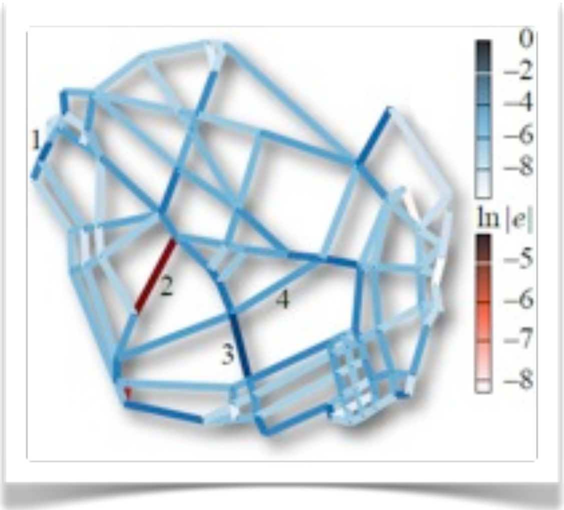
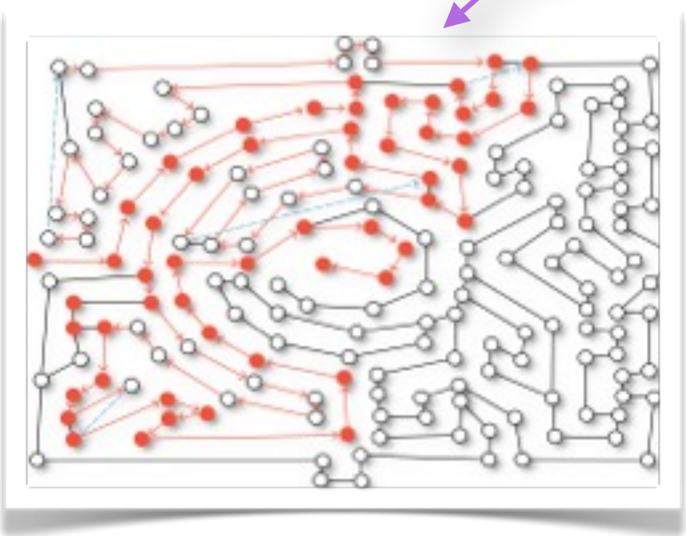
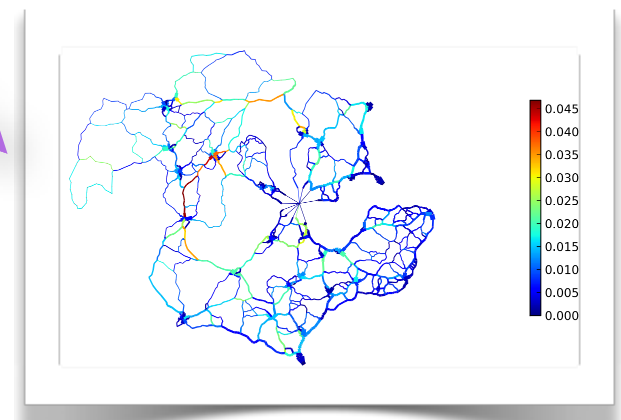
CABDyN Seminar Series: 13 November, 2012

Exploring road networks with greedy navigators and their core-periphery structures







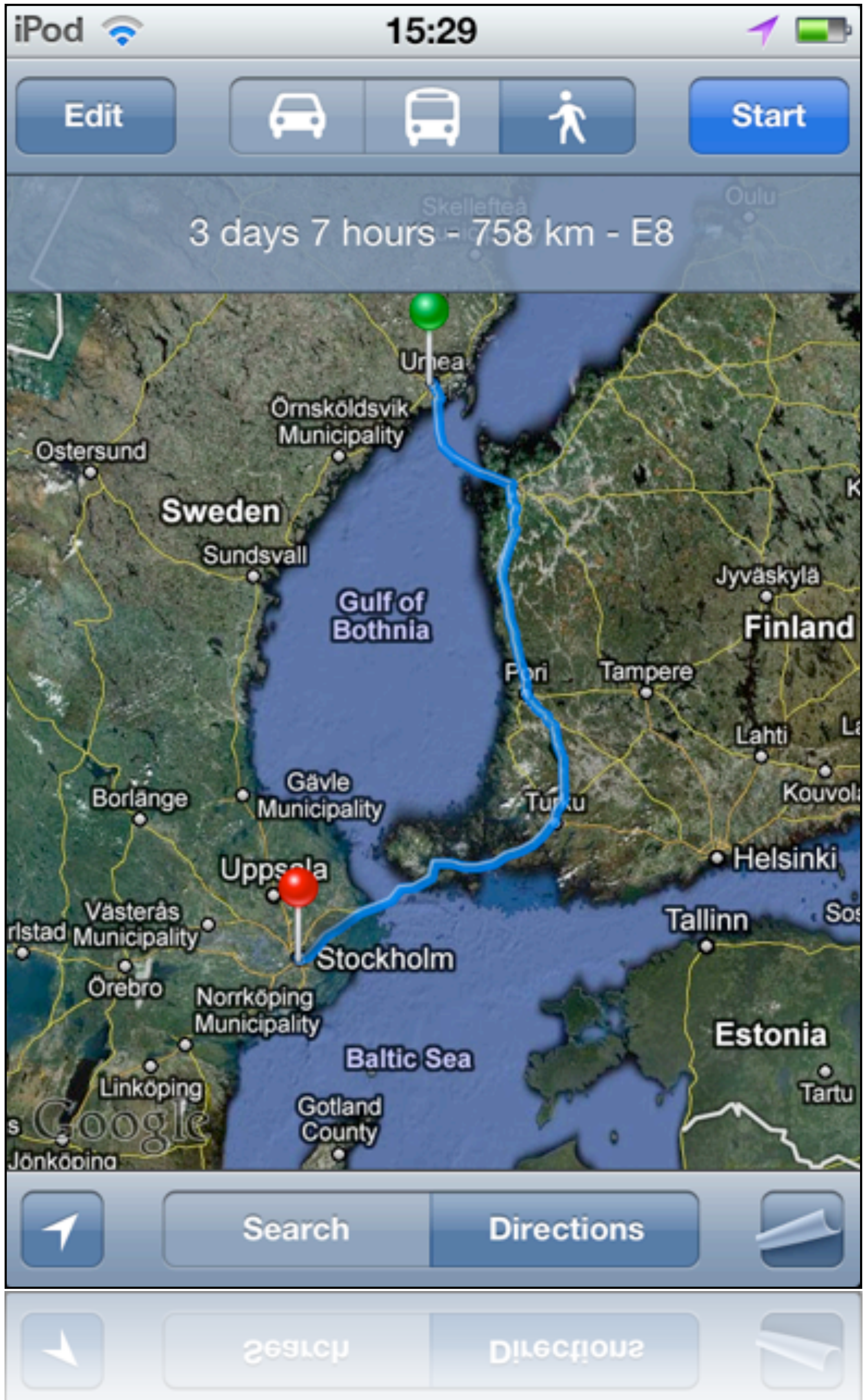
Sang Hoon Lee
Oxford Centre for Industrial and Applied Mathematics (OCIAM), Mathematical Institute, University of Oxford

lee@maths.ox.ac.uk

in collaboration with **Petter Holme** (Umeå/Sungkyunkwan/Stockholm Univ.) & **Mason A. Porter** (OCIAM, Univ. of Oxford)

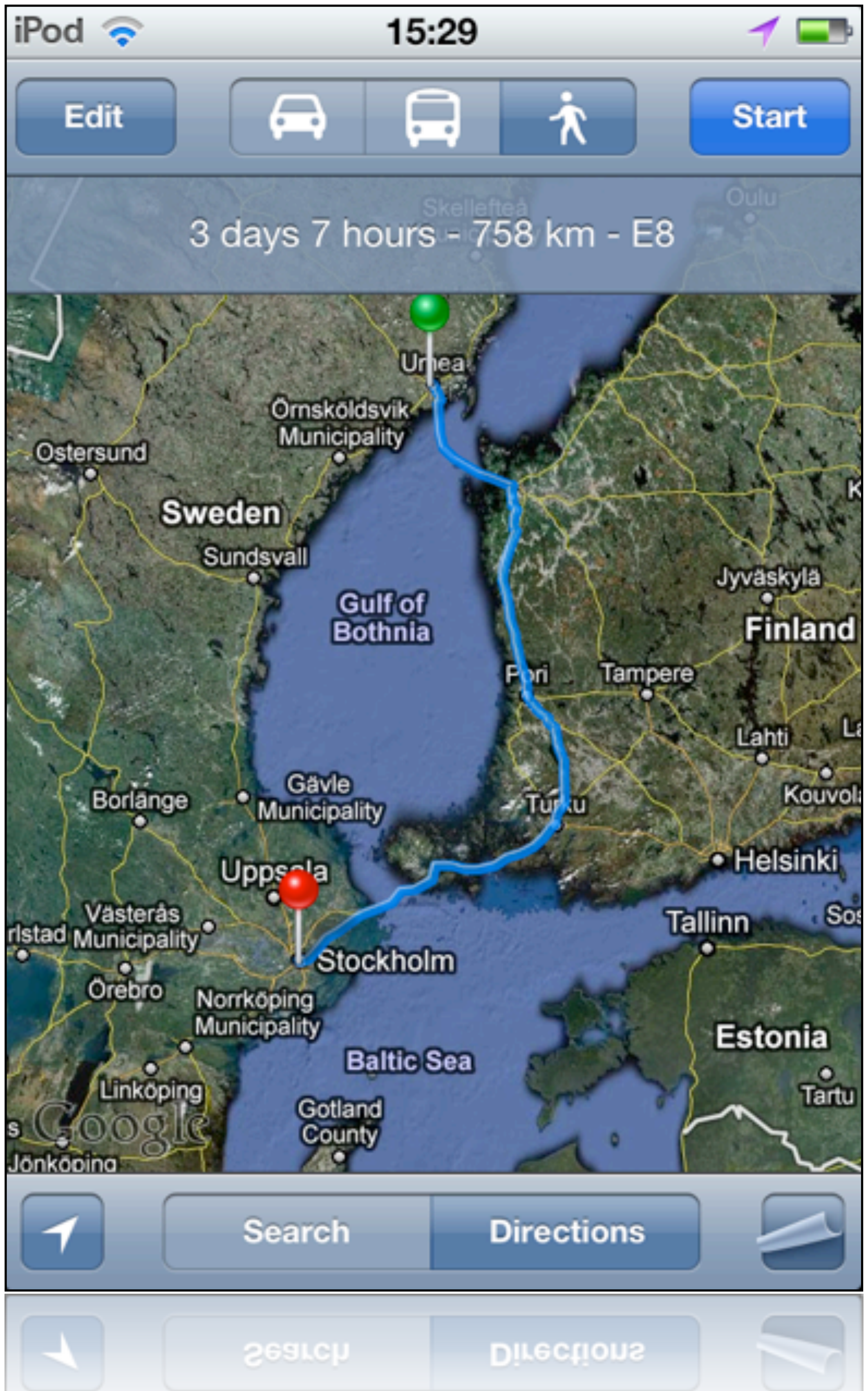
SHL & P. Holme, Physica A **390**, 3996 (2011); Phys. Rev. Lett. **108**, 128701 (2012); arXiv:1205.0537 (Eur. Phys. J.-Spec. Top., in press); arXiv:1206.6651.

Real navigation?

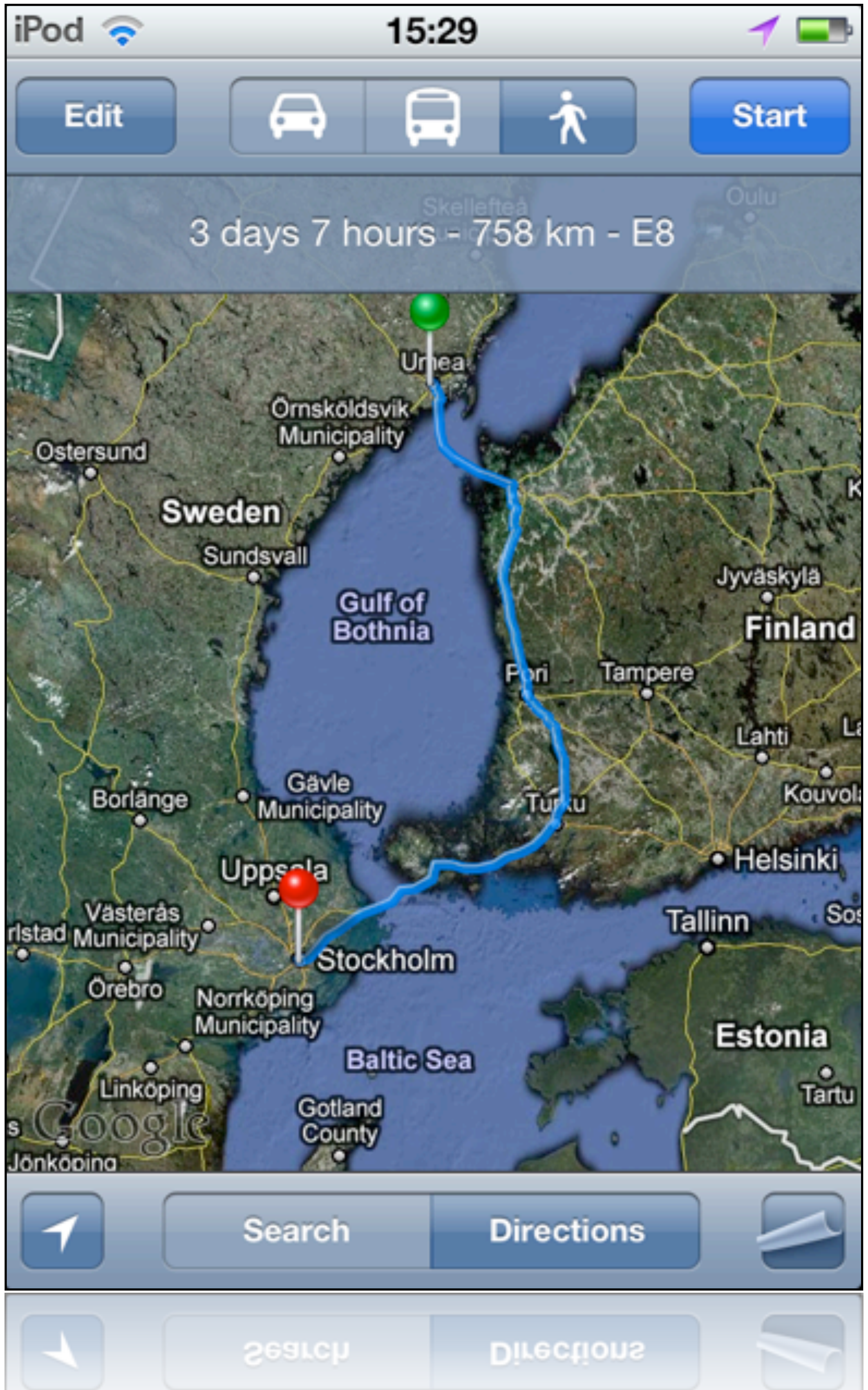


Real navigation?

"Which information does human being use?"



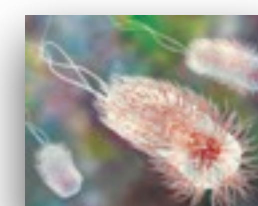
Real navigation?



