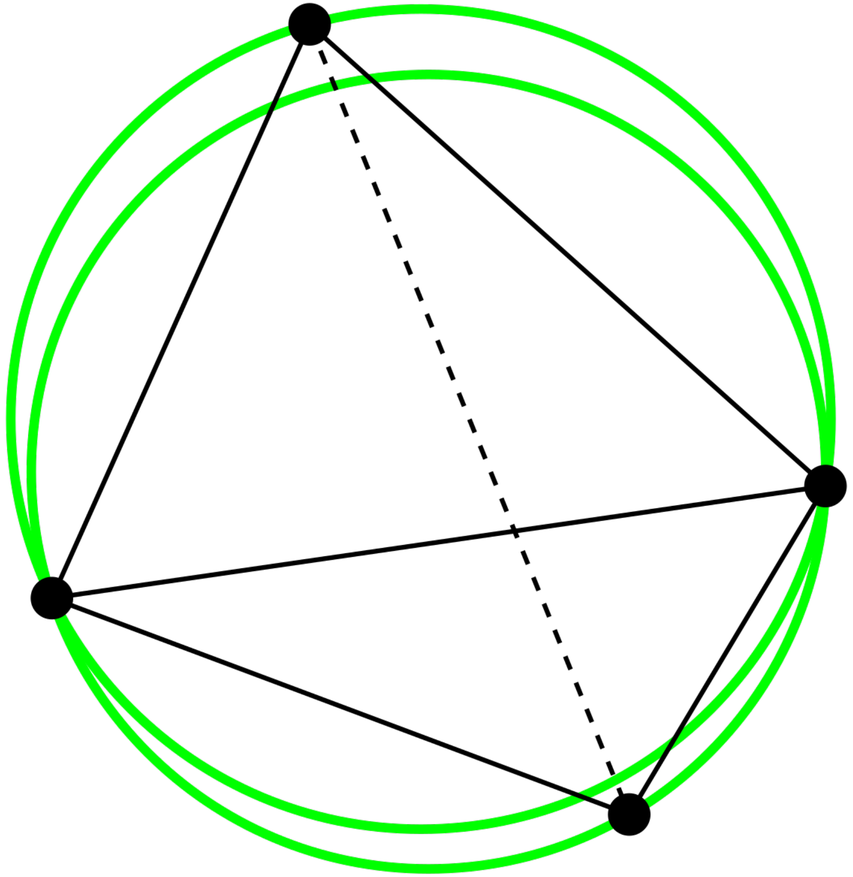
It is not unique

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

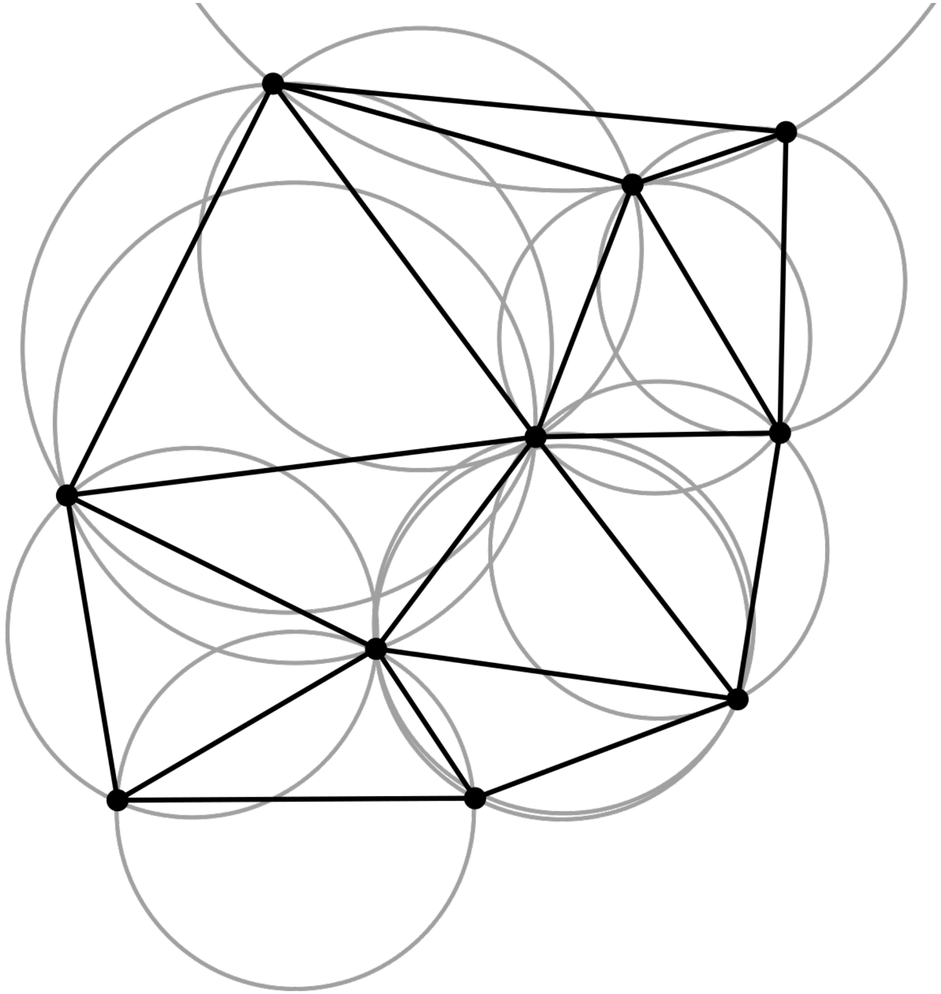


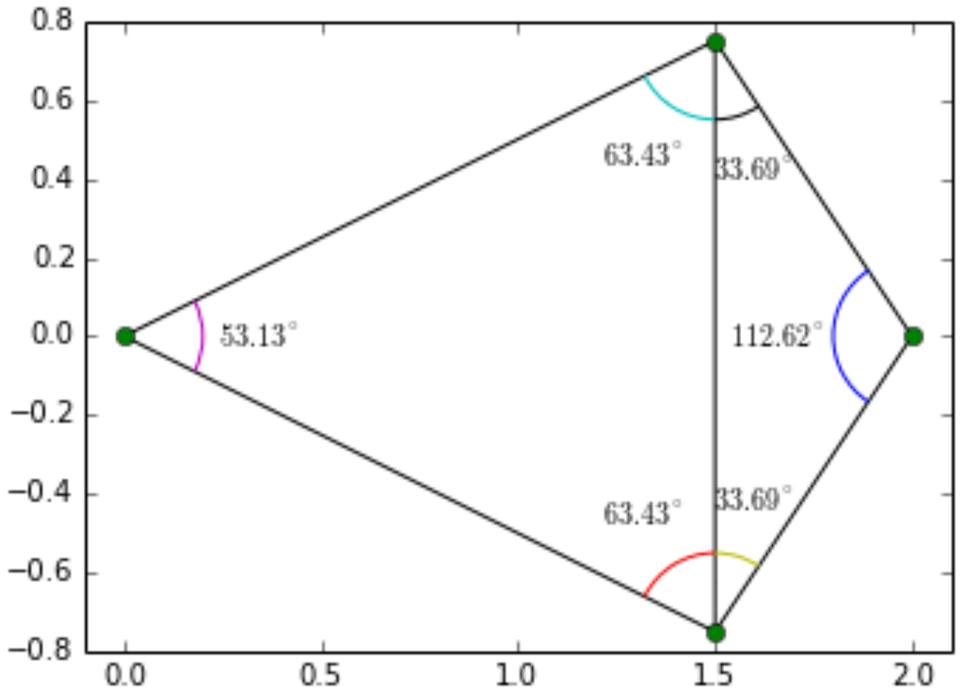


https://en.wikipedia.org/wiki/Delaunay_triangulation

A Delaunay triangulation maximizes the minimum of all angles

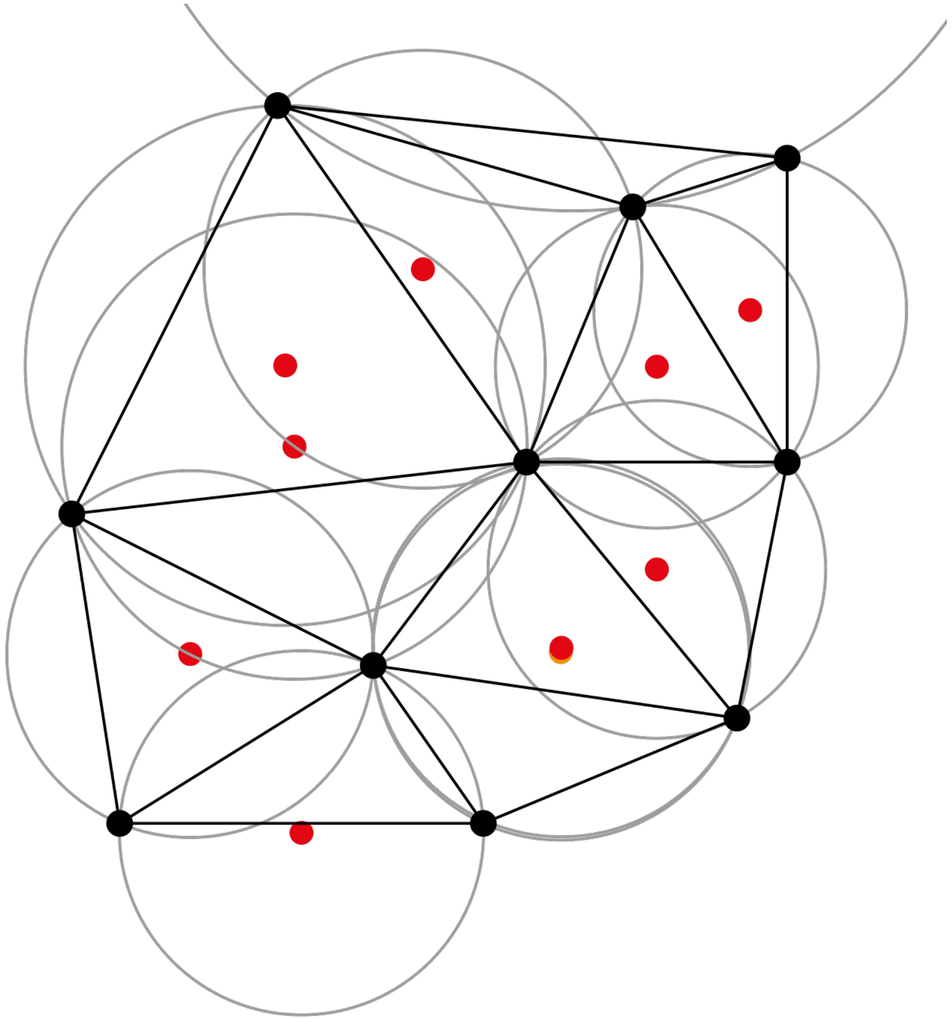
No point can be inside the circumcircle of any triangle