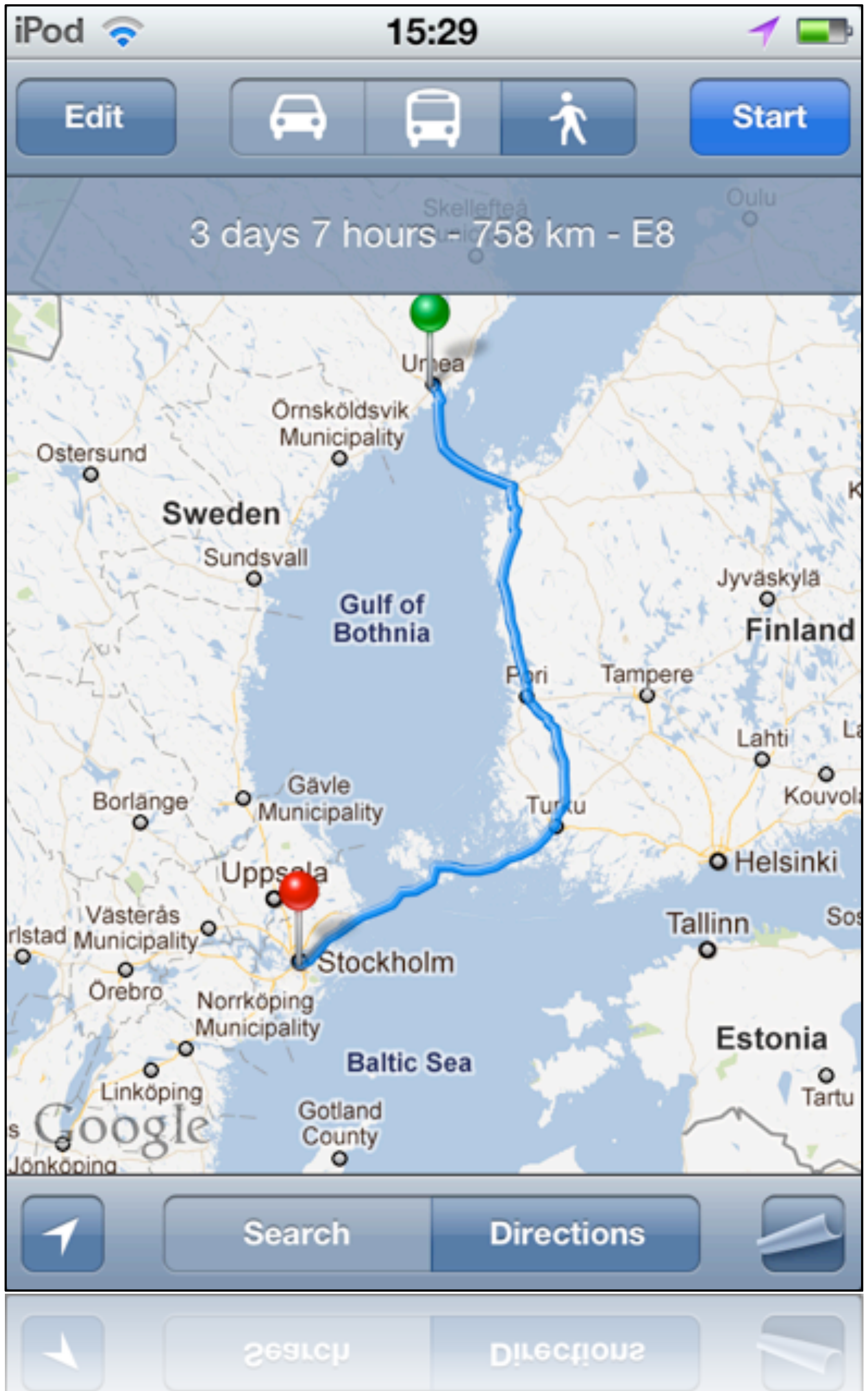
"Which information does human being use?"

"cognitive map" (map in mind/brain)

simple organisms also use chemotaxis to find the target



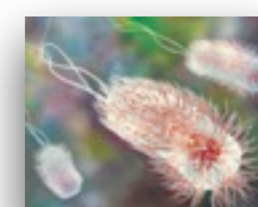
Real navigation?



"Which information does human being use?"

"cognitive map" (map in mind/brain)

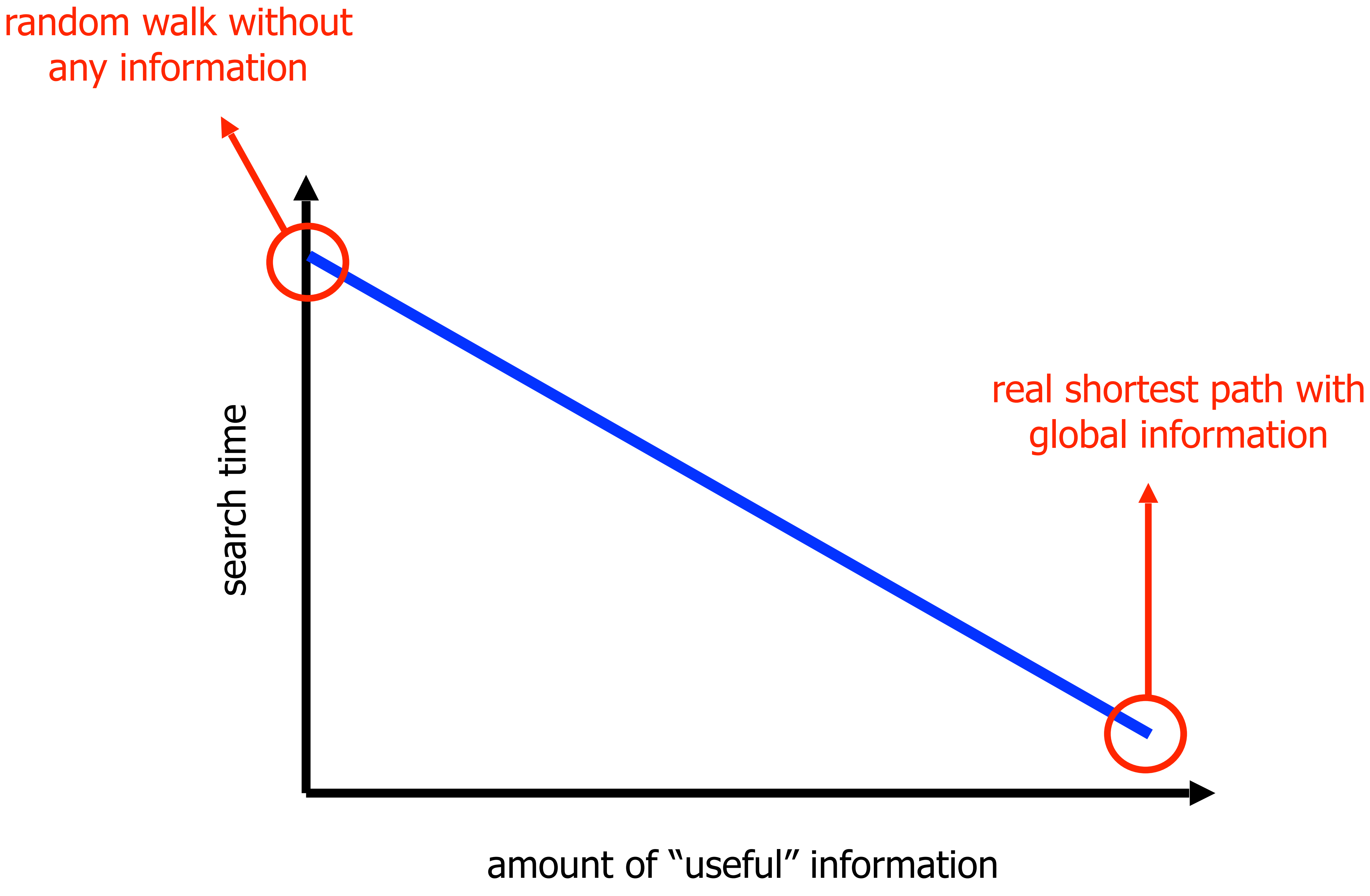
simple organisms also use chemotaxis to find the target



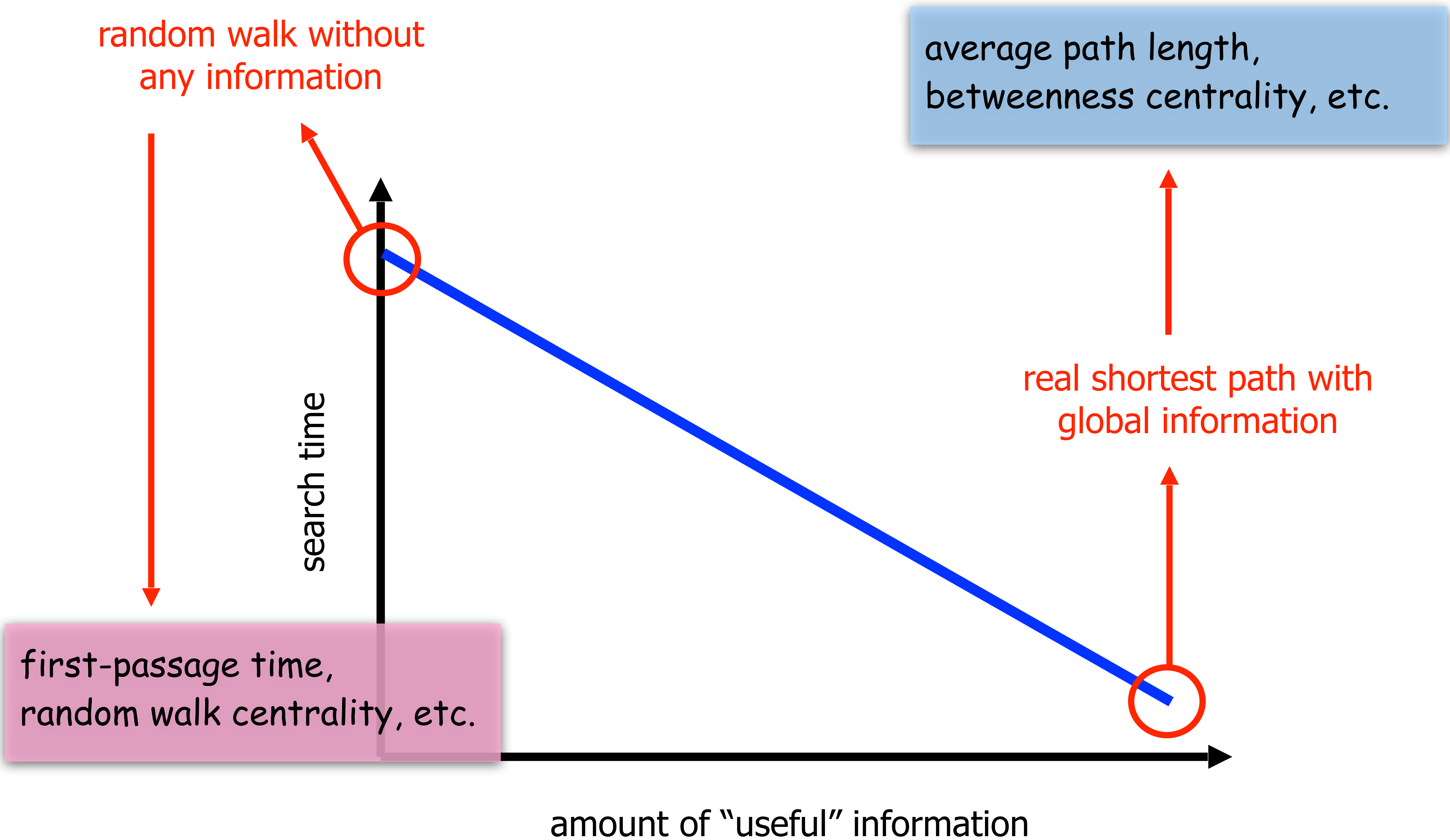
Simplify!
(distance/directional information)

spatial network on 2D

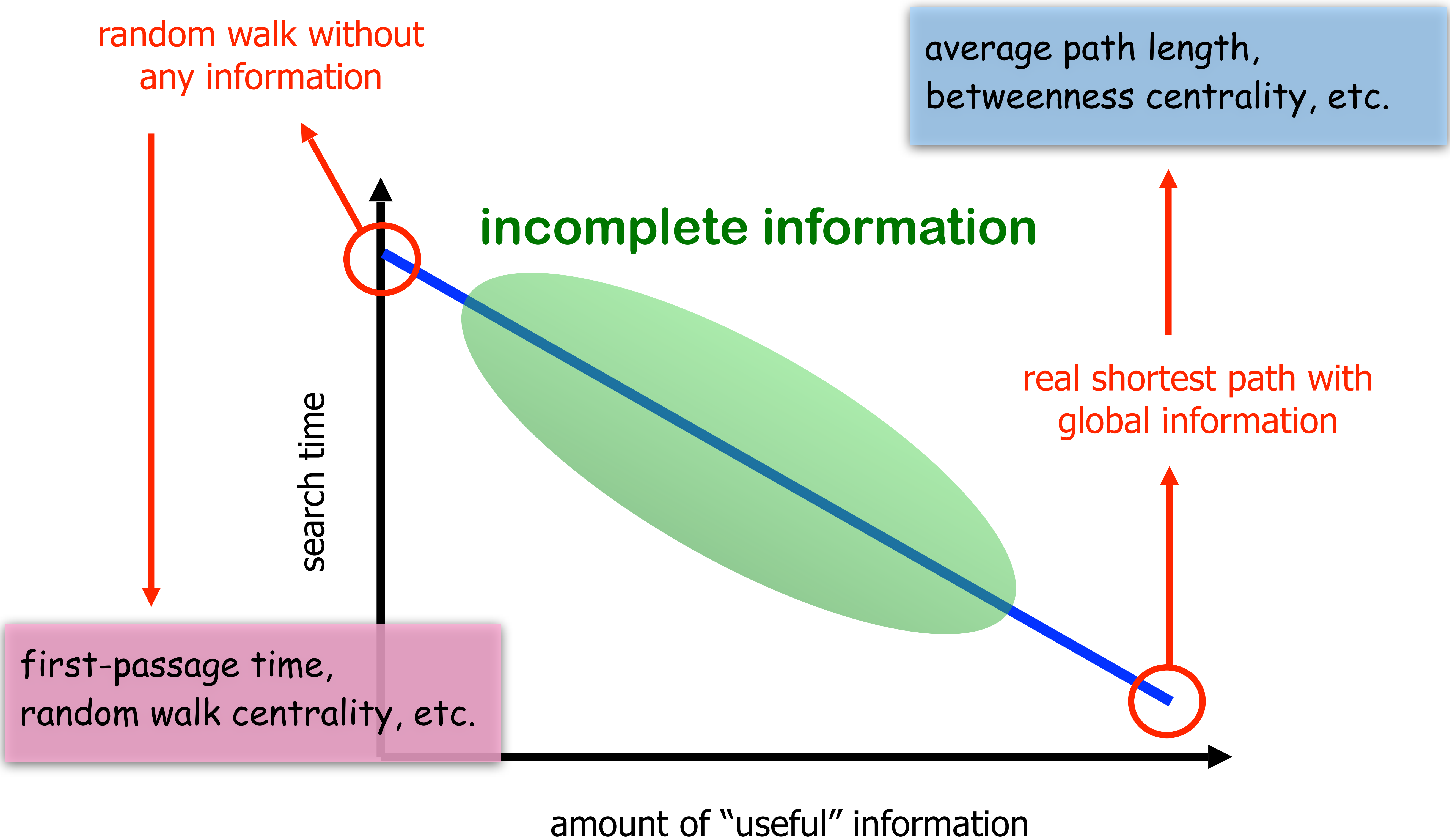
Big picture


random walk without
any information
search time
real shortest path with
global information
amount of “useful” information

Big picture


random walk without any information
average path length, betweenness centrality, etc.
search time
real shortest path with global information
first-passage time, random walk centrality, etc.
amount of "useful" information

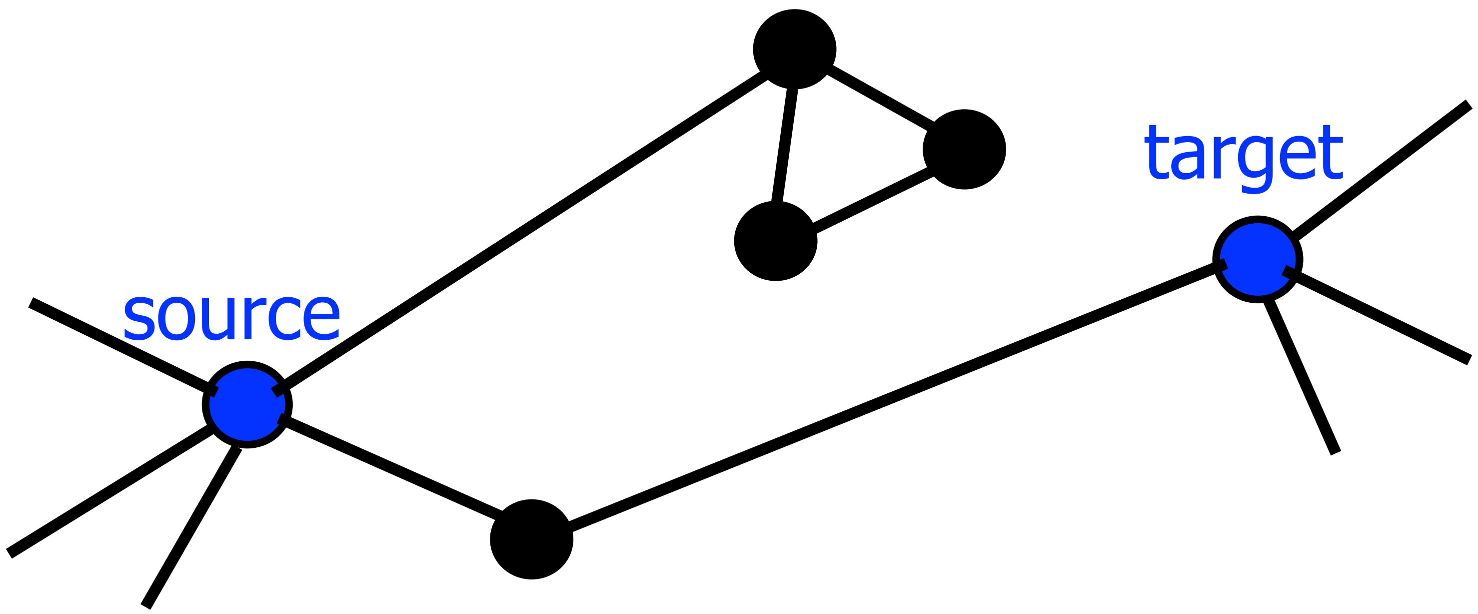
Big picture



random walk without
any information
average path length,
betweenness centrality, etc.
incomplete information
search time
real shortest path with
global information
first-passage time,
random walk centrality, etc.
amount of "useful" information

Greedy routing based on local geometric information

- "**moving to the neighbor closest to the target**"-strategy
- limitation: it sometimes fails, due to the possibility of being trapped

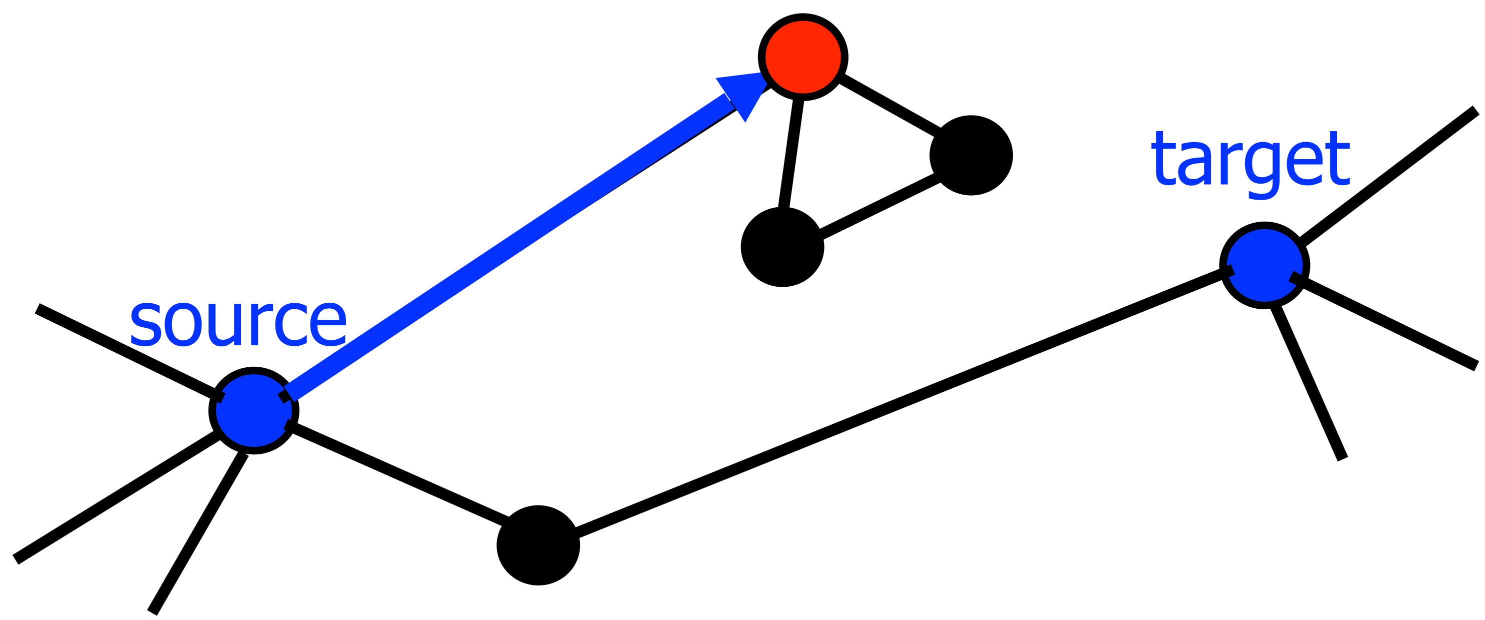


Note: look at the graph as shown in the 2D (Euclidean) space!

J.M. Kleinberg, Nature **406**, 845 (2000); M. Boguñá *et al.*, Nat. Phys. **5**, 74 (2008); PRL **102**, 058701 (2009).

Greedy routing based on local geometric information

- "**moving to the neighbor closest to the target**"-strategy
- limitation: it sometimes fails, due to the possibility of being trapped

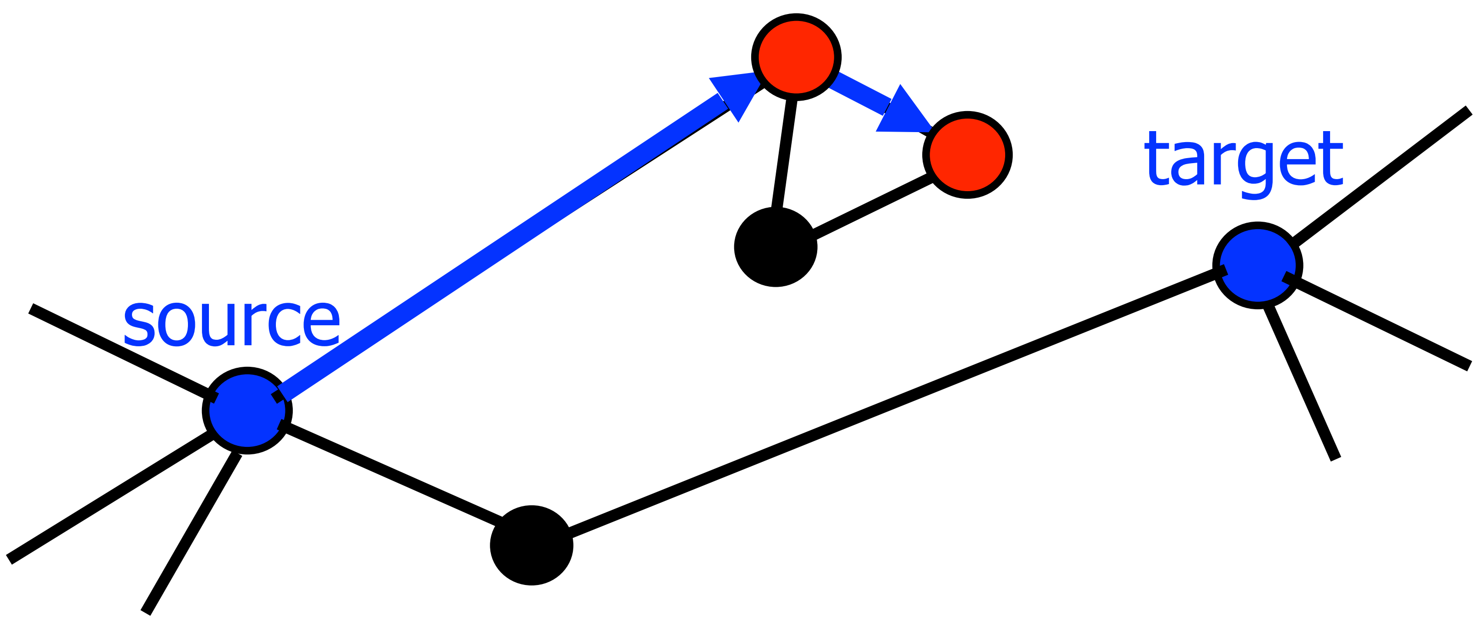


Note: look at the graph as shown in the 2D (Euclidean) space!

J.M. Kleinberg, Nature **406**, 845 (2000); M. Boguñá *et al.*, Nat. Phys. **5**, 74 (2008); PRL **102**, 058701 (2009).

Greedy routing based on local geometric information

- "**moving to the neighbor closest to the target**"-strategy
- limitation: it sometimes fails, due to the possibility of being trapped

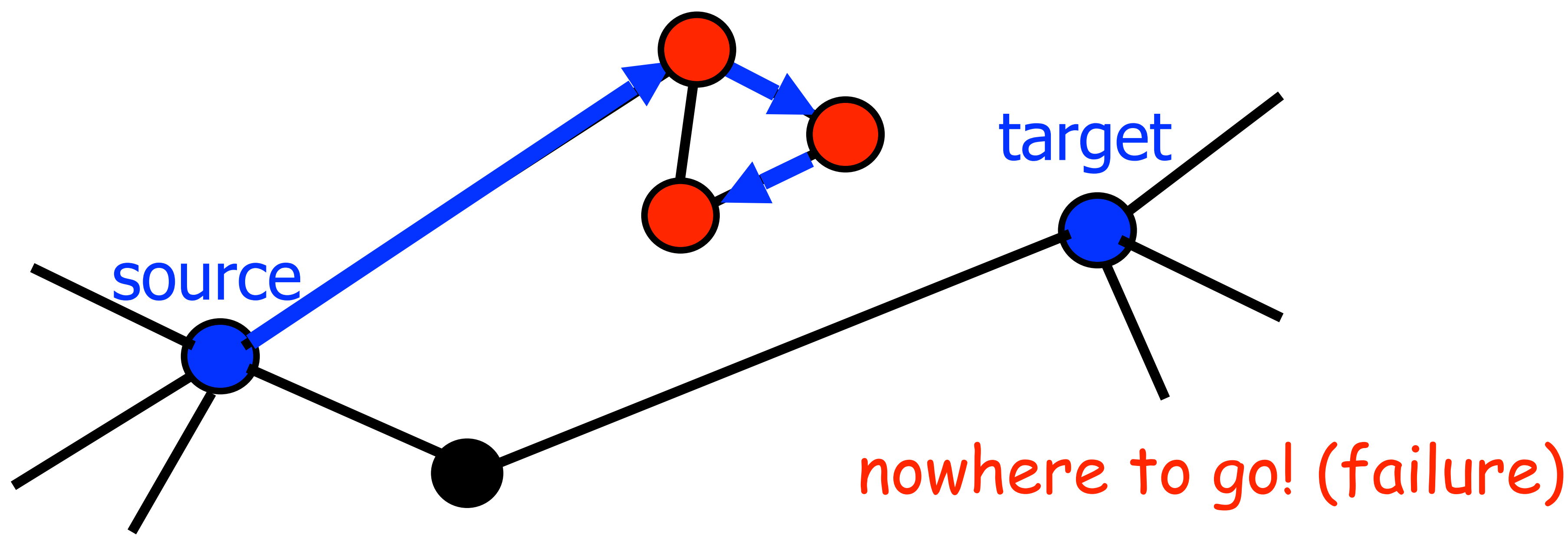


Note: look at the graph as shown in the 2D (Euclidean) space!

J.M. Kleinberg, Nature **406**, 845 (2000); M. Boguñá *et al.*, Nat. Phys. **5**, 74 (2008); PRL **102**, 058701 (2009).

Greedy routing based on local geometric information

- "**moving to the neighbor closest to the target**"-strategy
- limitation: it sometimes fails, due to the possibility of being trapped

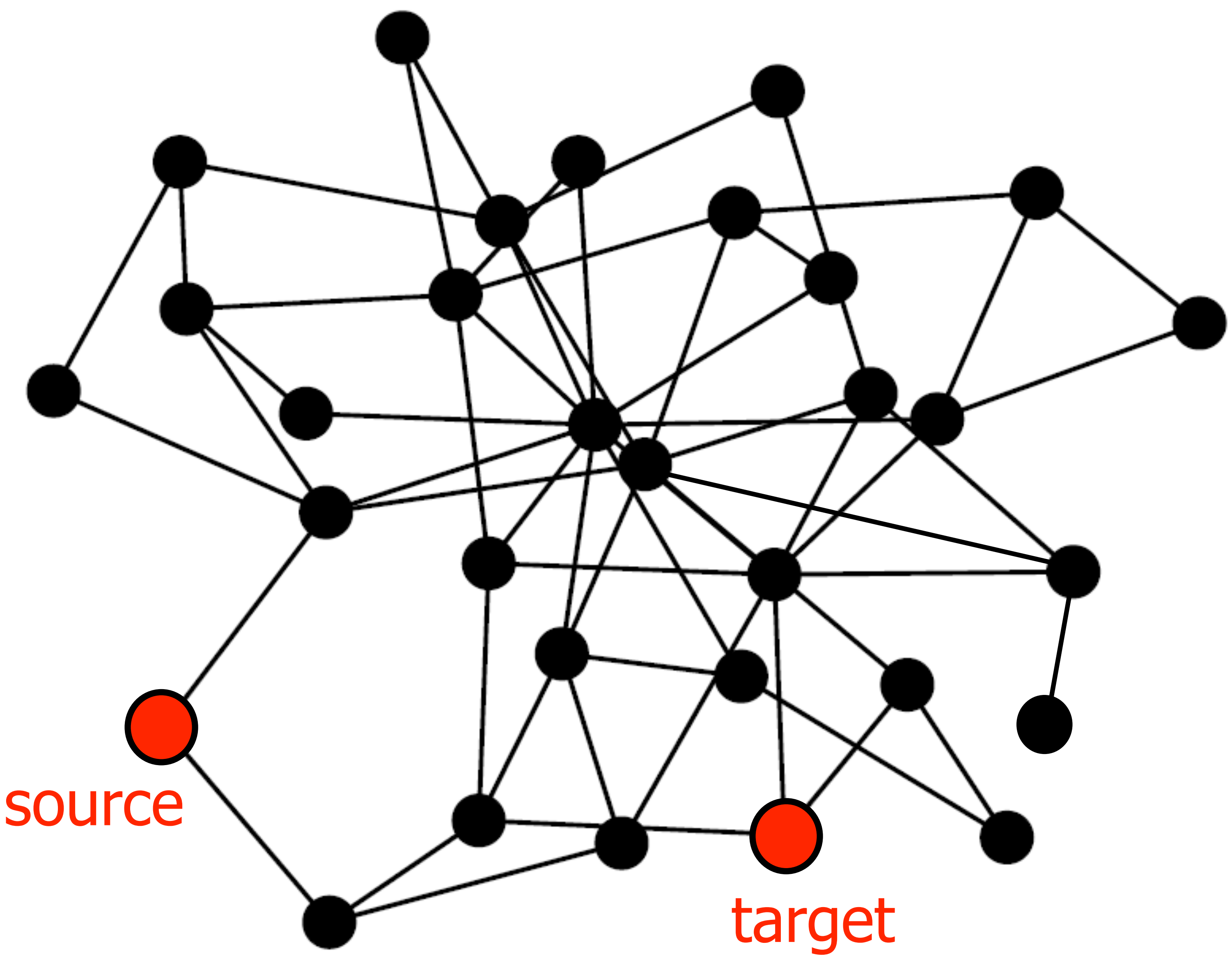


Note: look at the graph as shown in the 2D (Euclidean) space!

J.M. Kleinberg, Nature **406**, 845 (2000); M. Boguñá *et al.*, Nat. Phys. **5**, 74 (2008); PRL **102**, 058701 (2009).