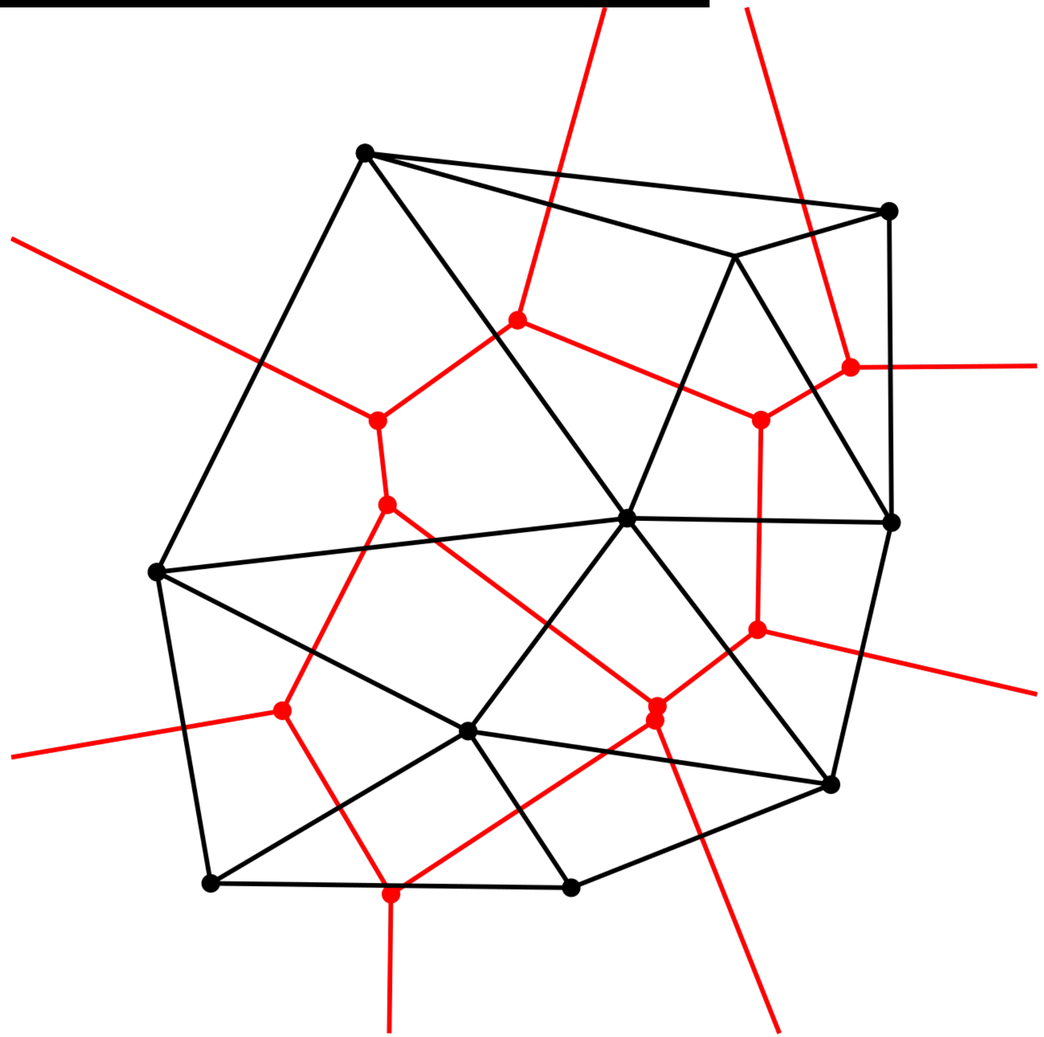




https://en.wikipedia.org/wiki/Delaunay_triangulation

A Delaunay triangulation maximizes the minimum of all angles

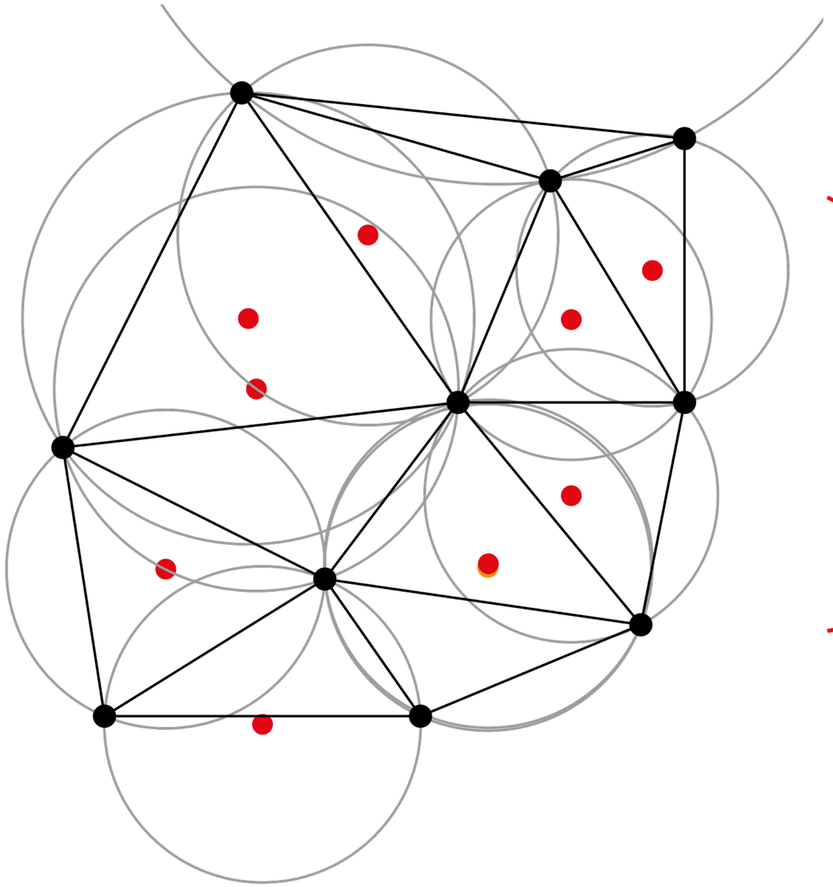


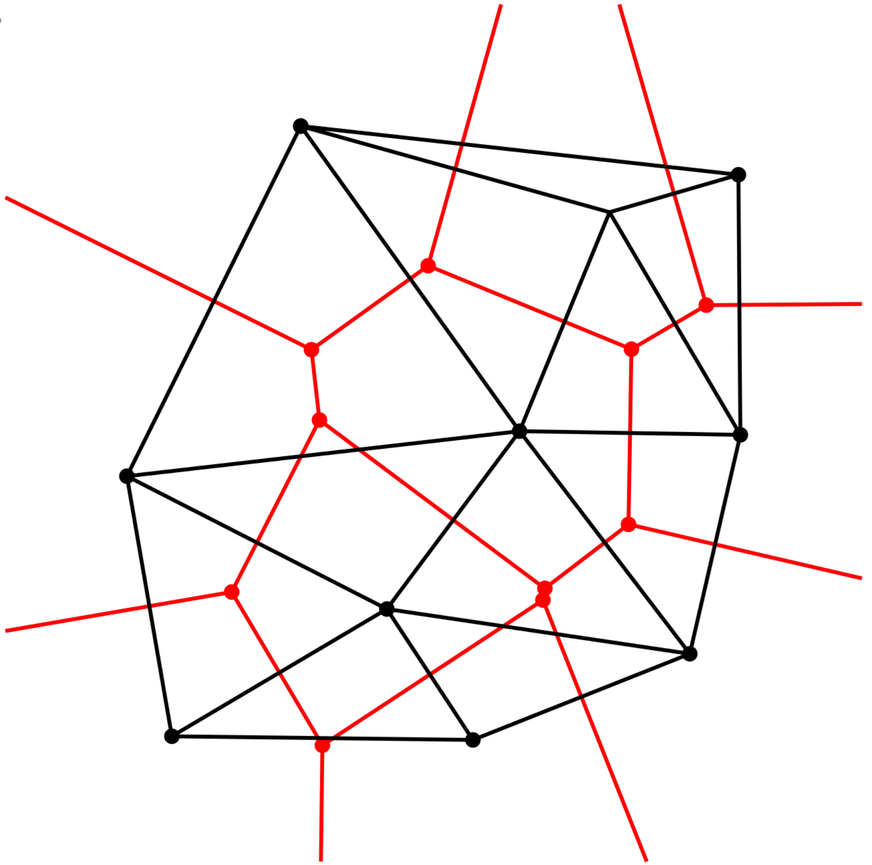


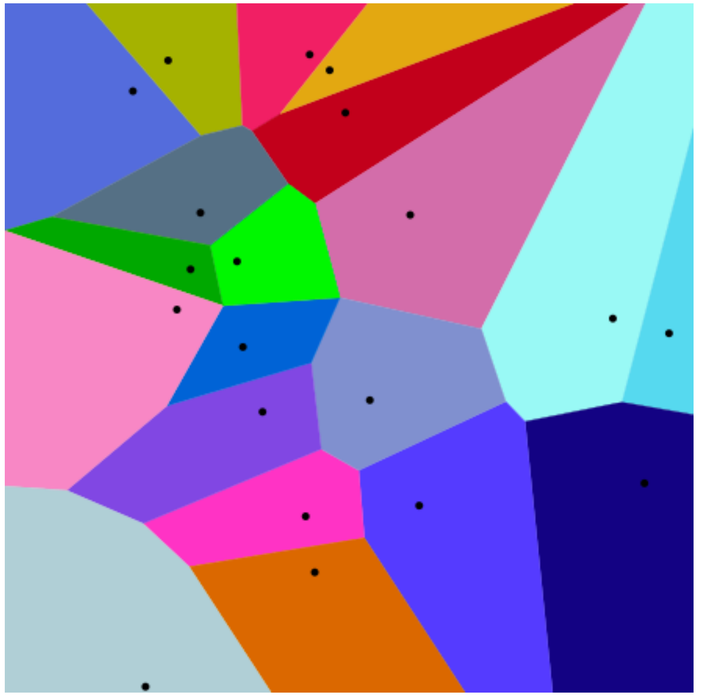
It is the dual graph of the Voronoi diagram

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

A Delaunay triangulation maximizes the minimum of all angles



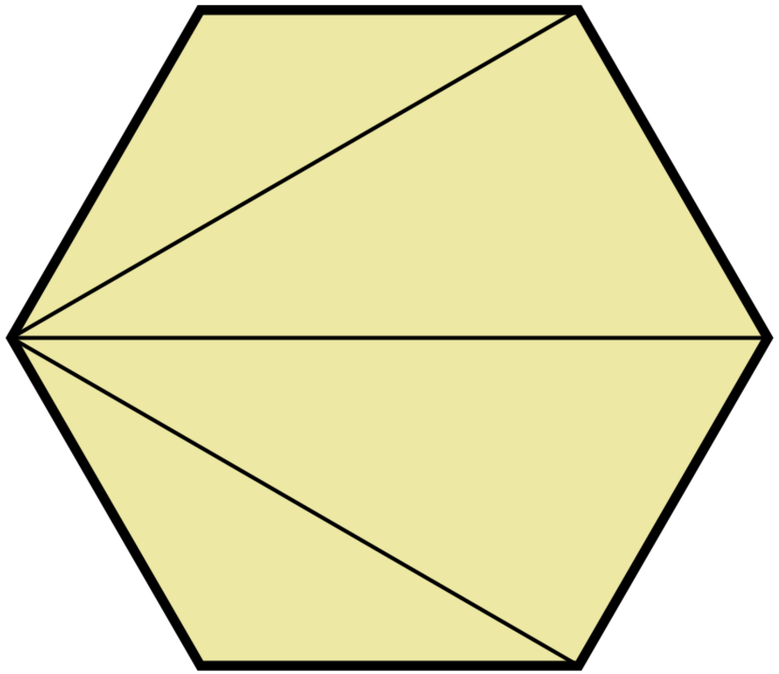


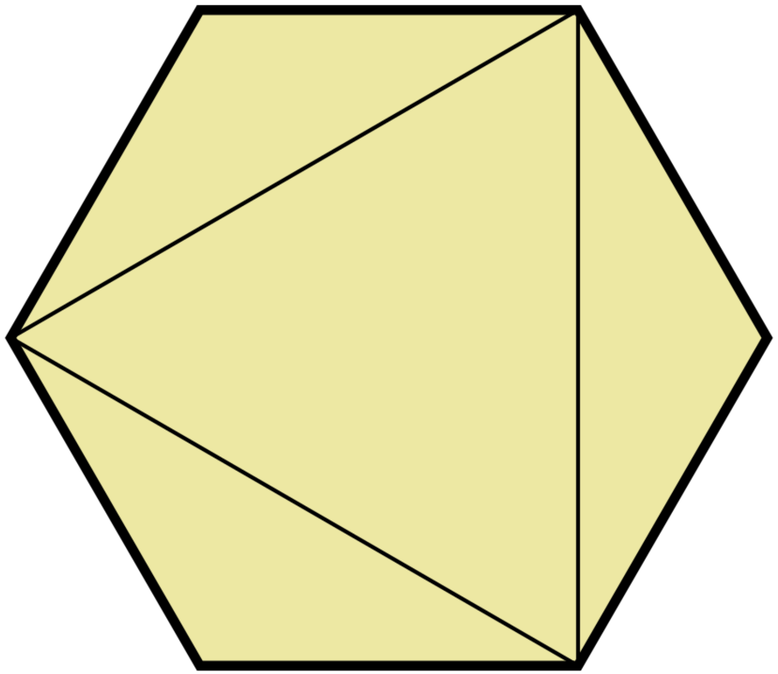


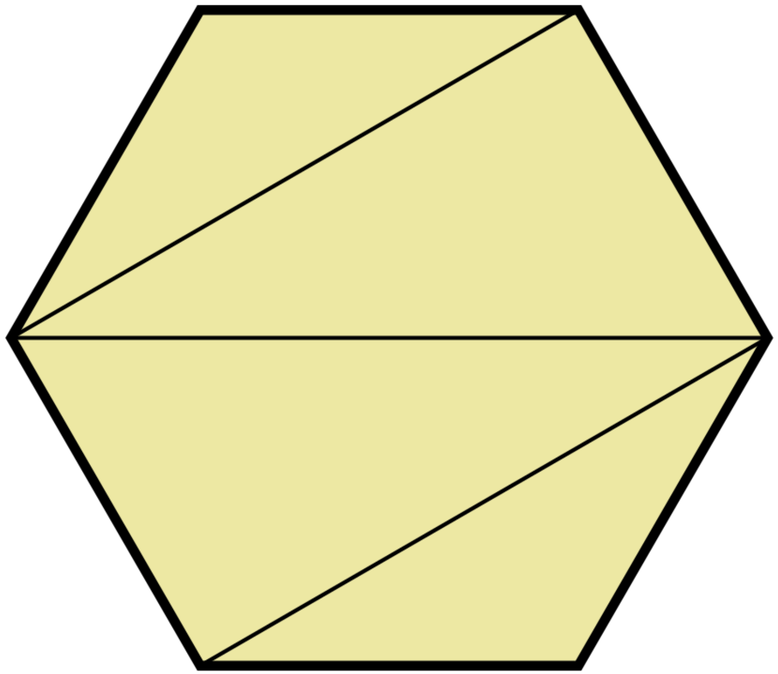
It is the dual graph of the Voronoi diagram