Simple model using **geometric** information

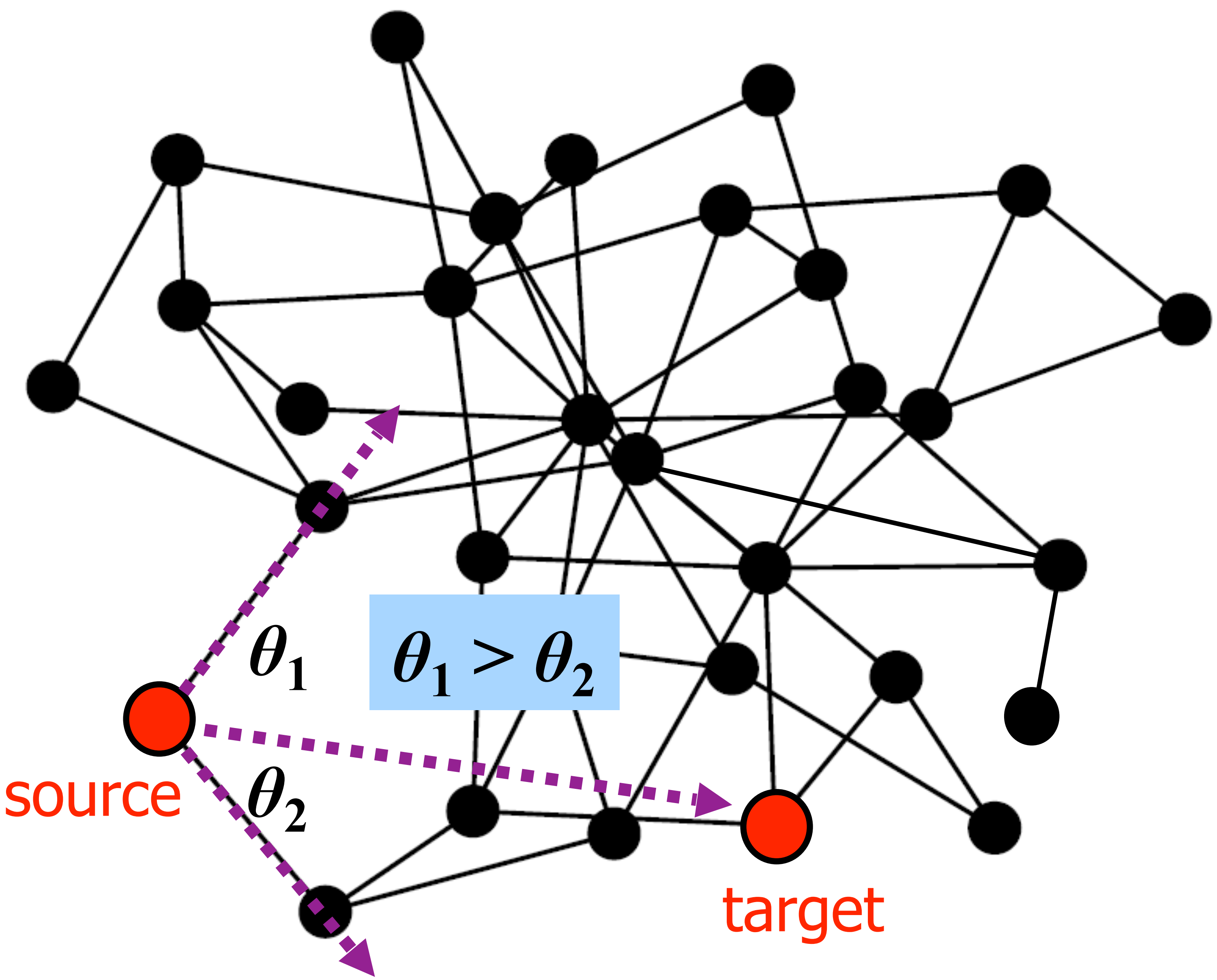
Greedy Spatial Navigation (GSN) protocol



Note: look at the graph as shown in the 2D (Euclidean) space!

Simple model using **geometric** information

Greedy Spatial Navigation (GSN) protocol

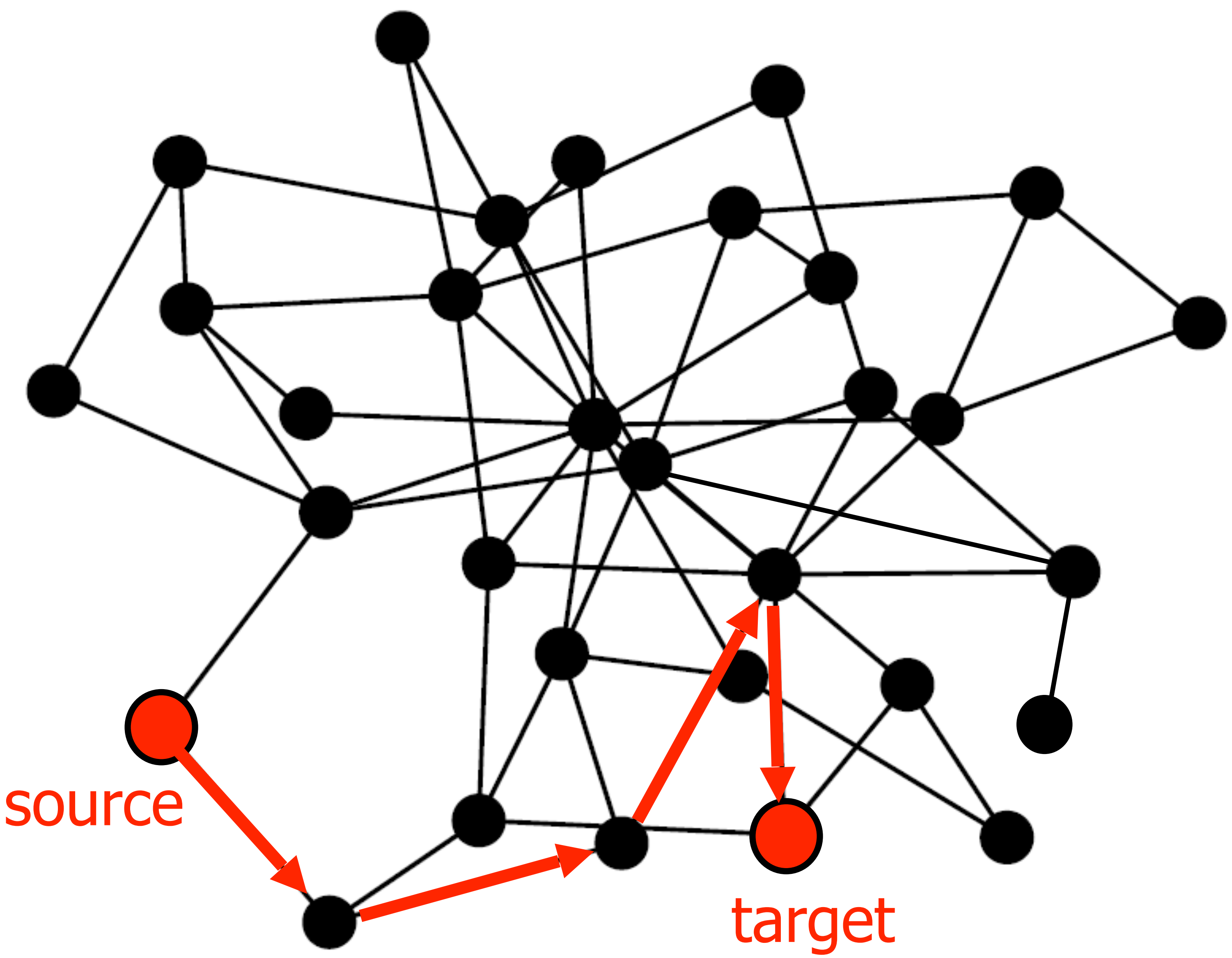


"biased" depth-first search (DFS) movement, based on direction: to the **unvisited** neighbor whose direction (from the current vertex) closest to the direction to the target!

Note: look at the graph as shown in the 2D (Euclidean) space!

Simple model using **geometric** information

Greedy Spatial Navigation (GSN) protocol

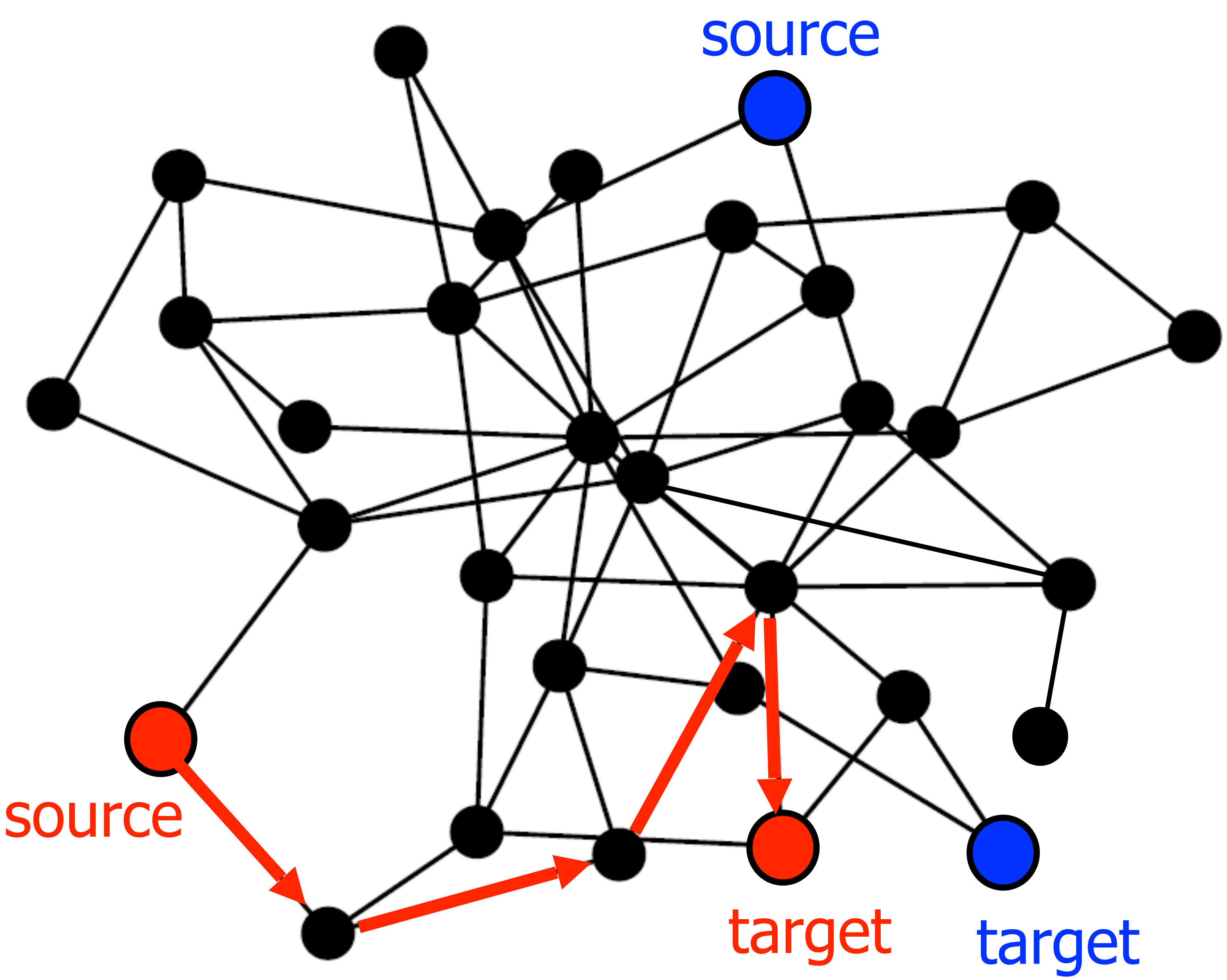


"biased" depth-first search (DFS) movement, based on direction: to the **unvisited** neighbor whose direction (from the current vertex) closest to the direction to the target!

Note: look at the graph as shown in the 2D (Euclidean) space!

Simple model using **geometric** information

Greedy Spatial Navigation (GSN) protocol



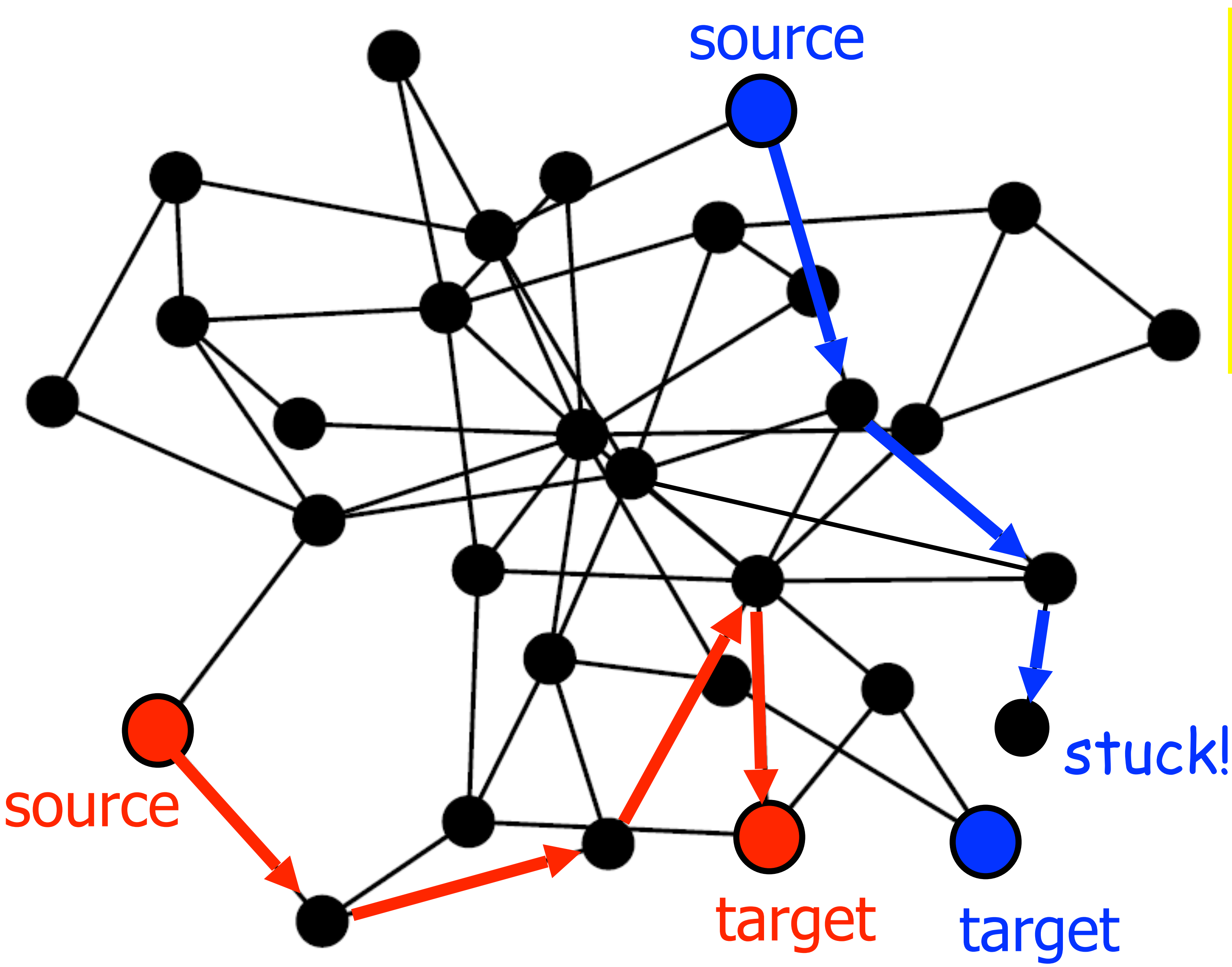
"biased" depth-first search (DFS) movement, based on direction: to the **unvisited** neighbor whose direction (from the current vertex) closest to the direction to the target!

"**backtracking**" to the deepest level above with an available vertex to step down on is possible, if it's stuck (finding the next-best way from there)
DFS ≠ self-avoiding walk

Note: look at the graph as shown in the 2D (Euclidean) space!

Simple model using **geometric** information

Greedy Spatial Navigation (GSN) protocol



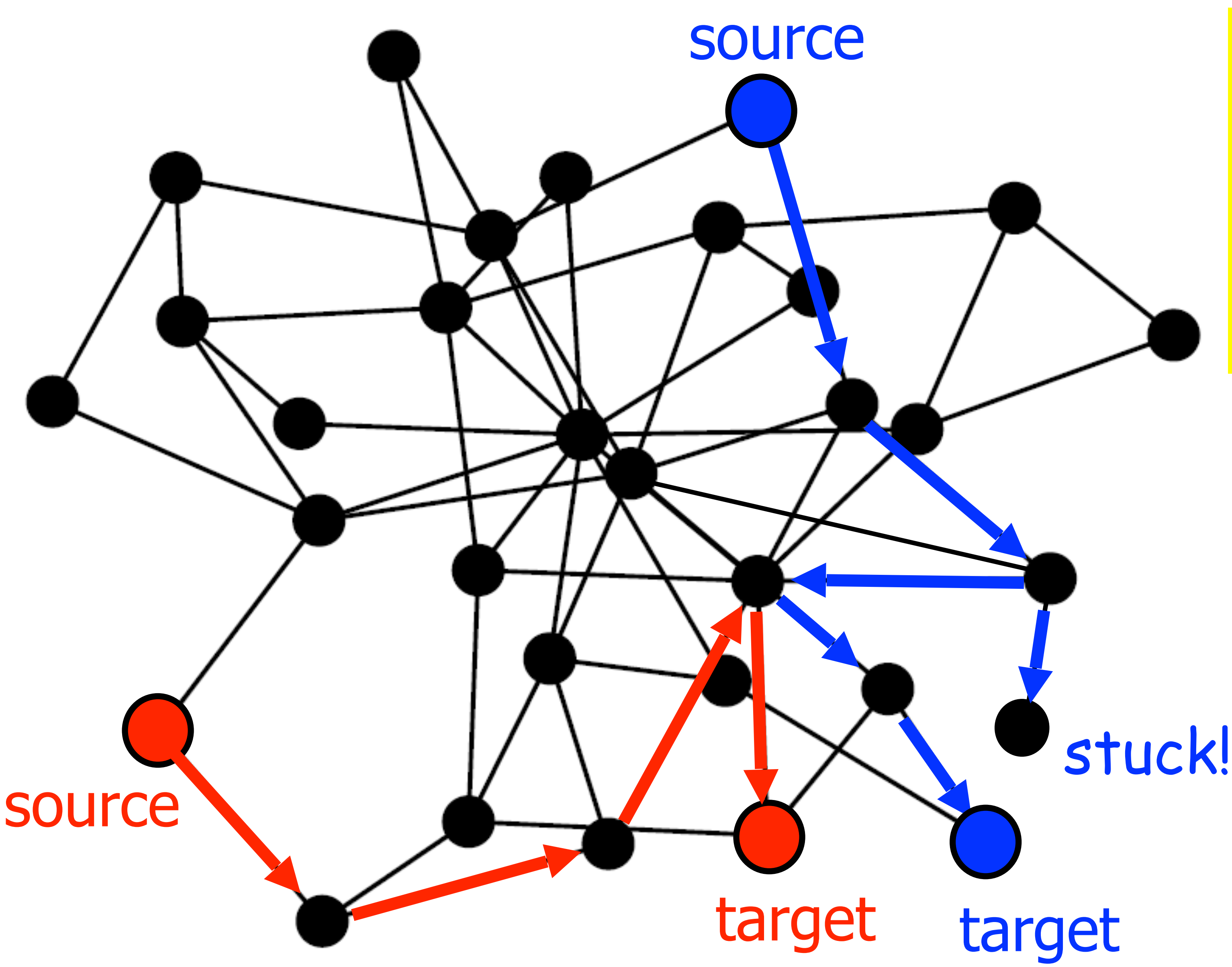
"biased" depth-first search (DFS) movement, based on direction: to the **unvisited** neighbor whose direction (from the current vertex) closest to the direction to the target!

"**backtracking**" to the deepest level above with an available vertex to step down on is possible, if it's stuck (finding the next-best way from there)
DFS ≠ self-avoiding walk

Note: look at the graph as shown in the 2D (Euclidean) space!

Simple model using **geometric** information

Greedy Spatial Navigation (GSN) protocol



"biased" depth-first search (DFS) movement, based on direction: to the **unvisited** neighbor whose direction (from the current vertex) closest to the direction to the target!

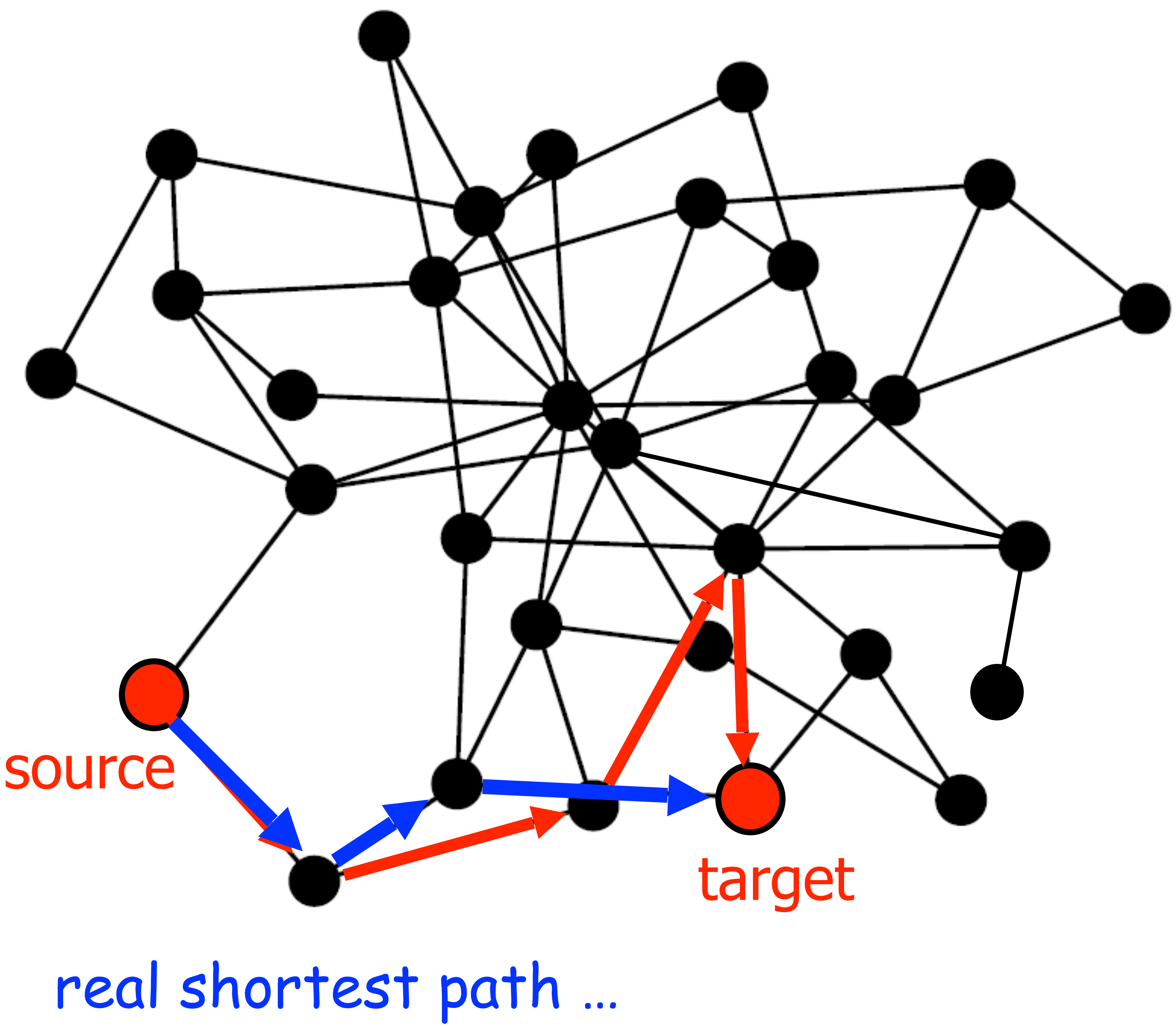
"**backtracking**" to the deepest level above with an available vertex to step down on is possible, if it's stuck (finding the next-best way from there)
DFS ≠ self-avoiding walk

→ greedy but (and?) smart navigator!

Note: look at the graph as shown in the 2D (Euclidean) space!

Simple model using **geometric** information

Greedy Spatial Navigation (GSN) protocol



"biased" depth-first search (DFS) movement, based on direction: to the **unvisited** neighbor whose direction (from the current vertex) closest to the direction to the target!

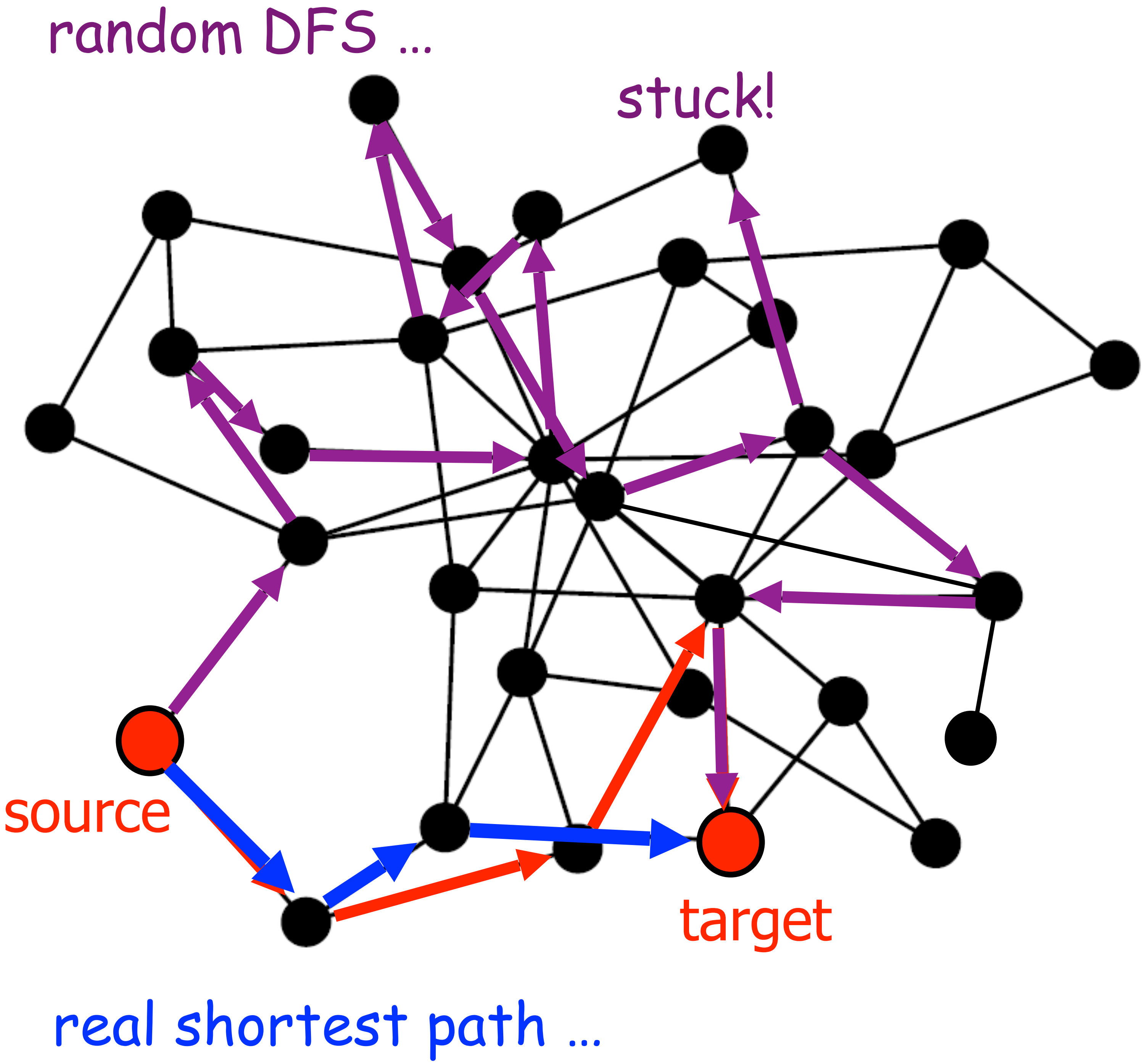
"**backtracking**" to the deepest level above with an available vertex to step down on is possible, if it's stuck (finding the next-best way from there)
DFS ≠ self-avoiding walk

→ greedy but (and?) smart navigator!

Note: look at the graph as shown in the 2D (Euclidean) space!

Simple model using **geometric** information

Greedy Spatial Navigation (GSN) protocol



"biased" depth-first search (DFS) movement, based on direction: to the **unvisited** neighbor whose direction (from the current vertex) closest to the direction to the target!

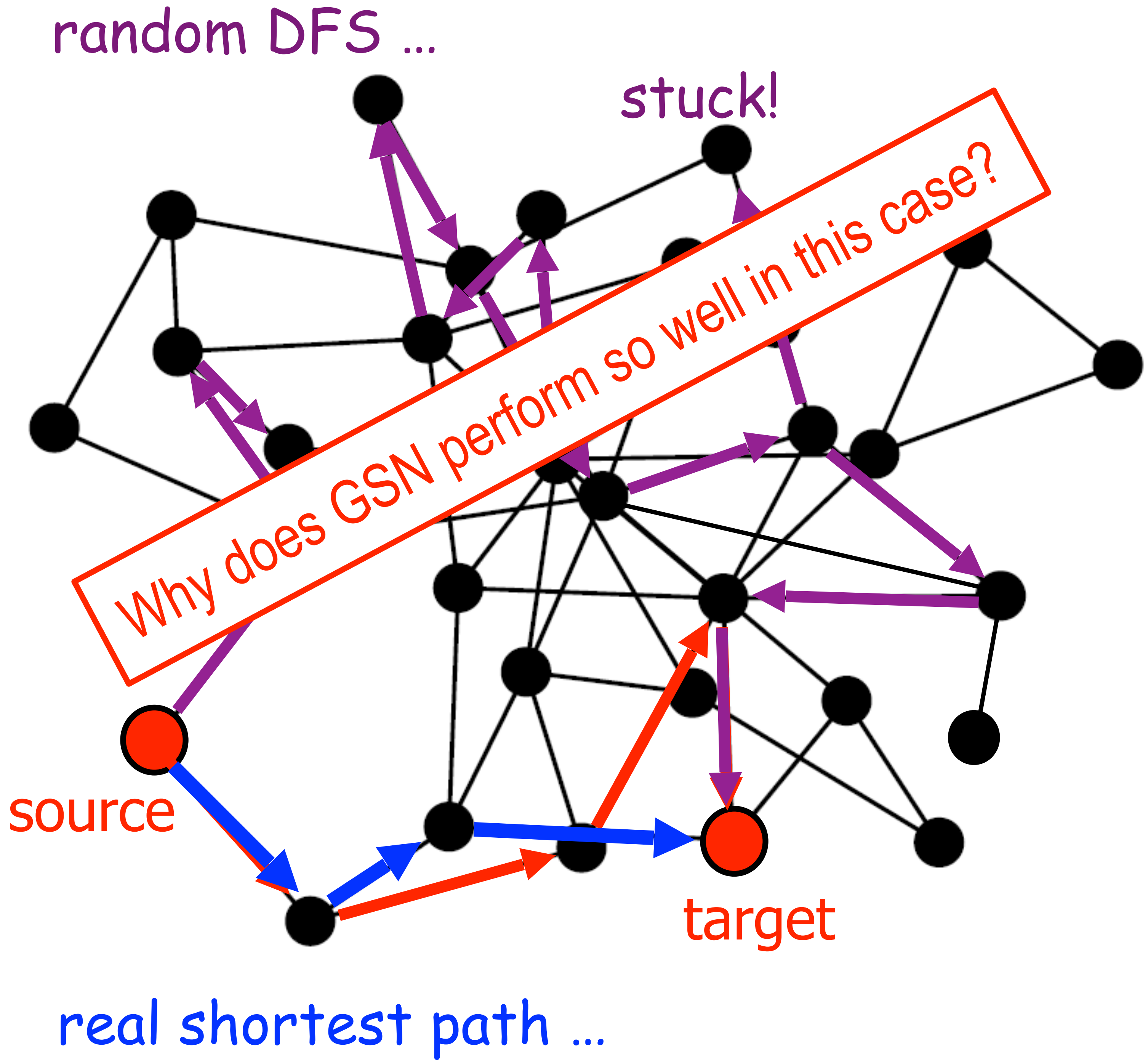
"**backtracking**" to the deepest level above with an available vertex to step down on is possible, if it's stuck (finding the next-best way from there)
DFS ≠ self-avoiding walk

→ greedy but (and?) smart navigator!