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

A minimum weight triangulation minimizes sum of edge lengths







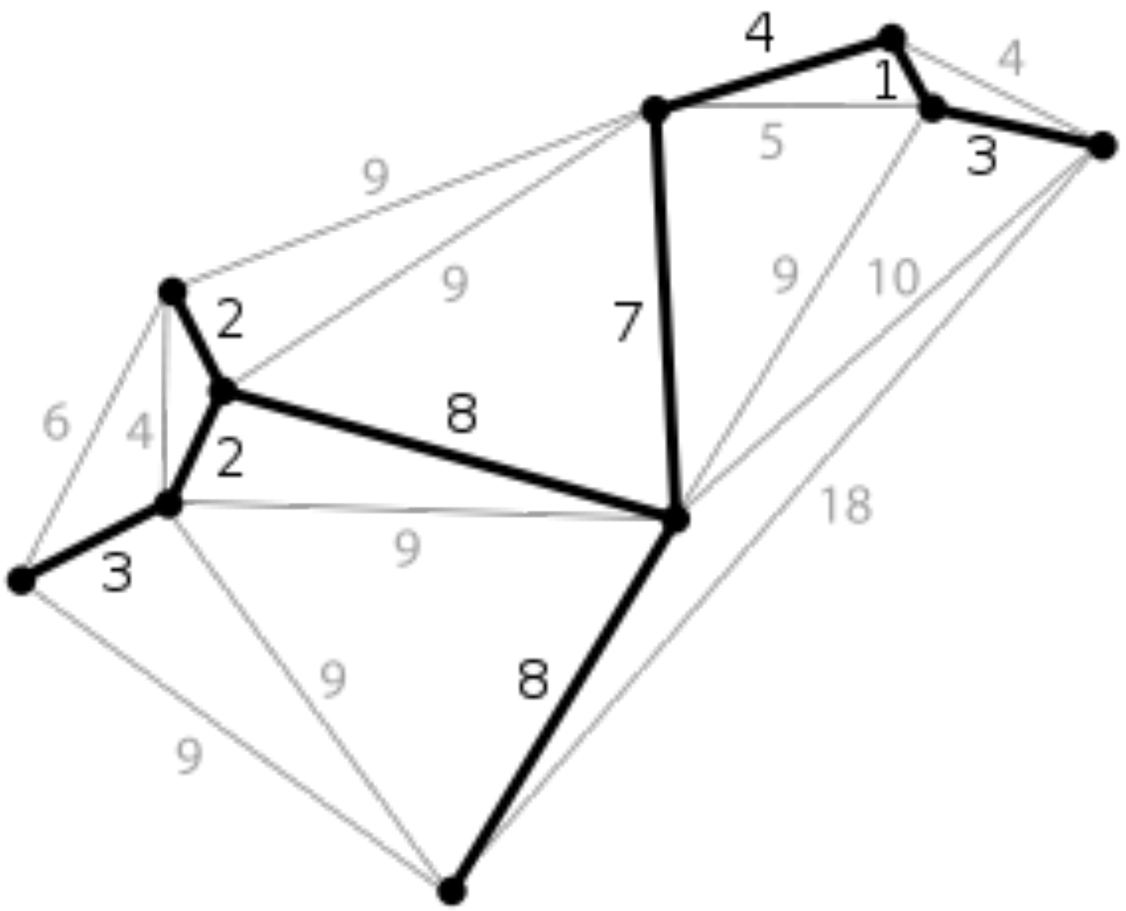
https://en.wikipedia.org/wiki/Minimum-weight_triangulation

Many other Point set triangulations exist

		minimize	maximize
minimum	angle		$O(n \log n)$ (Delaunay triangulation)
maximum		$O(n^2 \log n)$ [8] [9]	
minimum	area	$O(n^2)$ [10]	$O(n^2 \log n)$ [11]
maximum		$O(n^2 \log n)$ [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		$O(n^2)$ [14]	$O(n \log n)$ (using the Convex hull)
sum of		NP-hard (Minimum-weight triangulation)	
minimum	height		$O(n^2 \log n)$ [9]
maximum	slope	$O(n^3)$ [9]	

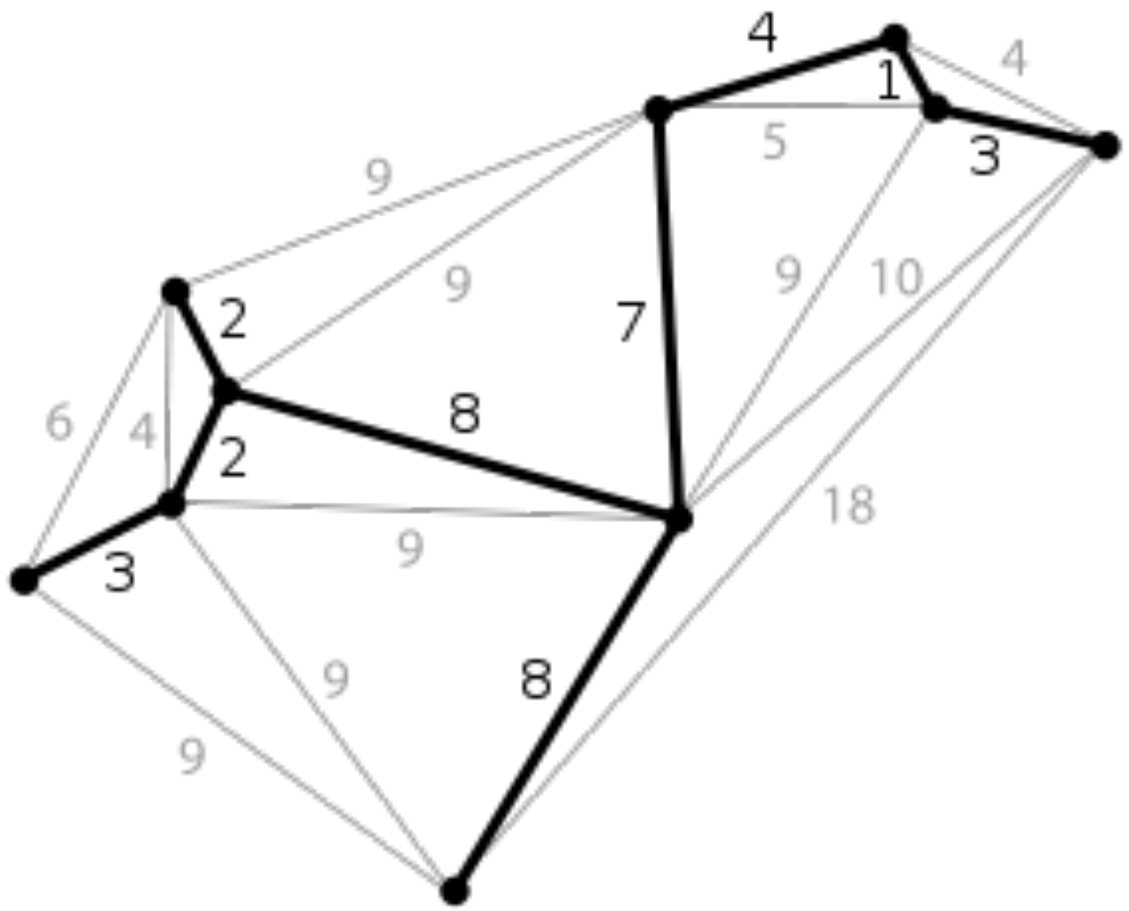
https://en.wikipedia.org/wiki/Point-set_triangulation

Spanning Trees



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

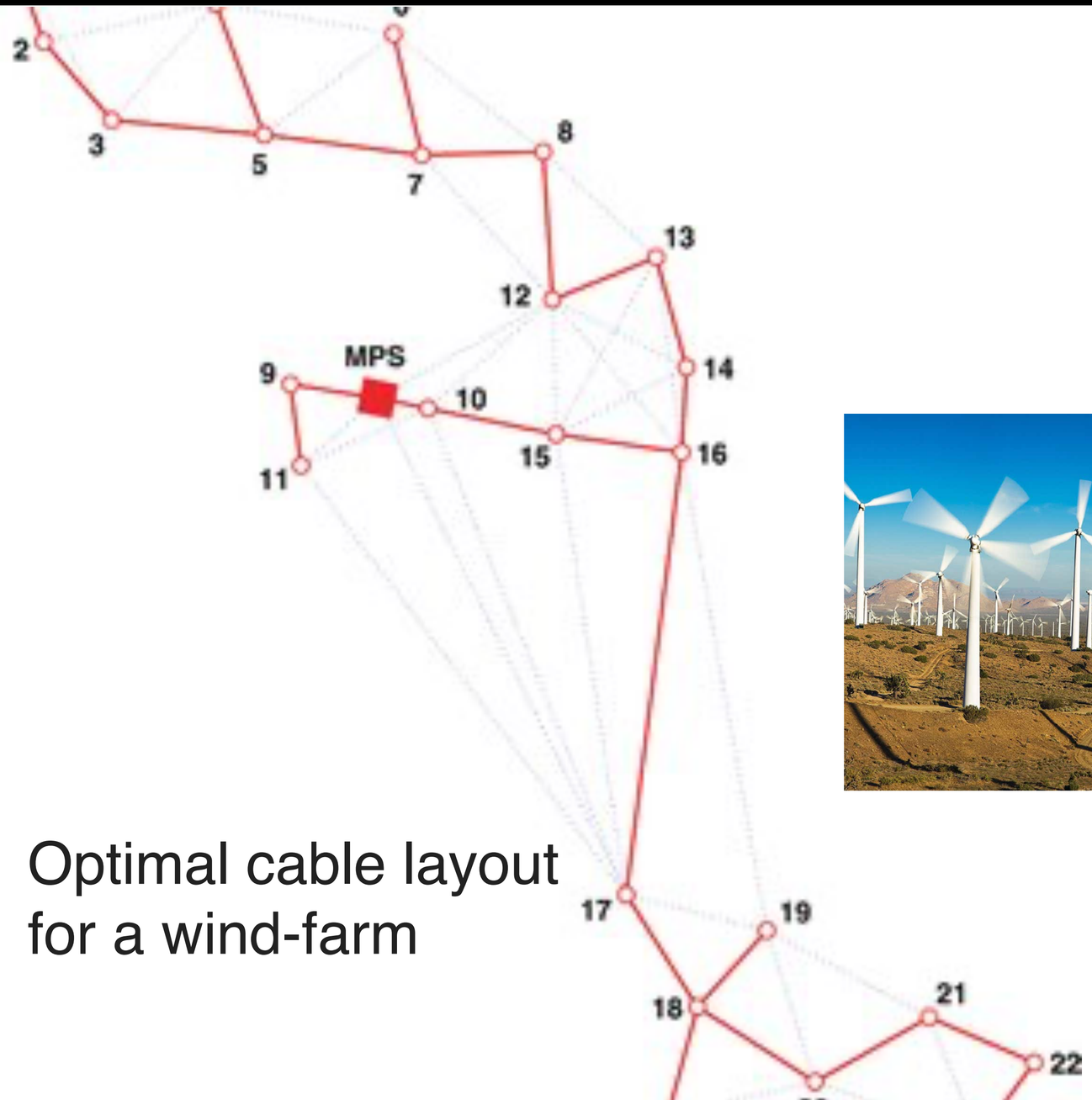
The MST is a subgraph that connects all nodes optimally



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

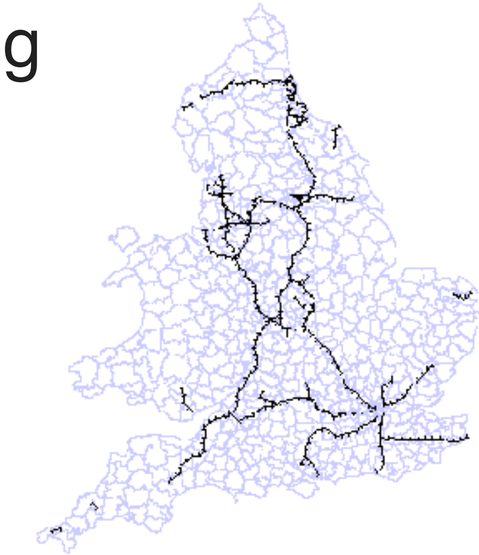
A minimum spanning tree (MST) is a spanning tree such that the sum of its link lengths is smaller than or equal to the sum of the link lengths of any other spanning tree.