Note: look at the graph as shown in the 2D (Euclidean) space!

Simple model using **geometric** information

Greedy Spatial Navigation (GSN) protocol



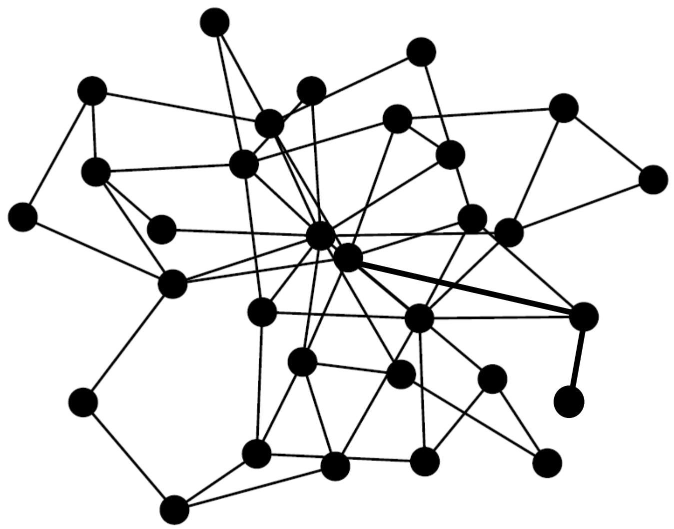
"biased" depth-first search (DFS) movement, based on direction: to the **unvisited** neighbor whose direction (from the current vertex) closest to the direction to the target!

"**backtracking**" to the deepest level above with an available vertex to step down on is possible, if it's stuck (finding the next-best way from there)
DFS ≠ self-avoiding walk

→ greedy but (and?) smart navigator!

Note: look at the graph as shown in the 2D (Euclidean) space!

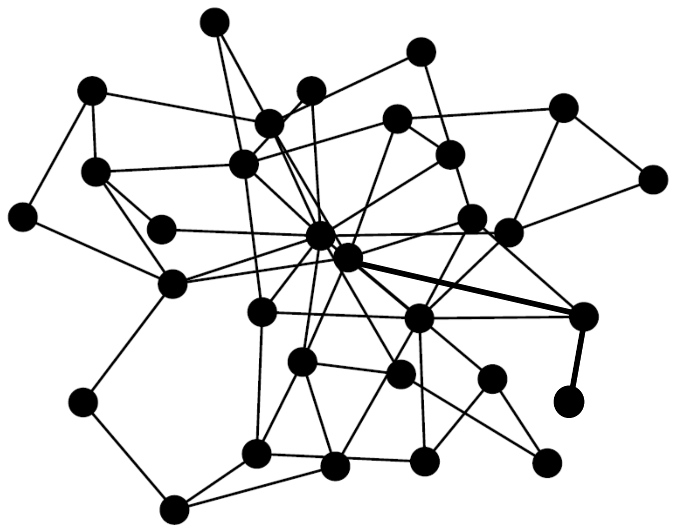
Secret of success: graph visualization technique



→ Kamada-Kawai (KK) spring-based graph layout

It looks good! (to human eyes)

Secret of success: graph visualization technique

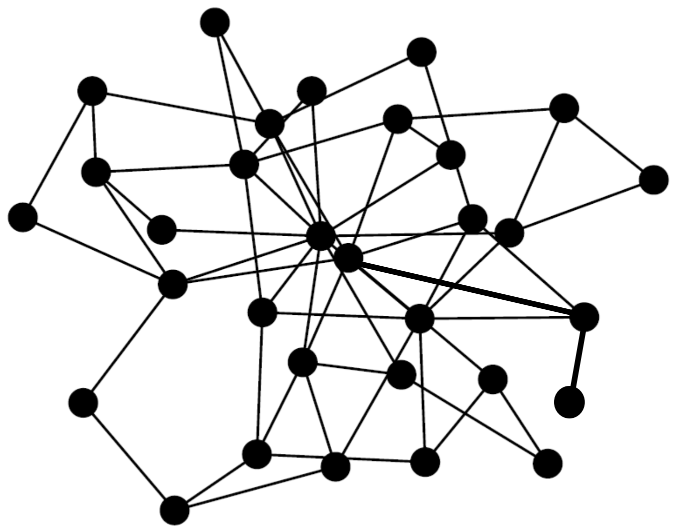


→ Kamada-Kawai (KK) spring-based graph layout

It looks good! (to human eyes)

side(?) effect: vertices closer in graphs tend to be located closer in (2D) geometric space as well!

Secret of success: graph visualization technique



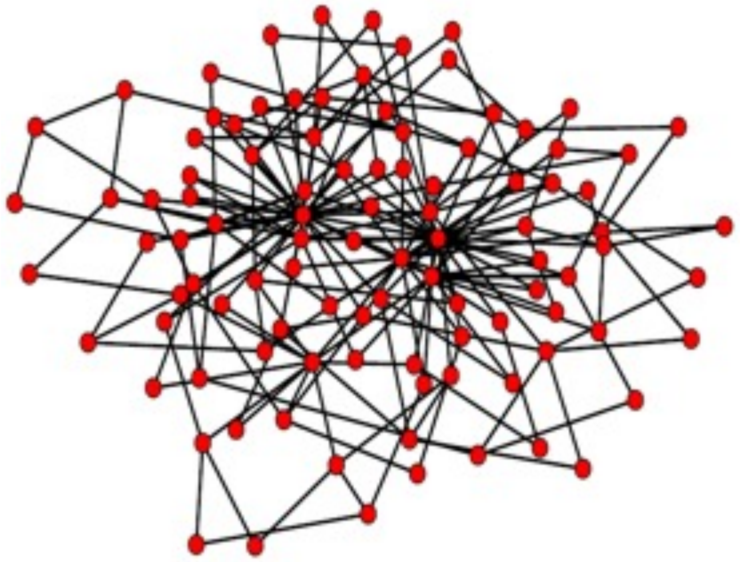
→ Kamada-Kawai (KK) spring-based graph layout

It looks good! (to human eyes)

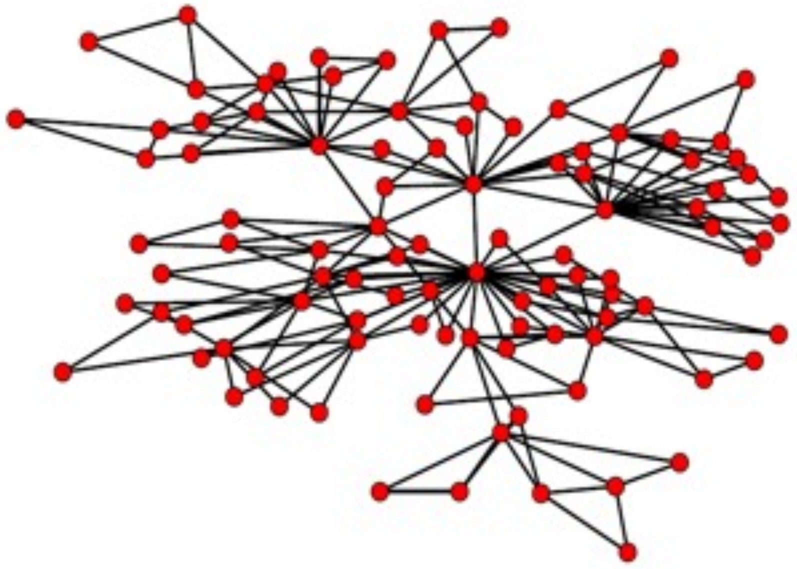
side(?) effect: vertices closer in graphs tend to be located closer in (2D) geometric space as well!

So, does the layout algorithm really help GSN?

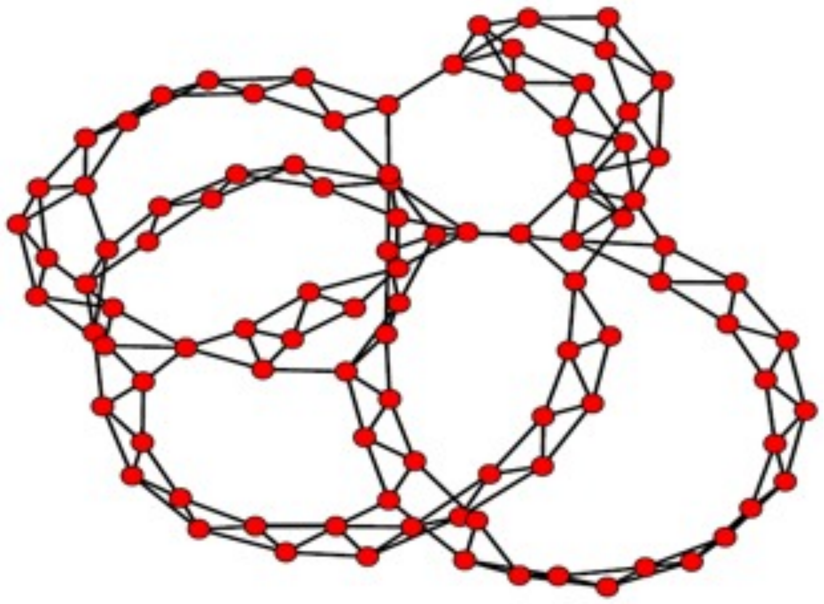
Barabási-Albert model
(purely topological)



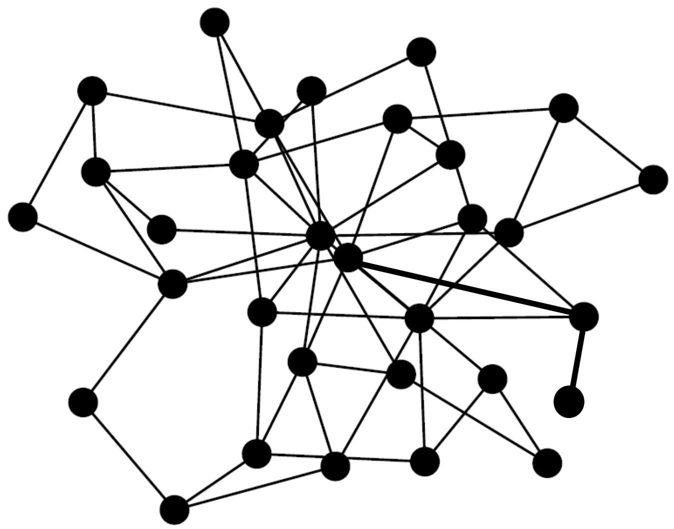
Holme-Kim model
(topological, with
non-vanishing clustering)



Watts-Strogatz model
(based on 1D ring)



Secret of success: graph visualization technique



→ Kamada-Kawai (KK) spring-based graph layout

It looks good! (to human eyes)

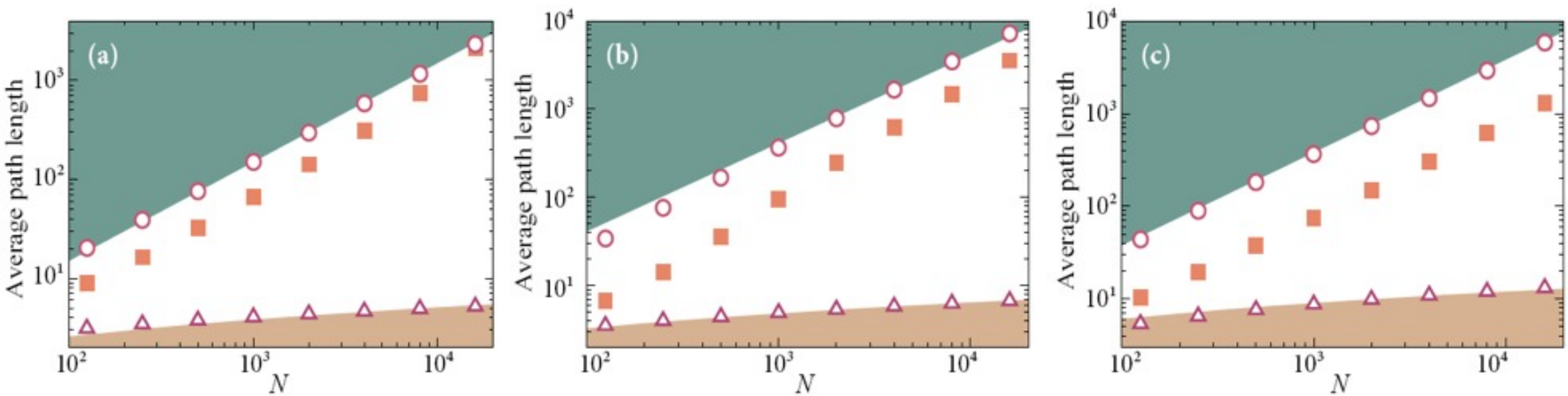
side(?) effect: vertices closer in graphs tend to be located closer in (2D) geometric space as well!

So, does the layout algorithm really help GSN?

Barabási-Albert model (purely topological)

Holme-Kim model (topological, with non-vanishing clustering)

Watts-Strogatz model (based on 1D ring)

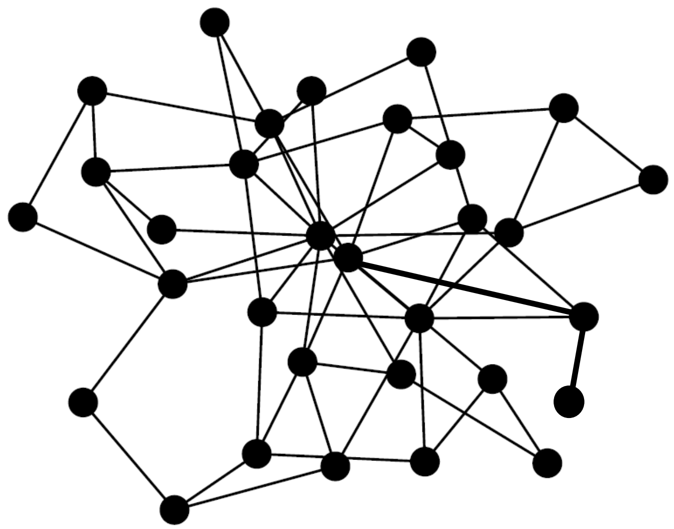


■ **biased Depth-First Search**

○ Random Depth-First Search

△ Topologically Shortest Path

Secret of success: graph visualization technique



→ Kamada-Kawai (KK) spring-based graph layout

It looks good! (to human eyes)

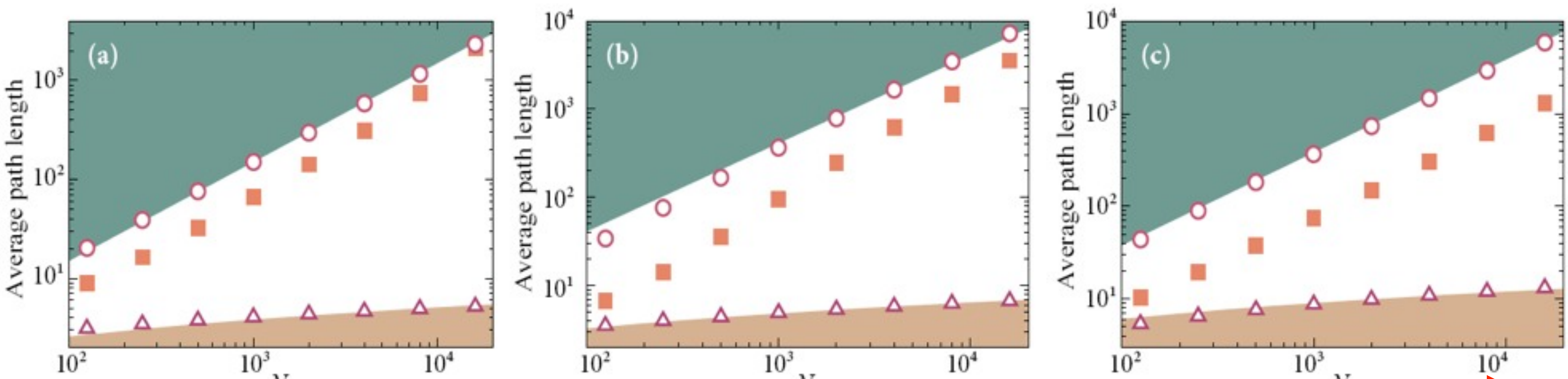
side(?) effect: vertices closer in graphs tend to be located closer in (2D) geometric space as well!