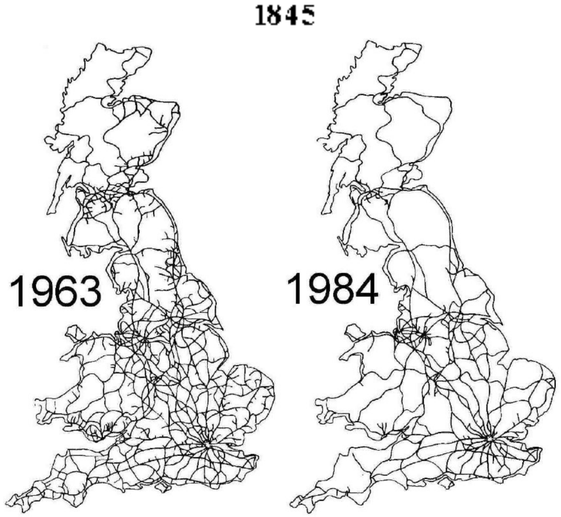
The MST has many applications for spatial networks



Optimal cable layout for a wind-farm

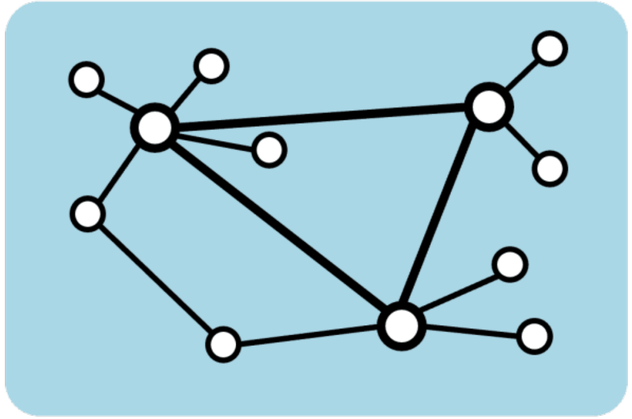
Railroad network planning





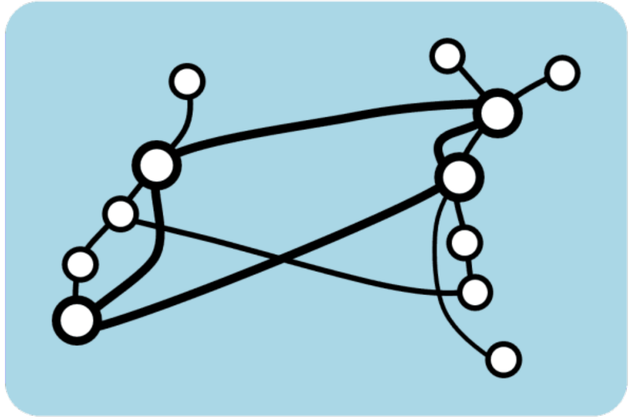
Spatial networks have different shapes and vulnerabilities

Air Networks



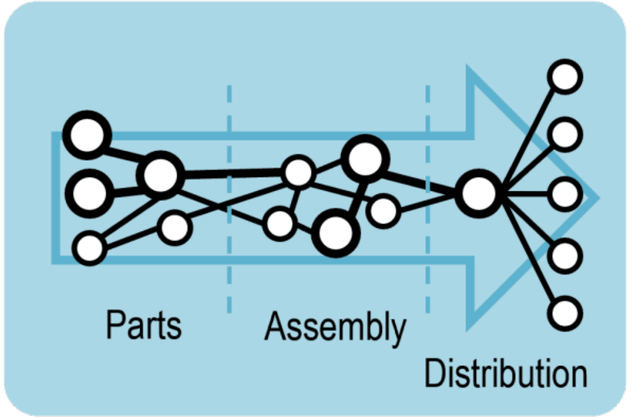
Nodal hierarchy (hub-and-spoke)

Maritime Networks



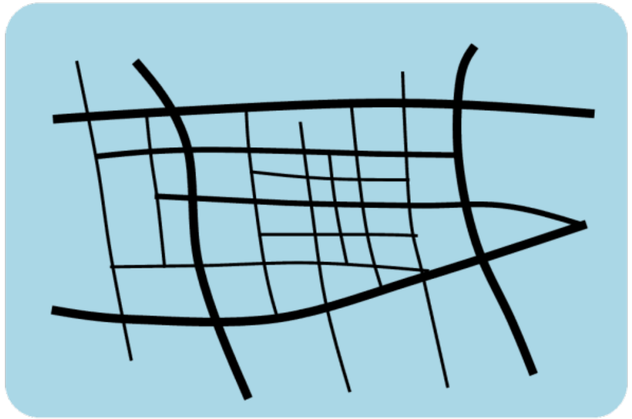
Circuitous nodal hierarchy

Logistical Networks



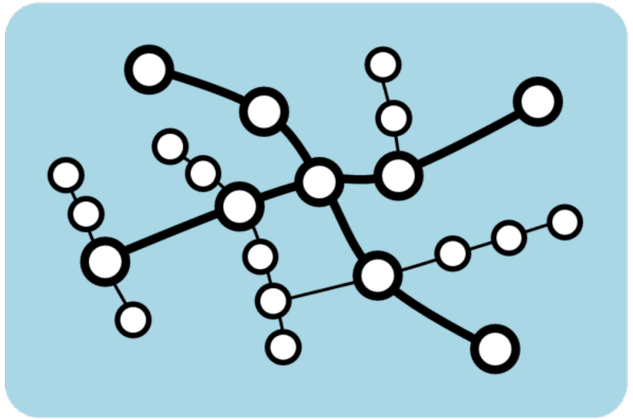
Sequential multi-nodal hierarchy

Road Networks



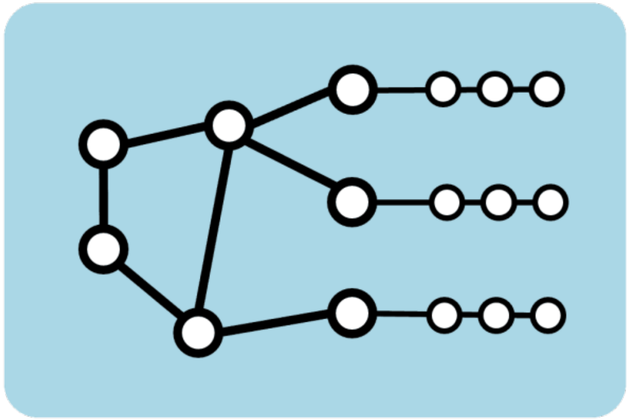
Hierarchical meshes

Rail Networks



Linear nodal hierarchy

Power Grids

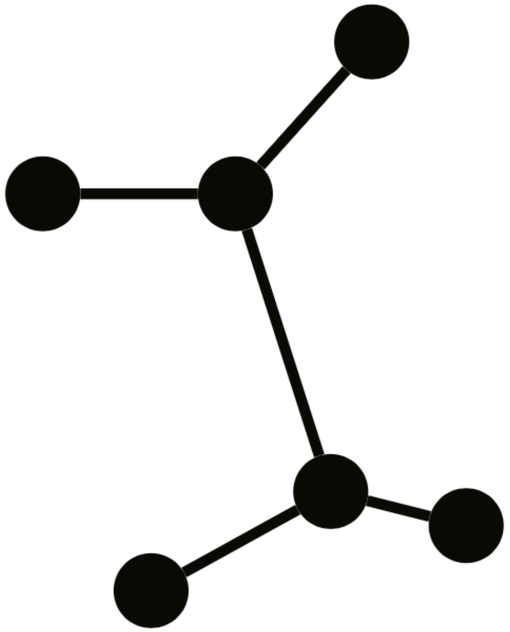


Sequential linear hierarchy

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

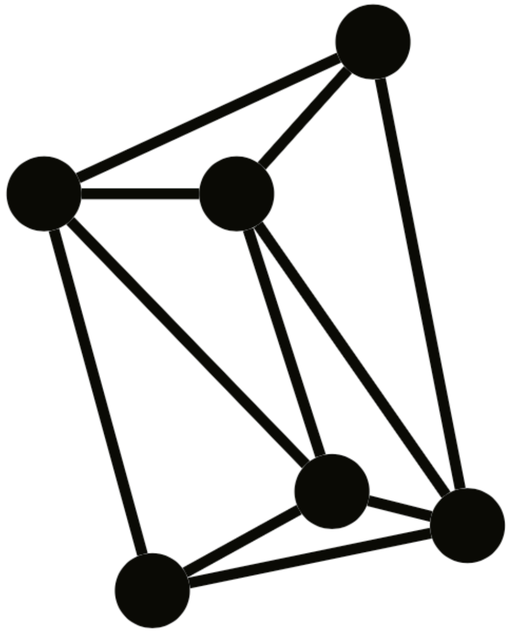
There is a tradeoff between economic investment and resilience

Minimum spanning tree



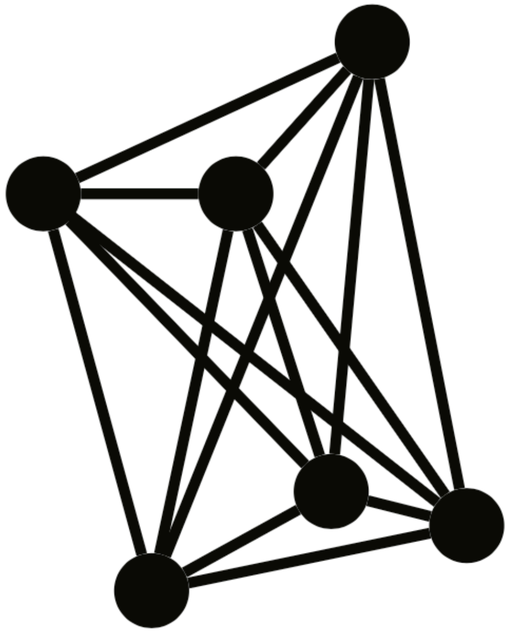
Investor's optimum

Triangulation



Cohesive planar network

Fully connected



Traveler's optimum

Economic

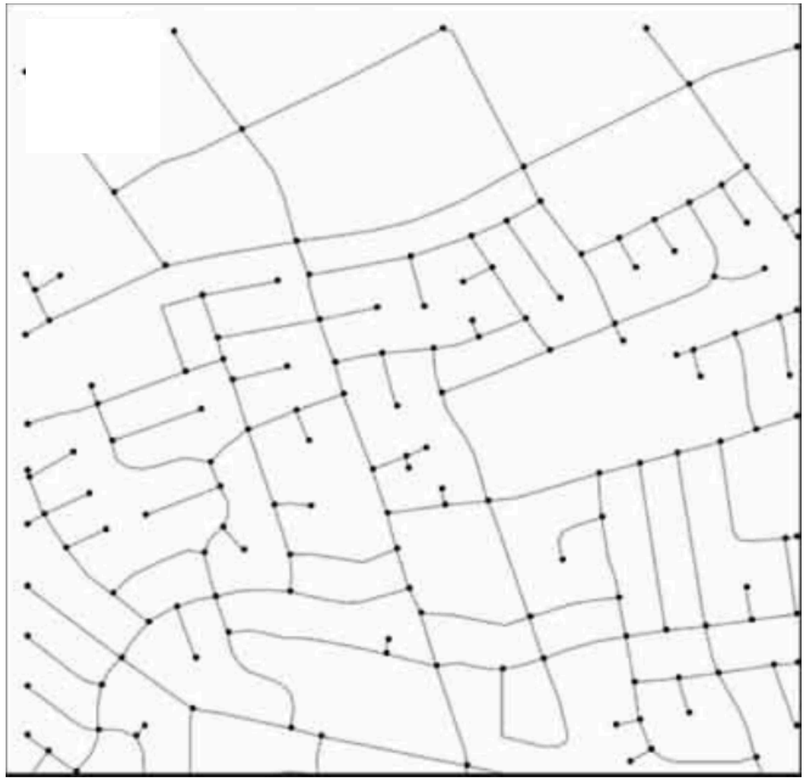
Resilient

Street networks

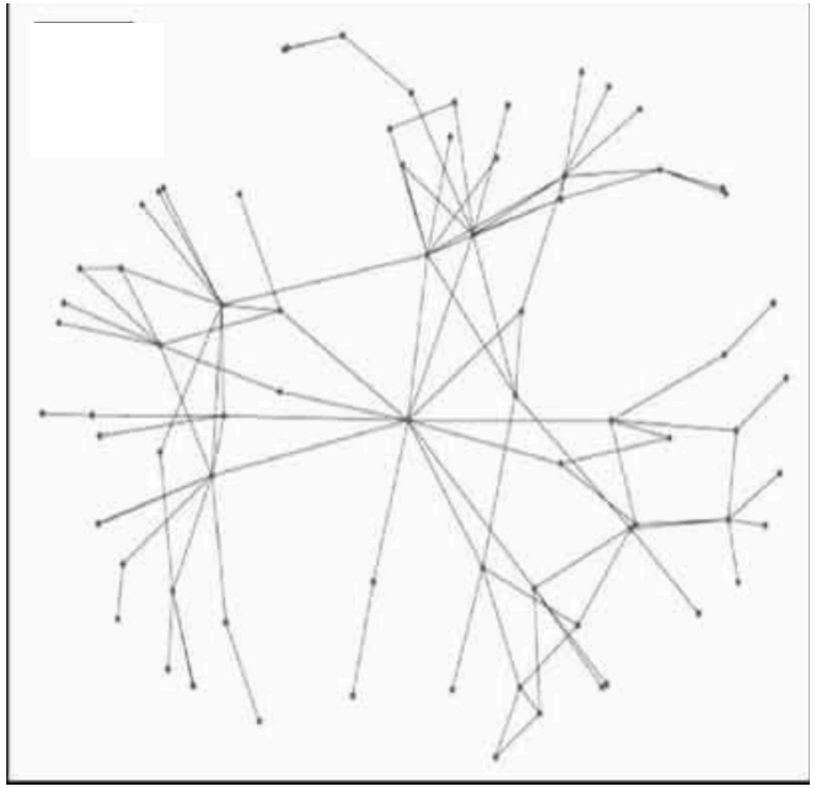
There are several ways of modeling street networks

Original

Primal

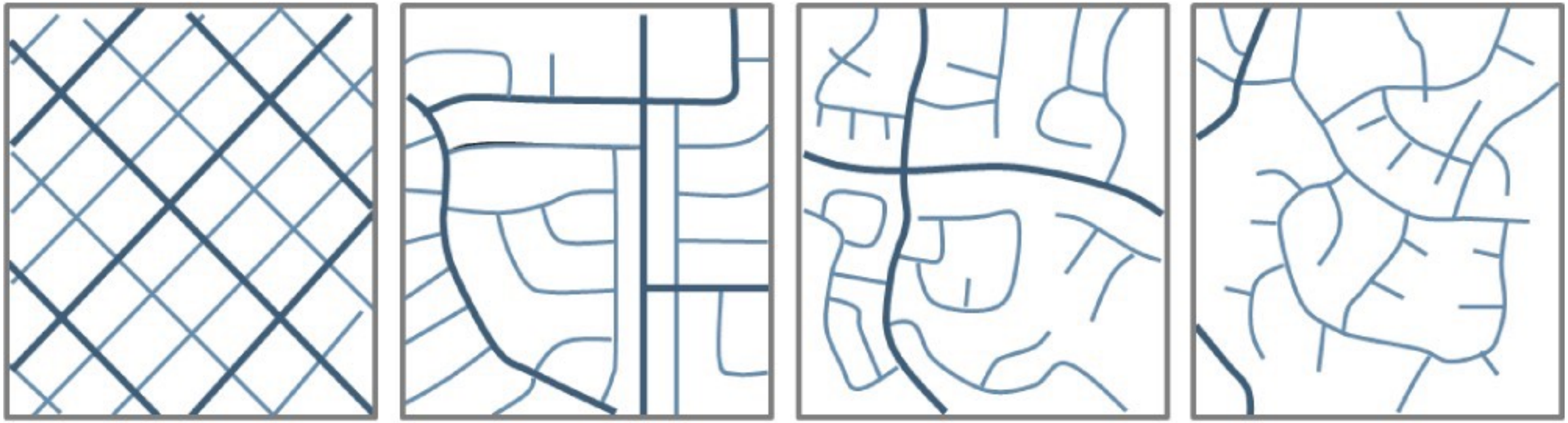


Dual



Porta el al 2006 Env Plan B

Street networks can look very different



Conventional Grid Pattern (c 1900)

Curvilinear Loop Pattern & Beginning of Cul-de-Sacs (1930-1950)

Conventional Cul-de-Sac Pattern (since 1950)