So, does the layout algorithm really help GSN?

Barabási-Albert model (purely topological)

Holme-Kim model (topological, with non-vanishing clustering)

Watts-Strogatz model (based on 1D ring)

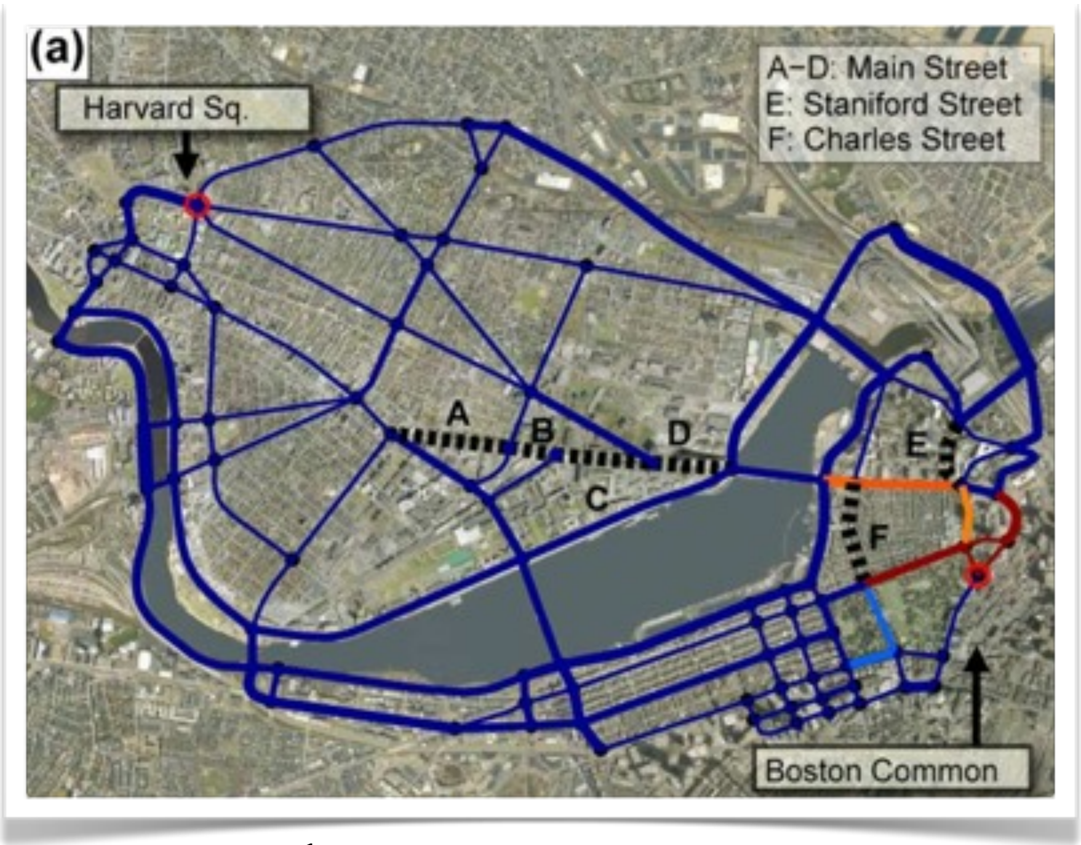


"exploitable" geometric information increased

■ **biased Depth-First Search**
○ Random Depth-First Search
△ Topologically Shortest Path

SHL and P. Holme, Physica A **390**, 3996 (2011).

Application to real spatial networks

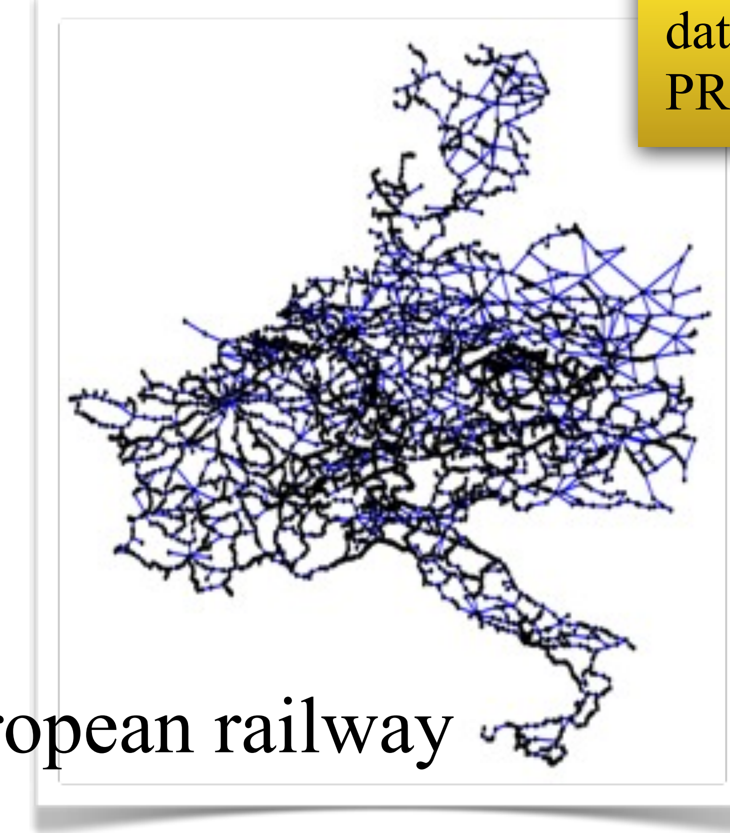


Boston road

New York road

data from H. Youn *et al.*, PRL **101**, 128701 (2008)

Switzerland railway



European railway

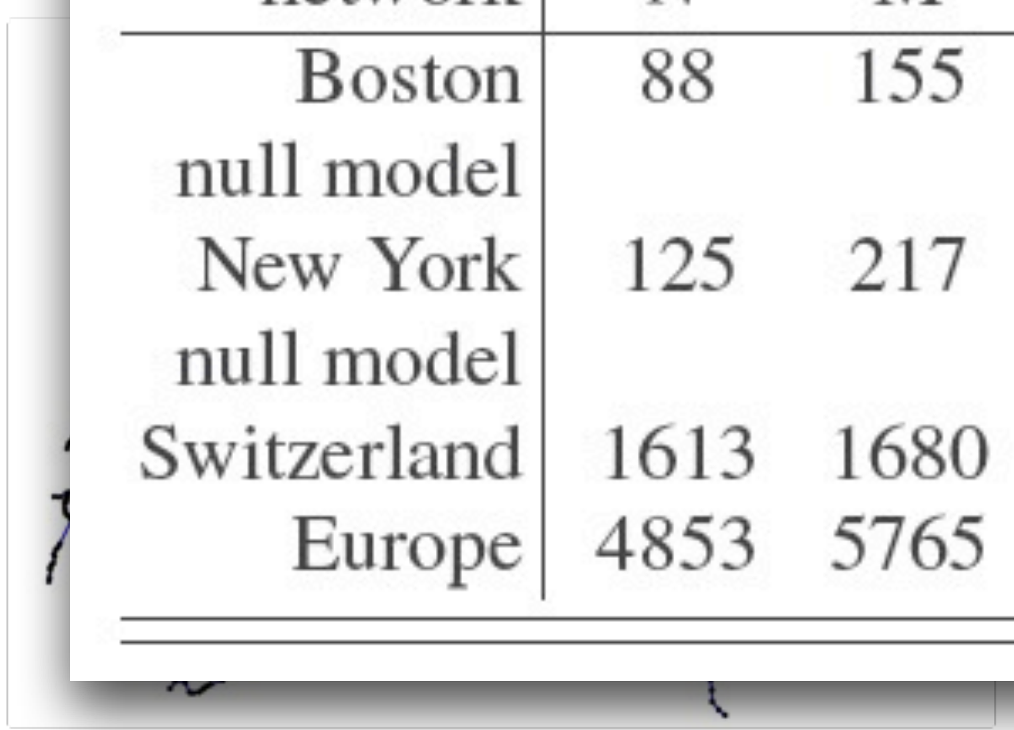
data from M. Kurant *et al.*, PRL **96**, 138701 (2006)

Application to real spatial networks

TABLE I: Properties of four empirical datasets. Performance of routing strategies for road and railway networks. For each network, the number of vertices N, the number of edges M, the average path length for GSN strategy d_g, real shortest path d, random DFS d_r, and navigability $\nu = d/d_g$ are shown in each column. Null models for Boston and New York roads are connected Erdős-Rényi random graphs [17] with the same N and M, where the geographic layout is given by Kamada-Kawai algorithm [15], and the results averaged over 10^3 samples are shown.

network	N	M	d_g	d	d_r	ν
Boston	88	155	6.82	5.72	30.75	84%
null model			8.606(9)	23.20(1)	3.6758(1)	37 %
New York	125	217	8.27	6.79	44.39	82%
null model			11.72(2)	33.51(2)	4.0300(1)	34 %
Switzerland	1613	1680	145.14	46.56	769.68	32%
Europe	4853	5765	143.69	50.87	2011.93	35%

Bosto



European railway

Switzerland railway

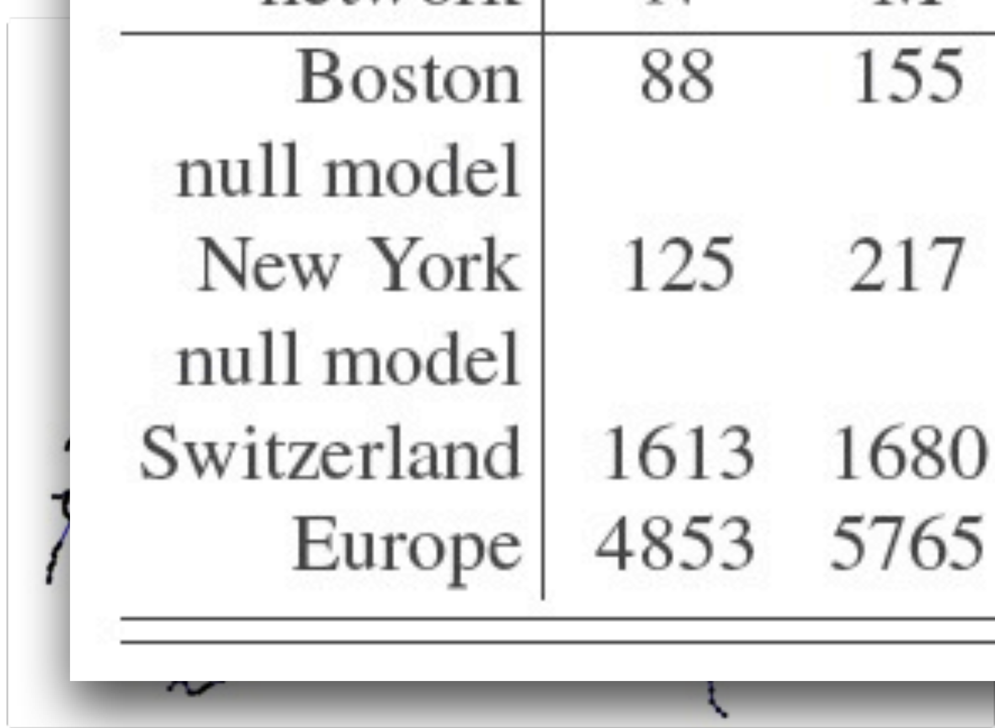
Application to real spatial networks

(a)

Bosto

TABLE I: Properties of four empirical datasets. Performance of routing strategies for road and railway networks. For each network, the number of vertices N, the number of edges M, the average path length for GSN strategy d_g, real shortest path d, random DFS d_r, and navigability $\nu = d/d_g$ are shown in each column. Null models for Boston and New York roads are connected Erdős-Rényi random graphs [17] with the same N and M, where the geographic layout is given by Kamada-Kawai algorithm [15], and the results averaged over 10^3 samples are shown.

network	N	M	d_g	d	d_r	ν
Boston	88	155	6.82	5.72	30.75	84%
null model			8.606(9)	23.20(1)	3.6758(1)	37 %
New York	125	217	8.27	6.79	44.39	82%
null model			11.72(2)	33.51(2)	4.0300(1)	34 %
Switzerland	1613	1680	145.14	46.56	769.68	32%
Europe	4853	5765	143.69	50.87	2011.93	35%



Switzerland railway

European railway

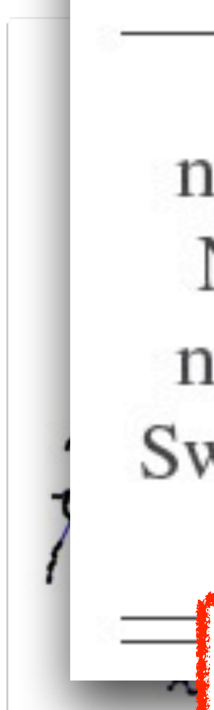
Application to real spatial networks

(a)

Bost

TABLE I: Properties of four empirical datasets. Performance of routing strategies for road and railway networks. For each network, the number of vertices N, the number of edges M, the average path length for GSN strategy d_g, real shortest path d, random DFS d_r, and navigability $\nu = d/d_g$ are shown in each column. Null models for Boston and New York roads are connected Erdős-Rényi random graphs [17] with the same N and M, where the geographic layout is given by Kamada-Kawai algorithm [15], and the results averaged over 10^3 samples are shown.