Arterial road

Local street

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

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

Spatial network measures can quantify these differences

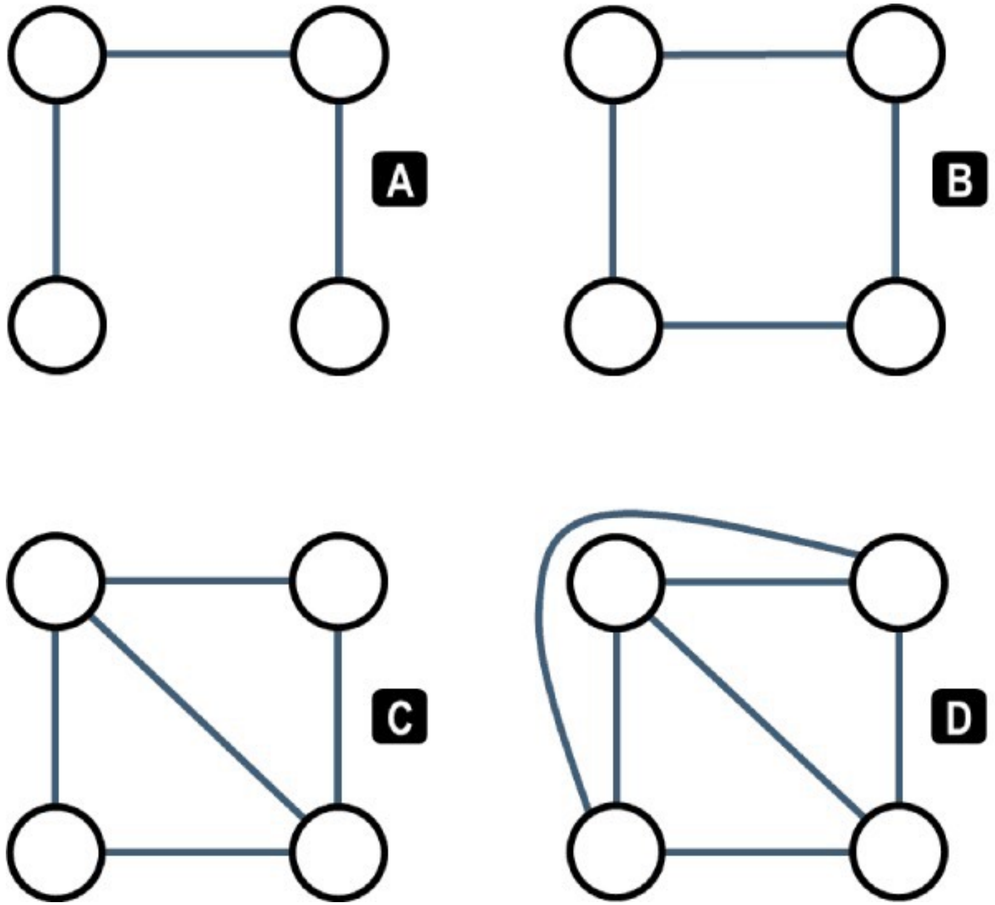
Network density: Links per surface area

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

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

Spatial network measures can quantify these differences

Alpha index: Existing versus possible cycles



$$\alpha = \frac{u}{2v-5} \qquad \alpha = \frac{e-v}{\frac{v(v-1)}{2}-(v-1)}$$

	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

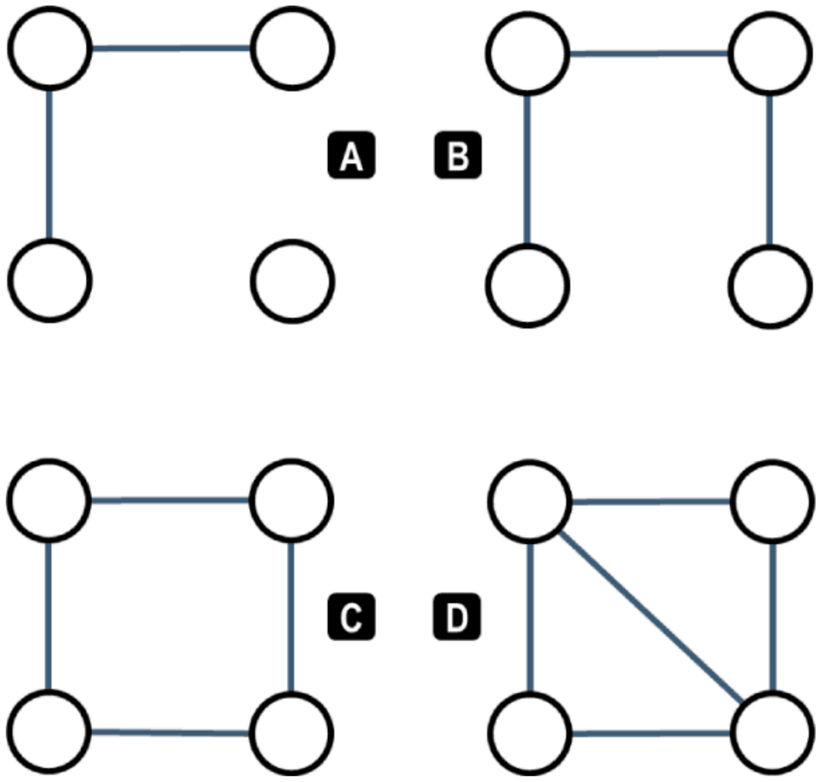
Tree

Max. planar

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

Spatial network measures can quantify these differences

Beta index: The level of connectivity



Beta Index in Graph

$$\beta = \frac{e}{v}$$

	e	v	Beta
A	2	4	0.5
B	3	4	0.75
C	4	4	1.0
D	5	4	1.25

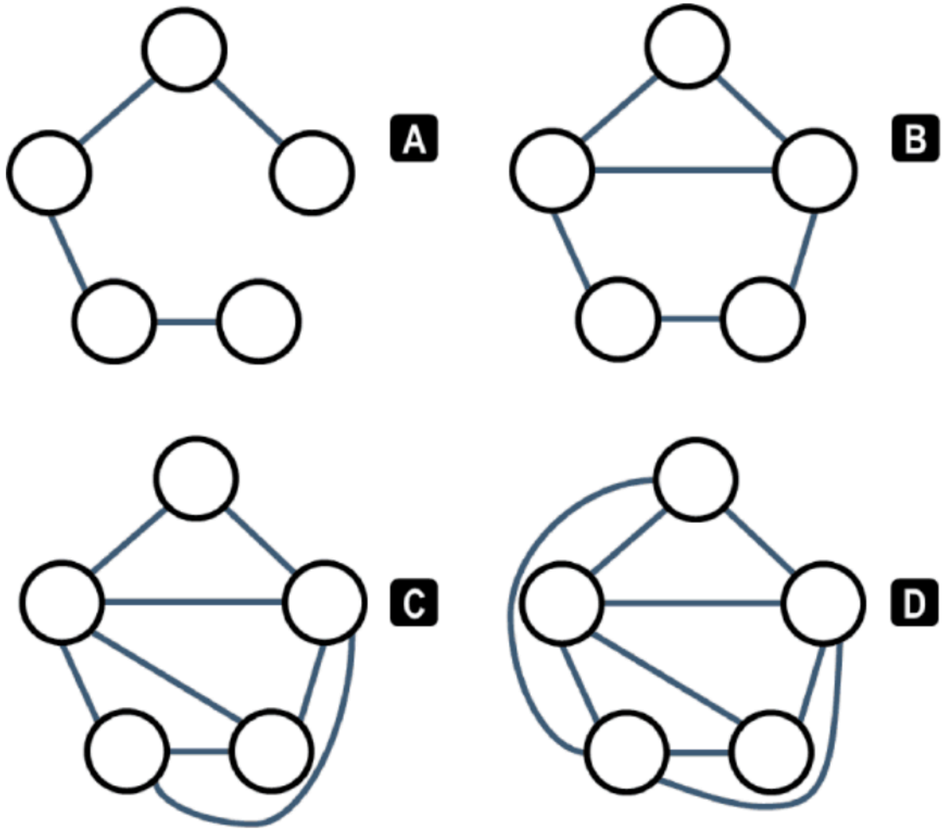
More simple

More complex

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

Spatial network measures can quantify these differences

Gamma index: The level of connectivity



Gamma Index in a Graph

$$\gamma = \frac{e}{3(v-2)}$$

	e	3(v-2)	Gamma
A	4	9	0.44
B	6	9	0.66
C	8	9	0.88
D	9	9	1.0

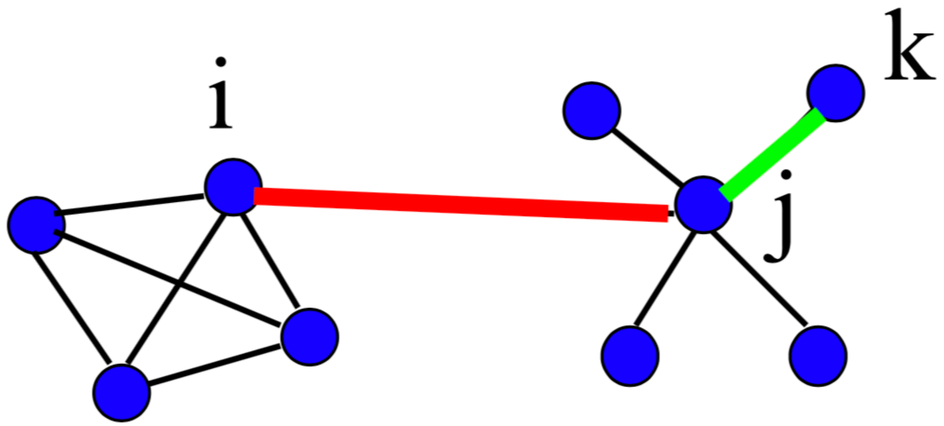
Less connected

Completely connected

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



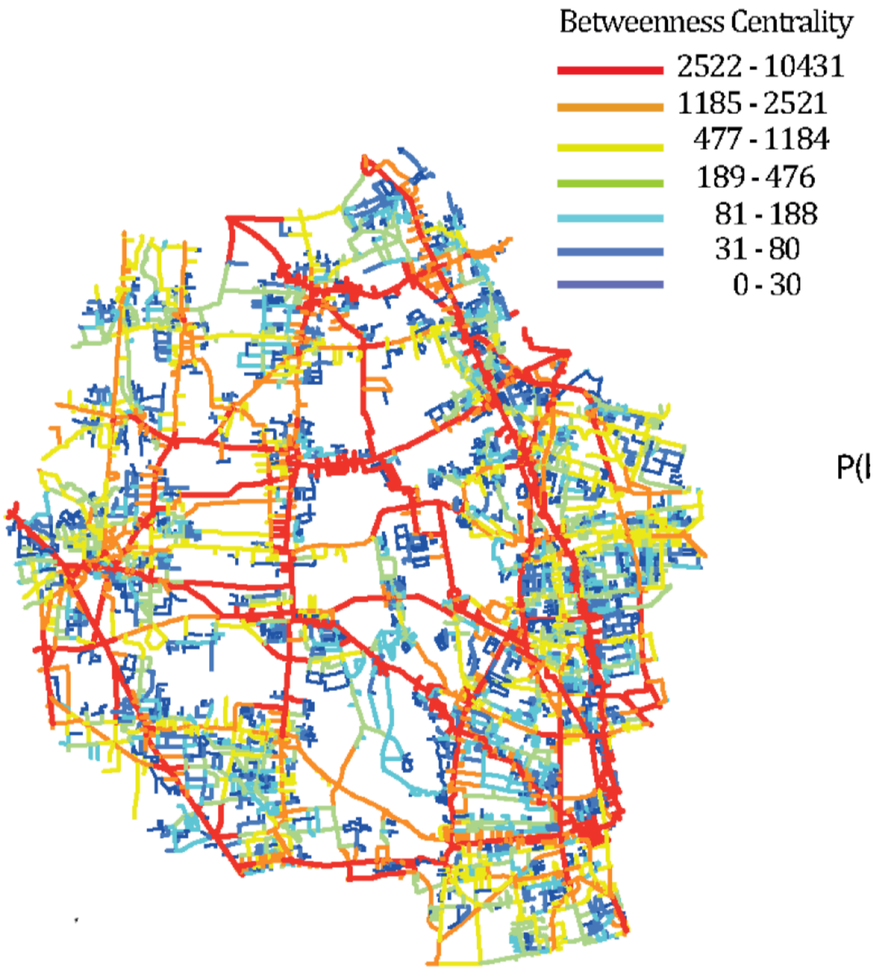
ij: large centrality

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.



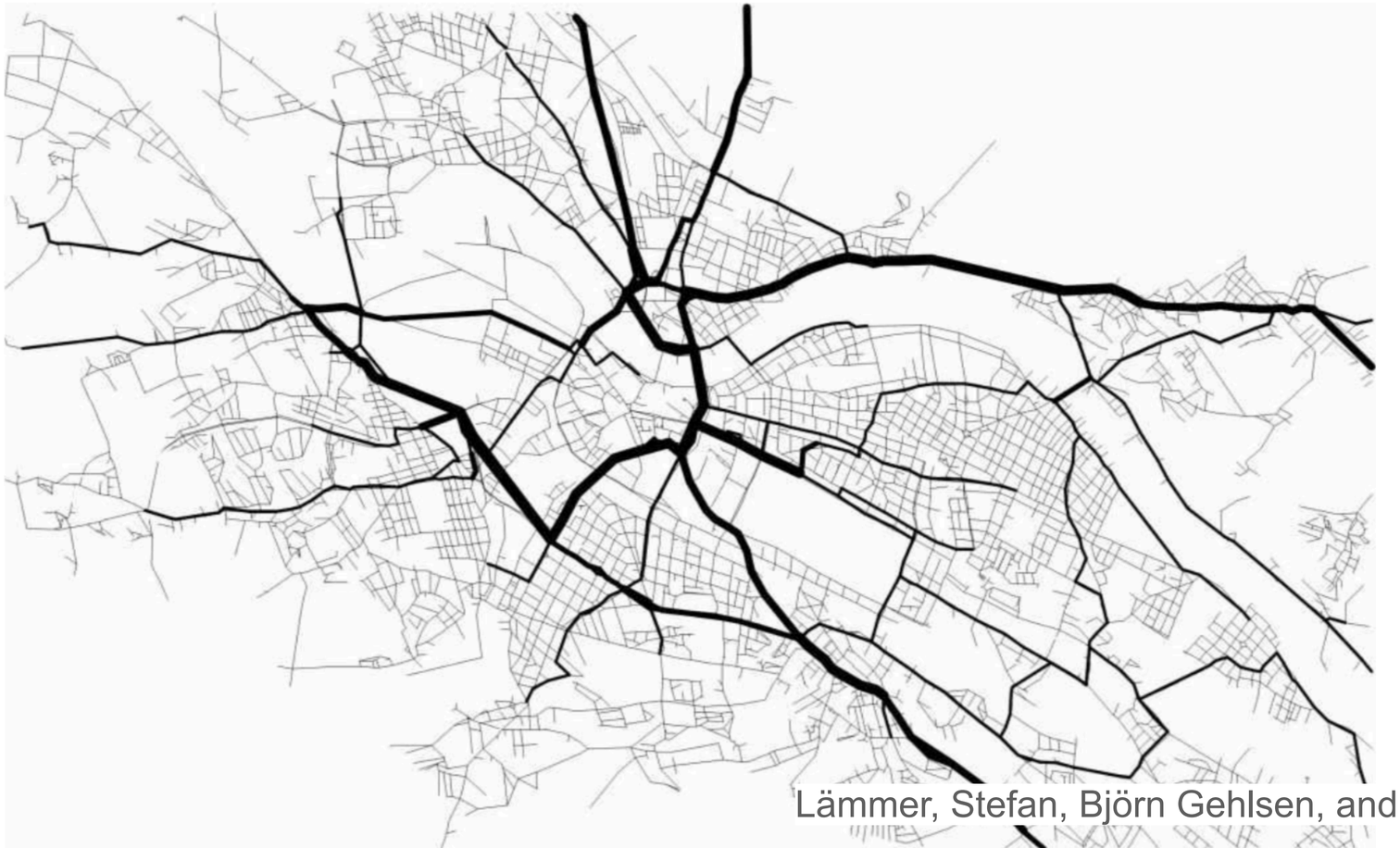
Measures the importance of a segment in the shortest paths flow

Gives the backbone of central roads

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:



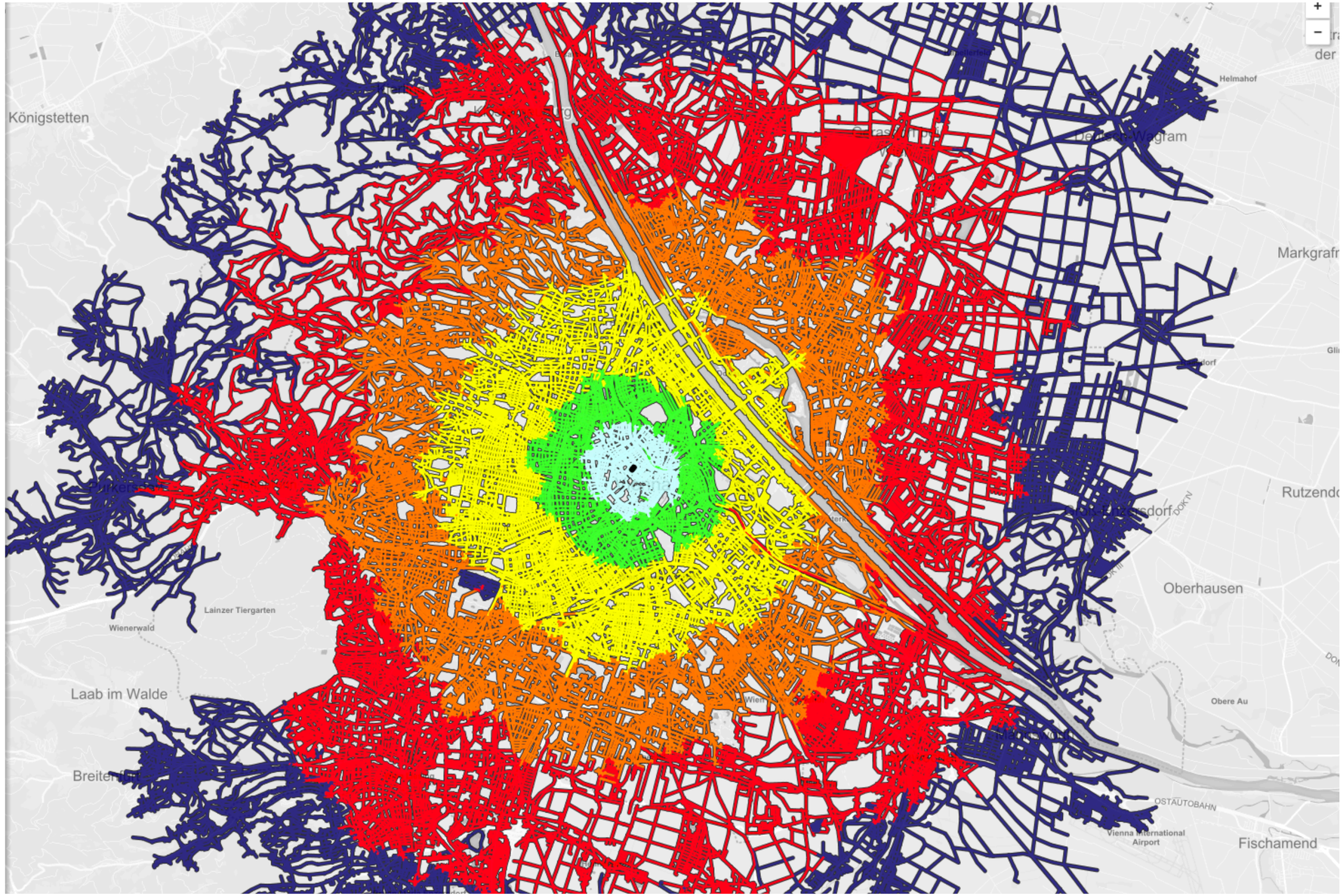
Lämmer, Stefan, Björn Gehlsen, and Dirk Helbing. *Physica A* 363.1 (2006): 89-95.

GrowBike.net

https://growbike.net/city/copenhagen

Applications of street networks

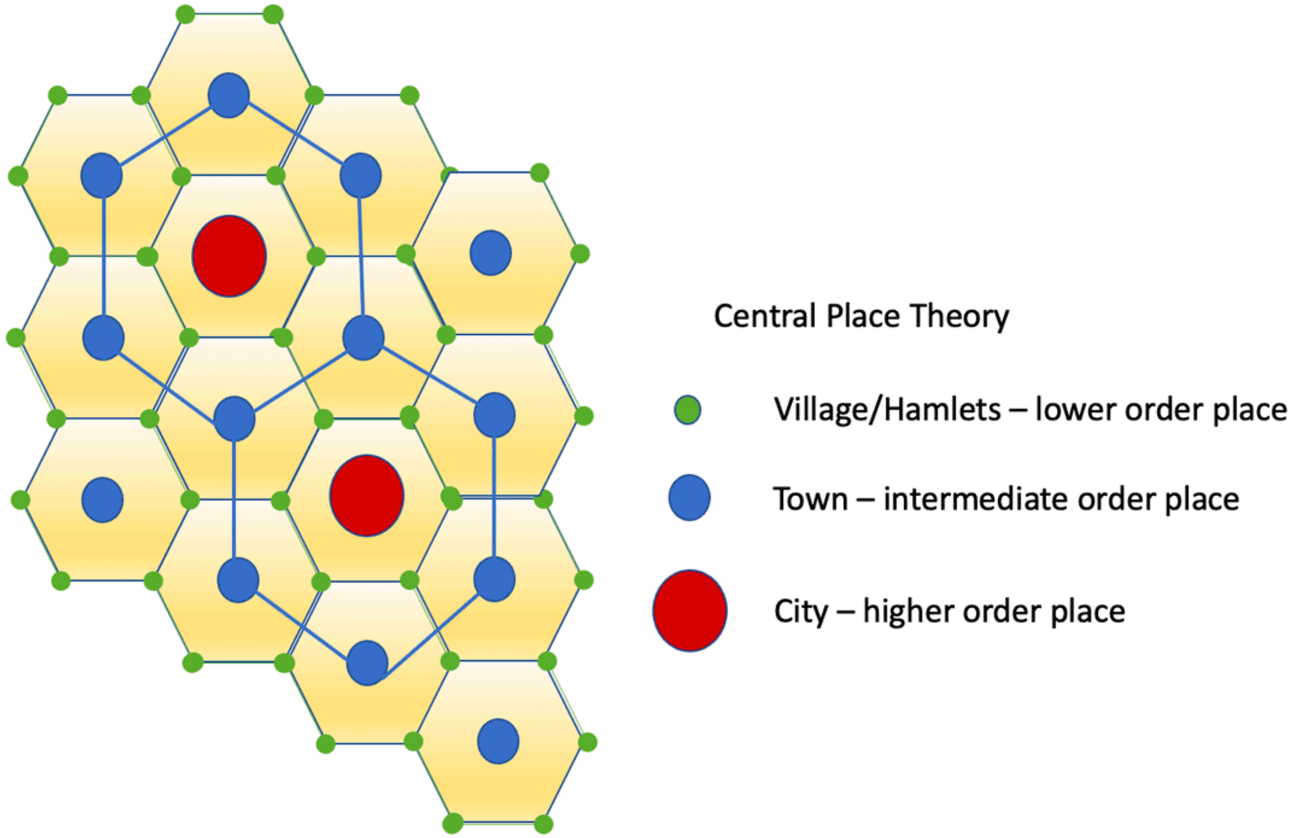
Street networks are used to model travel times



https://magazine.bikecitizens.net/analysis-data-planning-tool-for-bicycle-traffic/

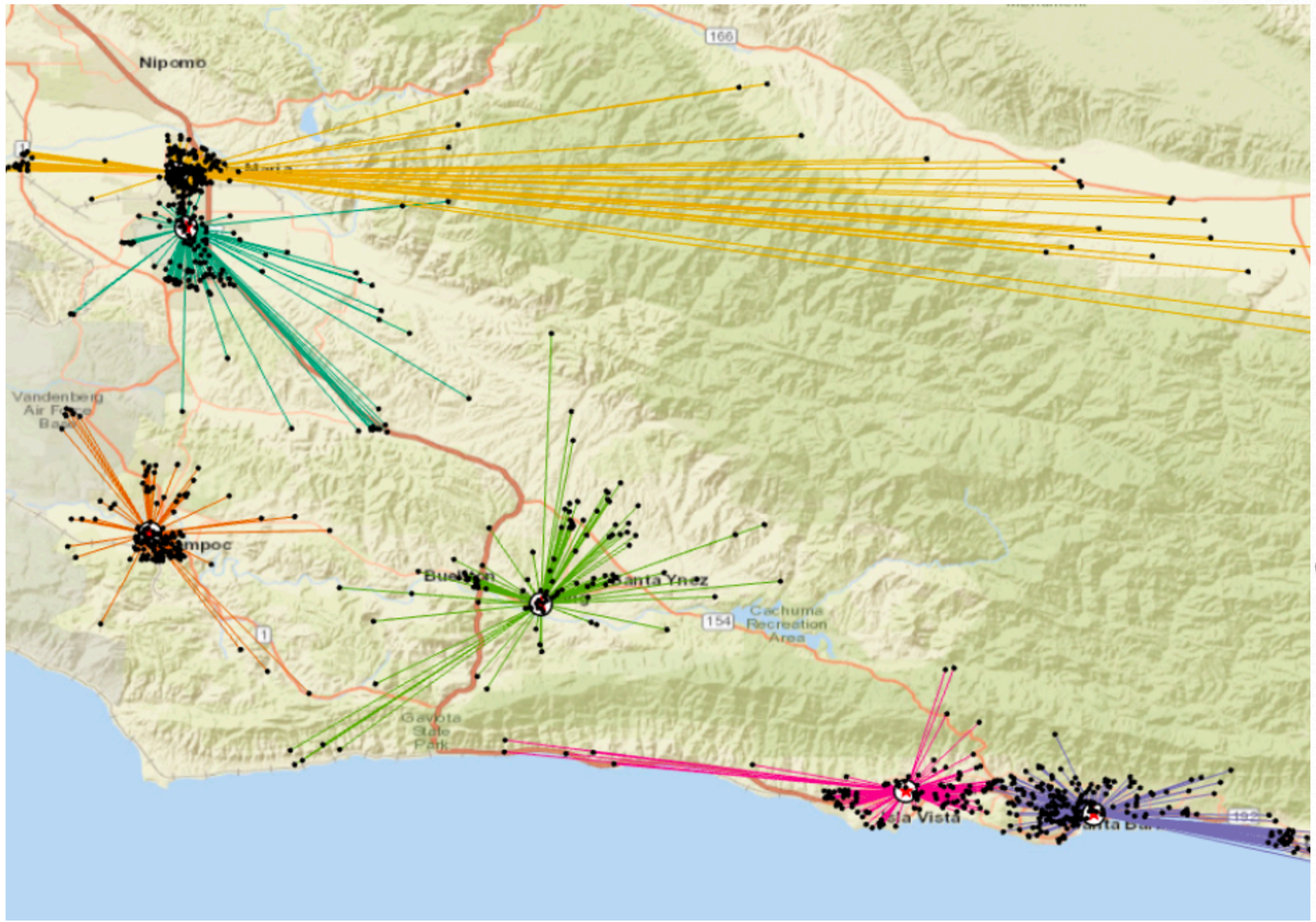
Spatial networks explain where we live, work and shop

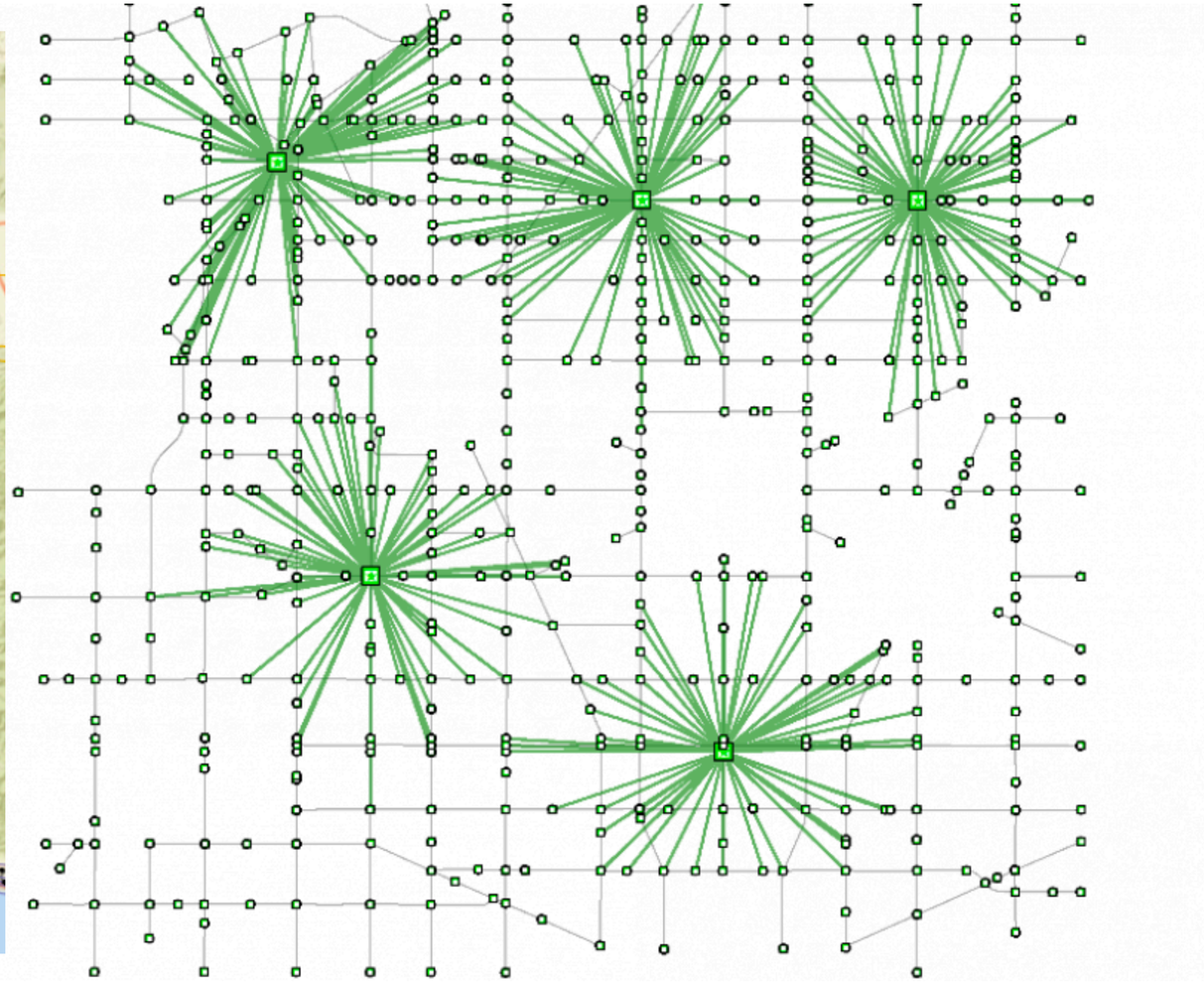
Central Place Theory



Spatial networks are used for planning

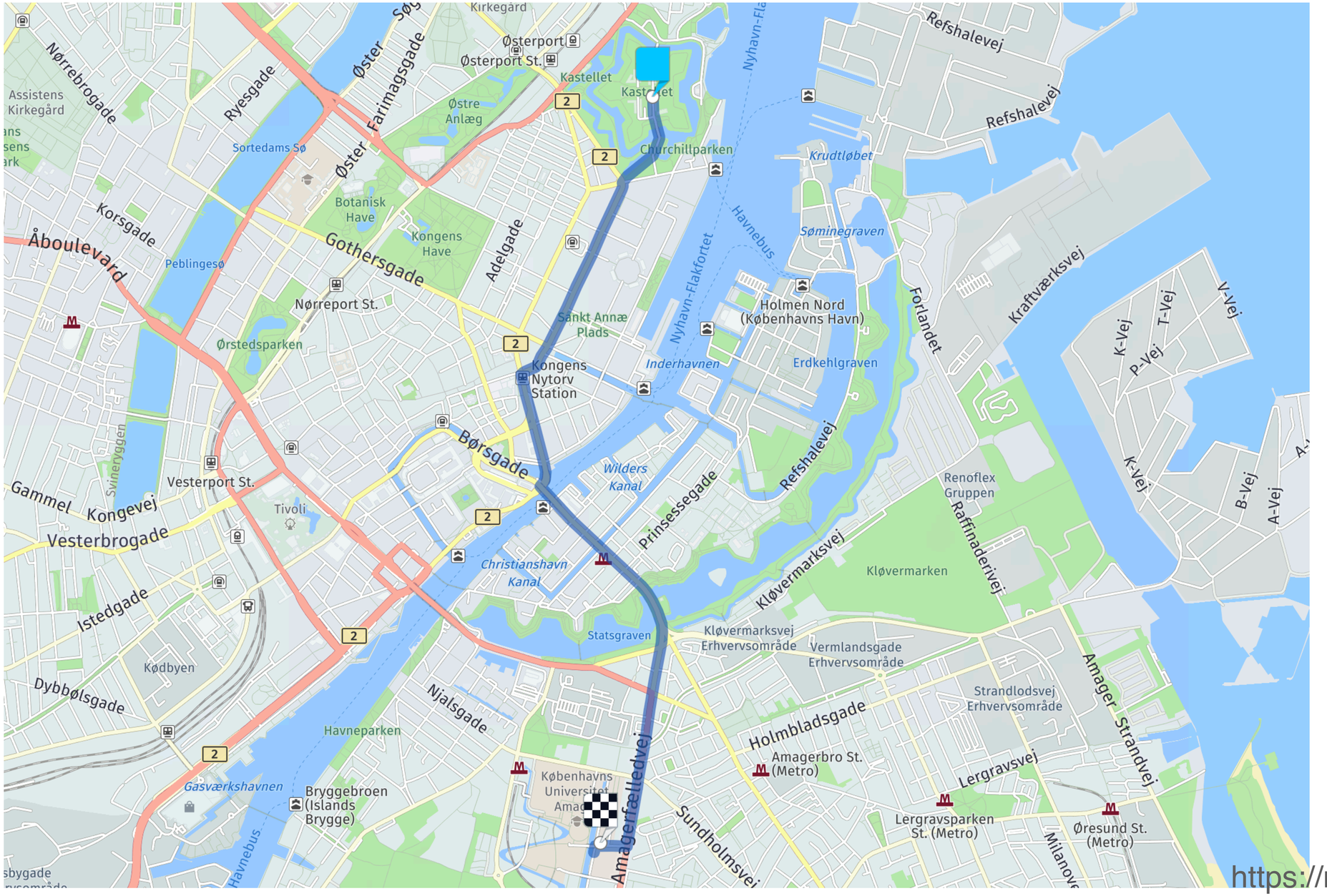
Location-allocation & spatial optimization





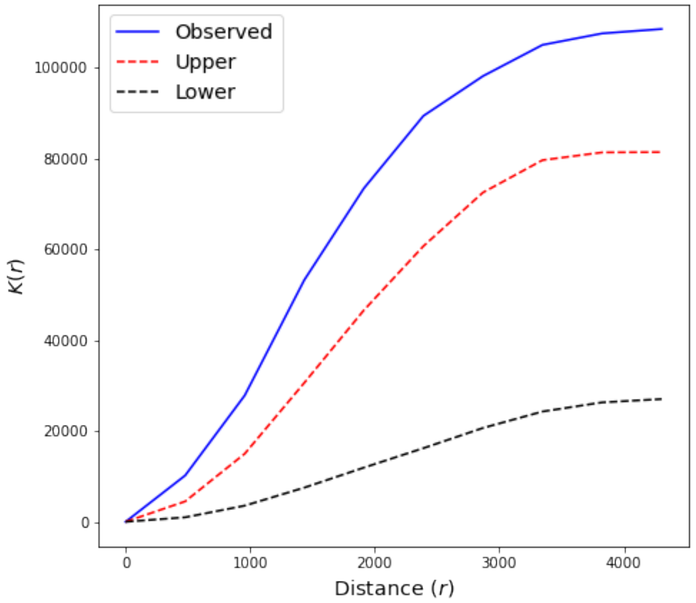
https://gistbok.ucgis.org/topic-keywords/optimization

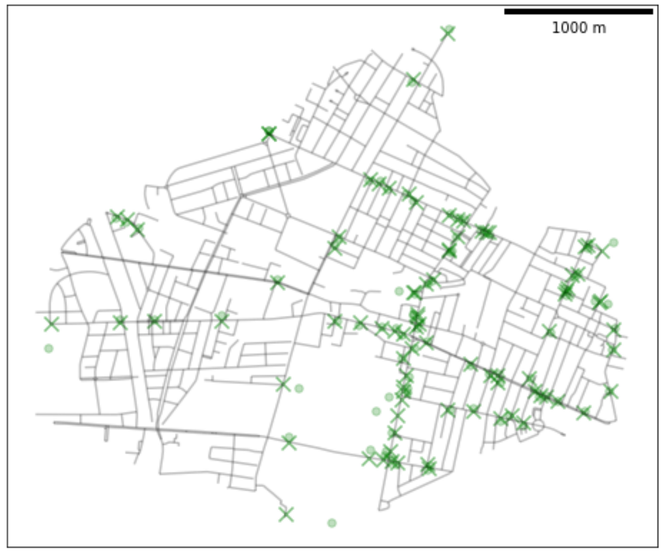
Most of us use street network data every day

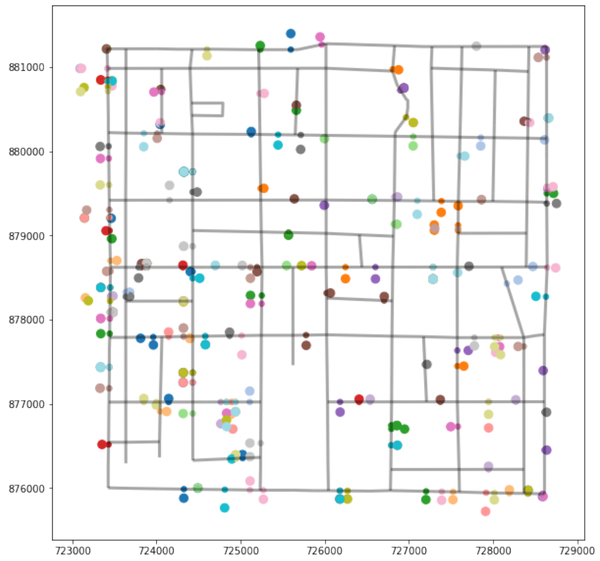


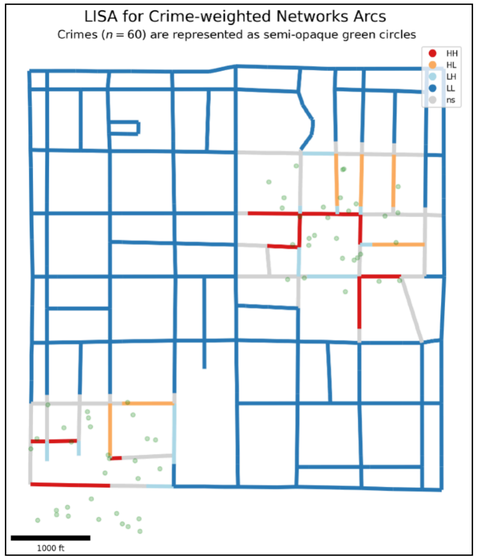
https://mobile.here.com/

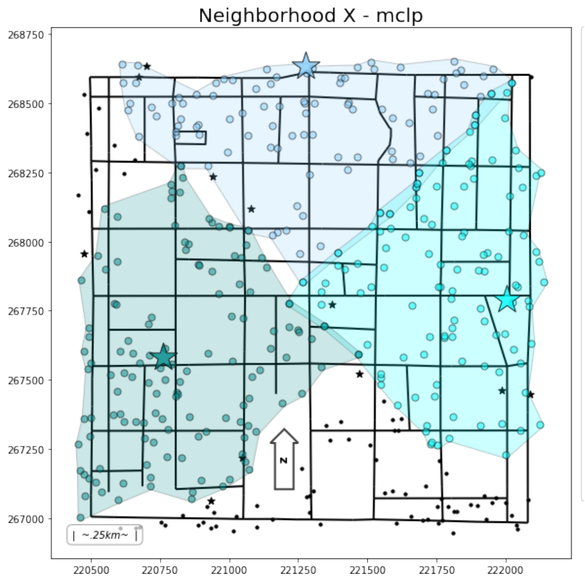
Useful Python libraries: spaghetti











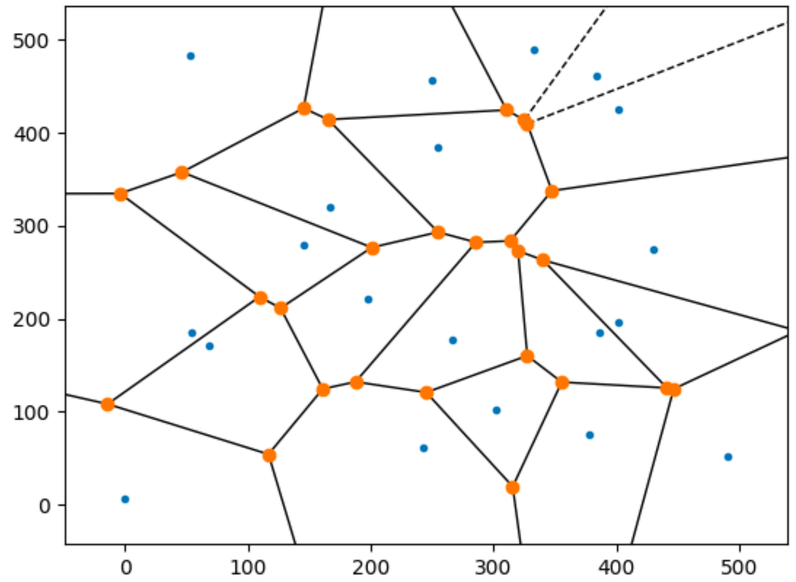
https://pysal-spaghetti.readthedocs.io/en/latest/index.html

spaghetti

```
# Create spaghetti network
ntw = spaghetti.Network(gdf)


# Create geodataframes with nodes and edges
vertices_df, arcs_df = spaghetti.element_as_gdf(
    ntw, vertices=ntw.vertex_coords, arcs=ntw.arcs
)

# snap POIS to network
ntw.snapobservations(gdf_pois, “pois", attribute=True)
```

scipy.spatial

```
from scipy.spatial import Voronoi, voronoi_plot_2d

vor = Voronoi(coords)

fig = plt.figure(figsize=(10,10))
axes = fig.add_axes([0, 0, 1, 1])
voronoi_plot_2d(vor, ax=axes, line_width = 2, point_size=20);
```



Useful Python libraries: Pandana

For *fast* network/accessibility analysis

https://udst.github.io/pandana/

Useful Python libraries: Pandana

```
# create pandana network object
network = pandana.Network(node_x=nodes['x'],
                          node_y=nodes['y'],
                          edge_from=edges['u'],
                          edge_to=edges['v'],
                          edge_weights=edges.edge_weight
```

https://udst.github.io/pandana/

Useful Python libraries: Pandana

```
# set the POIs locations on the network
network.set_pois(category=‘category_name',
                 maxdist=1000
                 maxitems=3, # number of places to find
                 x_col=centroids.x,
                 y_col=centroids.y)
```

https://udst.github.io/pandana/

Useful Python libraries: Pandana

```
# find distances to nearest POIs from all nodes
distances = network.nearest_pois(distance=1000,
                                 category='category_name',
                                 num_pois=3)
```

https://udst.github.io/pandana/

Other useful Python libraries:

OSMnx

NetworkX

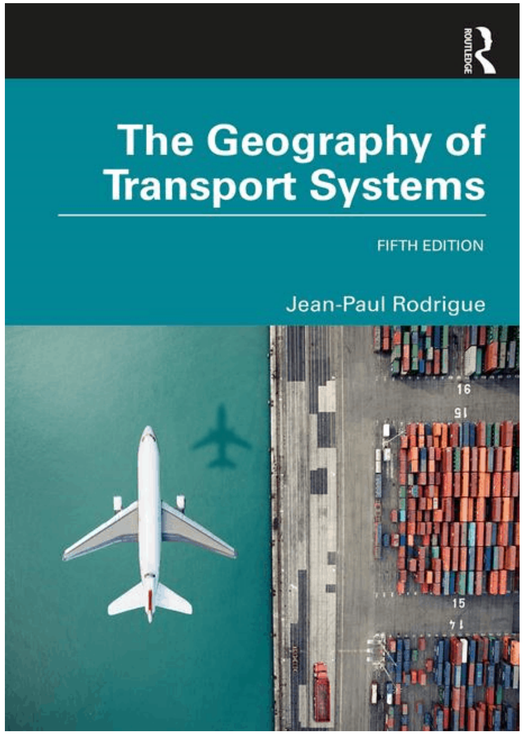
momepy

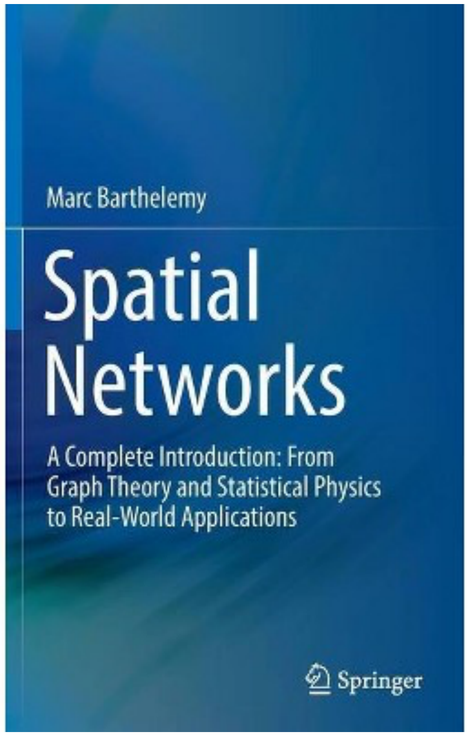
UrbanAccess

snkit

Jupyter

Sources and further materials for today's class





https://transportgeography.org/

https://spatial-analytics.readthedocs.io/en/latest/lessons/L3/spatial-network-analysis.html

Next week: Raster data & rasterio