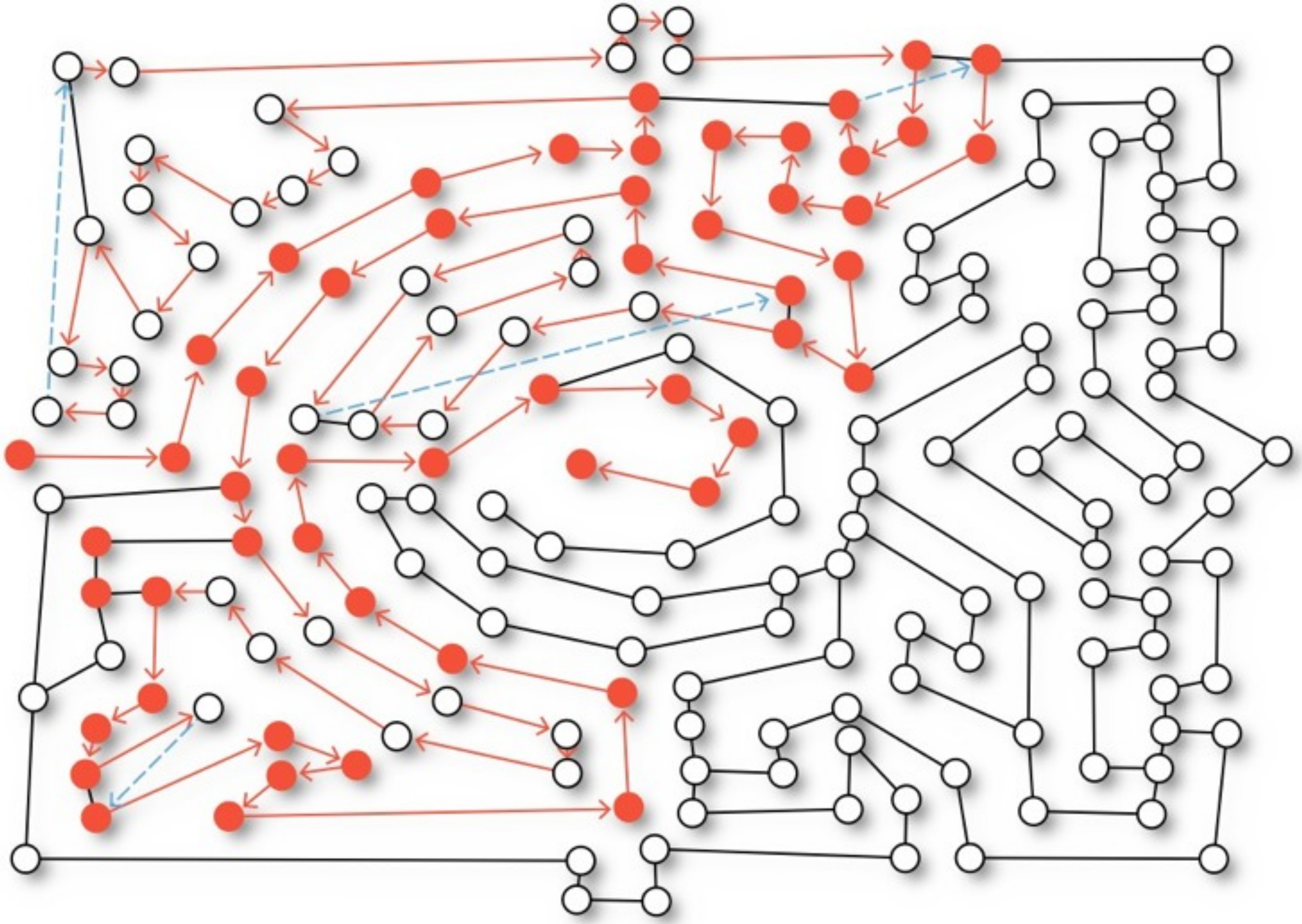
network	N	M	d_g	d	d_r	ν
Boston	88	155	6.82	5.72	30.75	84%
null model			8.606(9)	23.20(1)	3.6758(1)	37 %
New York	125	217	8.27	6.79	44.39	82%
null model			11.72(2)	33.51(2)	4.0300(1)	34 %
Switzerland	1613	1680	145.14	46.56	769.68	32%
Europe	4853	5765	143.69	50.87	2011.93	35%



quite efficient strategy for these real transport networks!

Switzerland railway

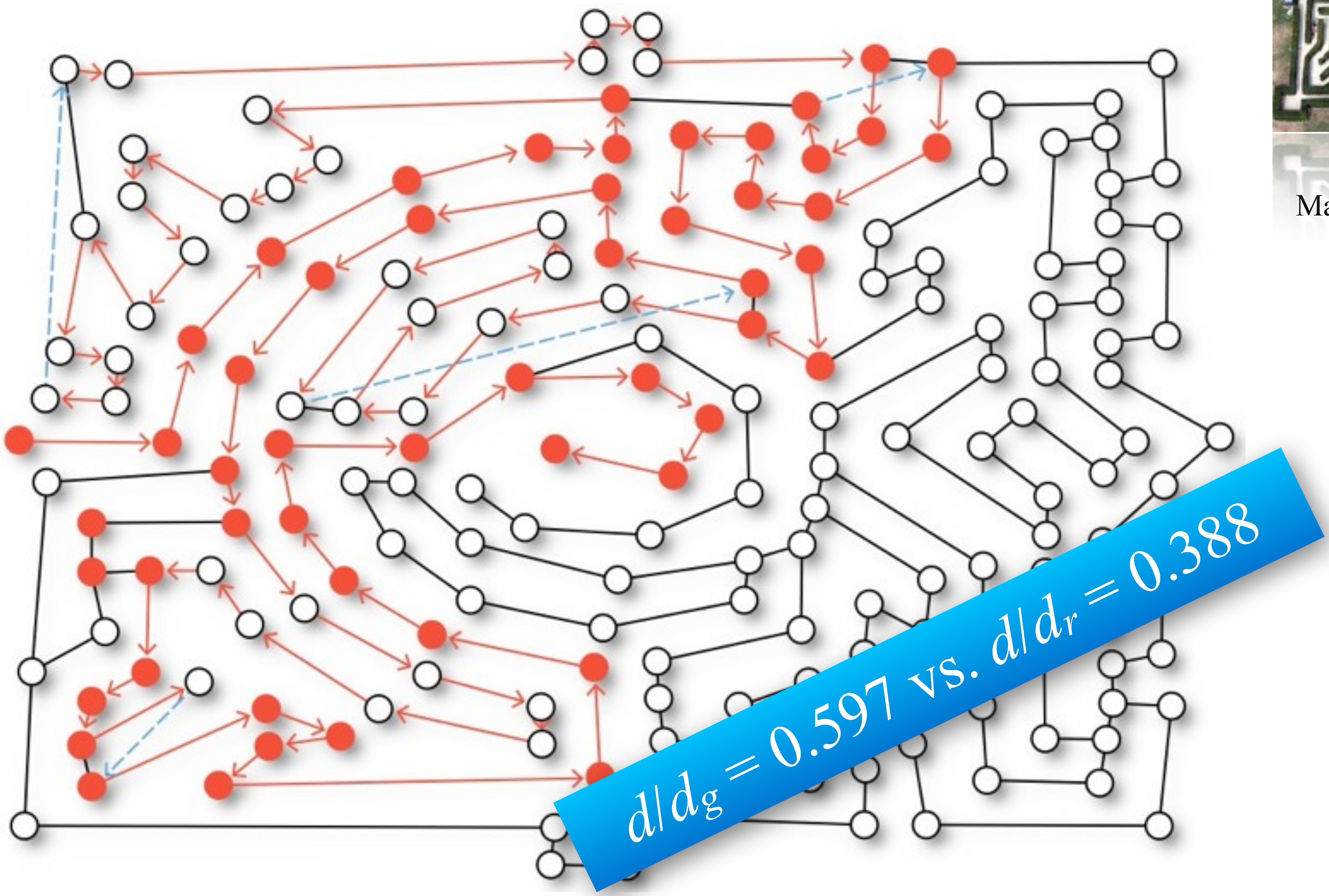
GSN works in a maze!



Maze in Leeds Castle, Kent, England

real shortest path shown in filled vertices (d = **52** steps)
GSN pathway (d_g = **87** steps) shown in arrows
average random DFS pathway (d_r = 134(1) steps)

GSN works in a maze!



Maze in Leeds Castle, Kent, England

real shortest path shown in filled vertices (d = **52** steps)

GSN pathway (d_g = **87** steps) shown in arrows

average random DFS pathway (d_r = 134(1) steps)

New centrality based on GSN:
Navigator Centrality n for vertex/edge

vertex **Navigator Centrality**

$$n(v) \sim \sum_{i \neq j} \sigma_{ivj}^{V}$$

$$\sigma_{ivj}^{V} = 1 \text{ if GSN path goes from } i \text{ to } j \text{ via } v, 0 \text{ otherwise}$$

edge **Navigator Centrality**

$$n(l) \sim \sum_{i \neq j} \sigma_{ilj}^{E}$$

$$\sigma_{ilj}^{E} = 1 \text{ if GSN path goes from } i \text{ to } j \text{ via } l, 0 \text{ otherwise}$$

New centrality based on GSN:

Navigator Centrality n for vertex/edge

vertex **Betweenness** **Centrality**

$$b(v) \sim \sum_{i \neq j} \sigma^{V}_{ivj} / \sigma^{V}_{ij}$$

$\sigma^{V}_{ivj} = 1$ if **shortest path** goes from i to j via v, 0 otherwise

$\sigma^{V}_{ij} = \#$ of shortest paths going from i to j

edge **Betweenness** **Centrality**

$$b(l) \sim \sum_{i \neq j} \sigma^{E}_{ilj} / \sigma^{V}_{ij}$$

$\sigma^{E}_{ilj} = 1$ if **shortest path** goes from i to j via l, 0 otherwise

New centrality based on GSN:
Navigator Centrality n for vertex/edge

vertex **Navigator Centrality**

$$n(v) \sim \sum_{i \neq j} \sigma_{ivj}^{V}$$

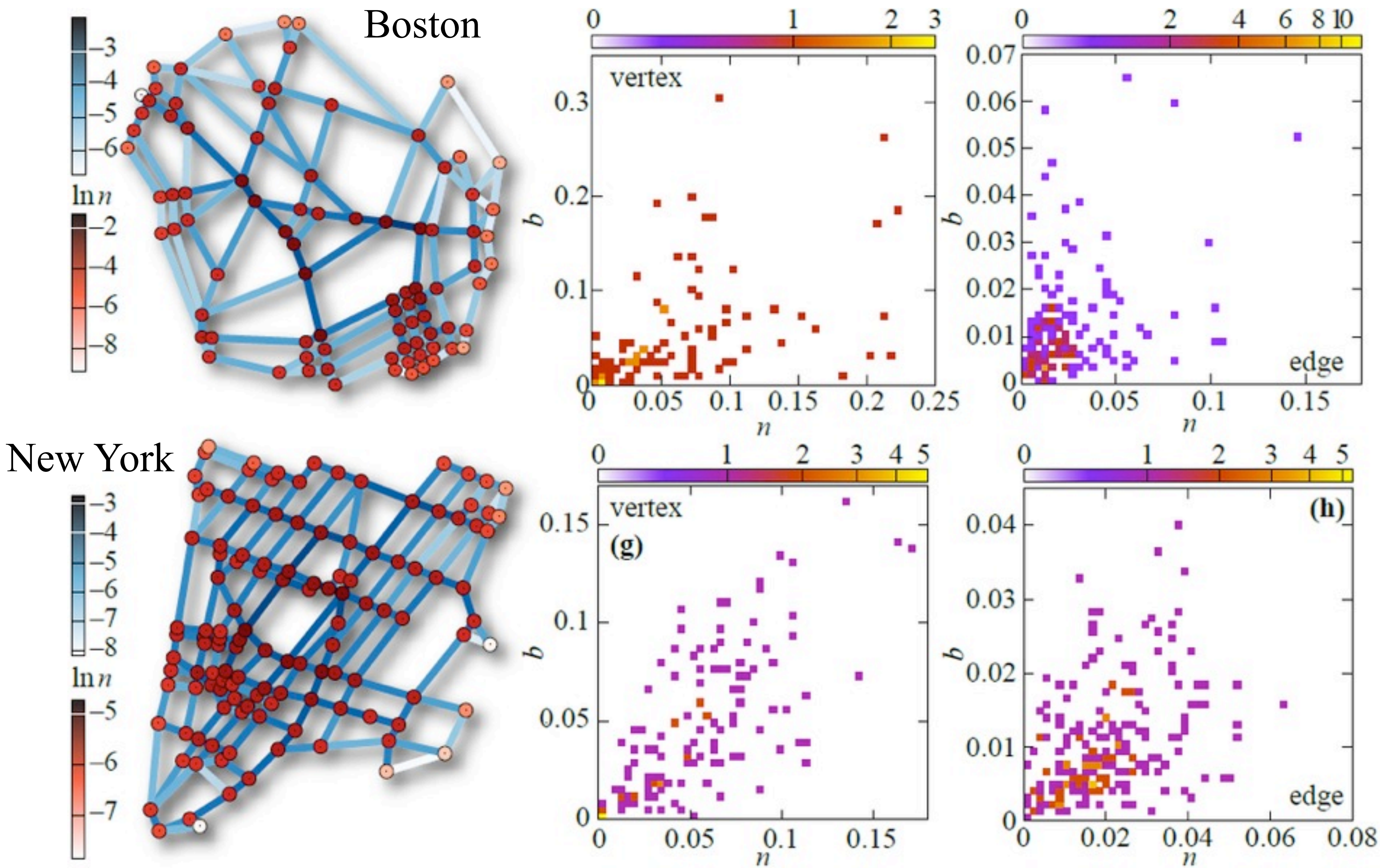
$\sigma_{ivj}^{V} = 1$ if GSN path goes from i to j via v, 0 otherwise

edge **Navigator Centrality**

$$n(l) \sim \sum_{i \neq j} \sigma_{ilj}^{E}$$

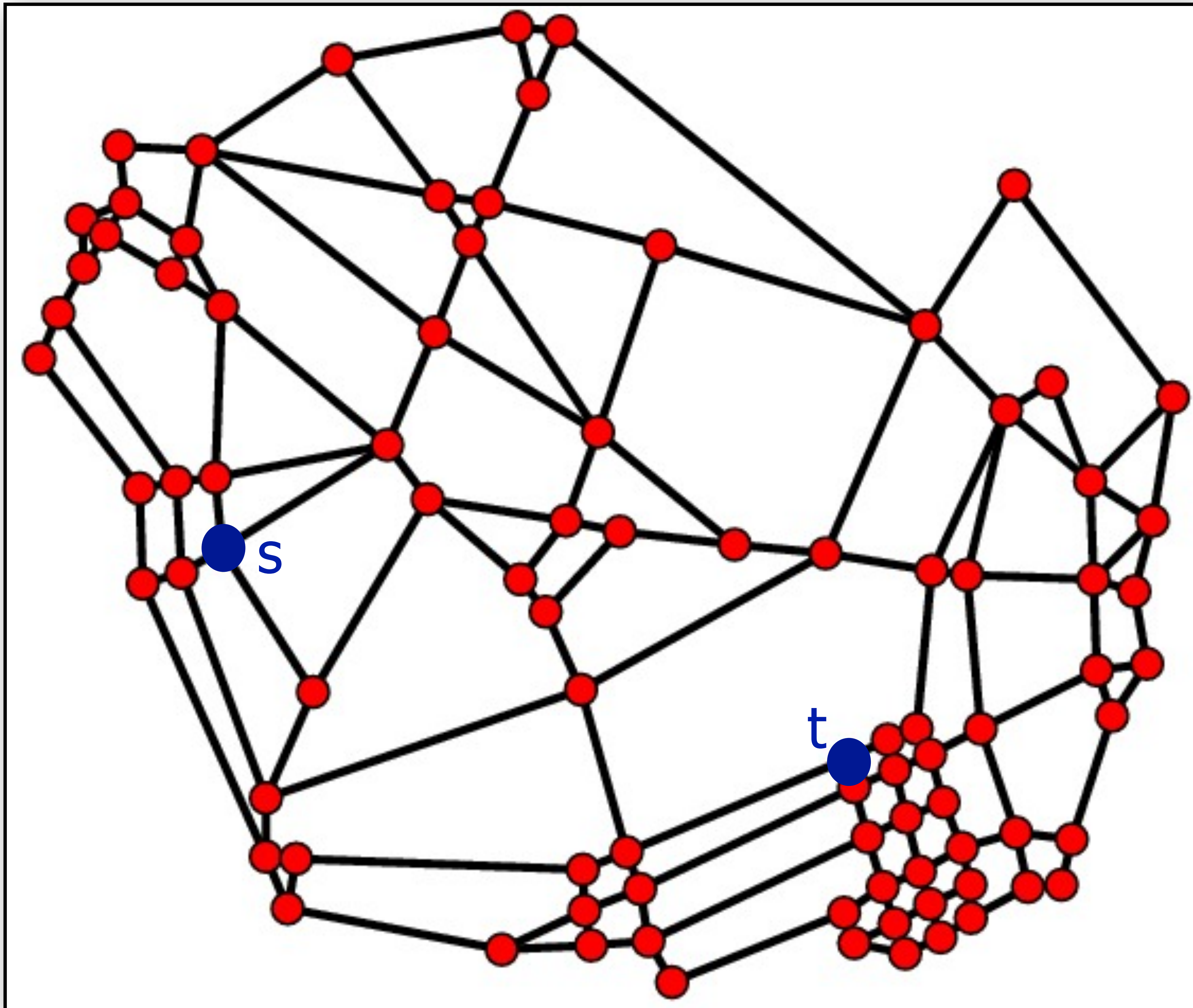
$\sigma_{ilj}^{E} = 1$ if GSN path goes from i to j via l, 0 otherwise

Navigator centralities for roads

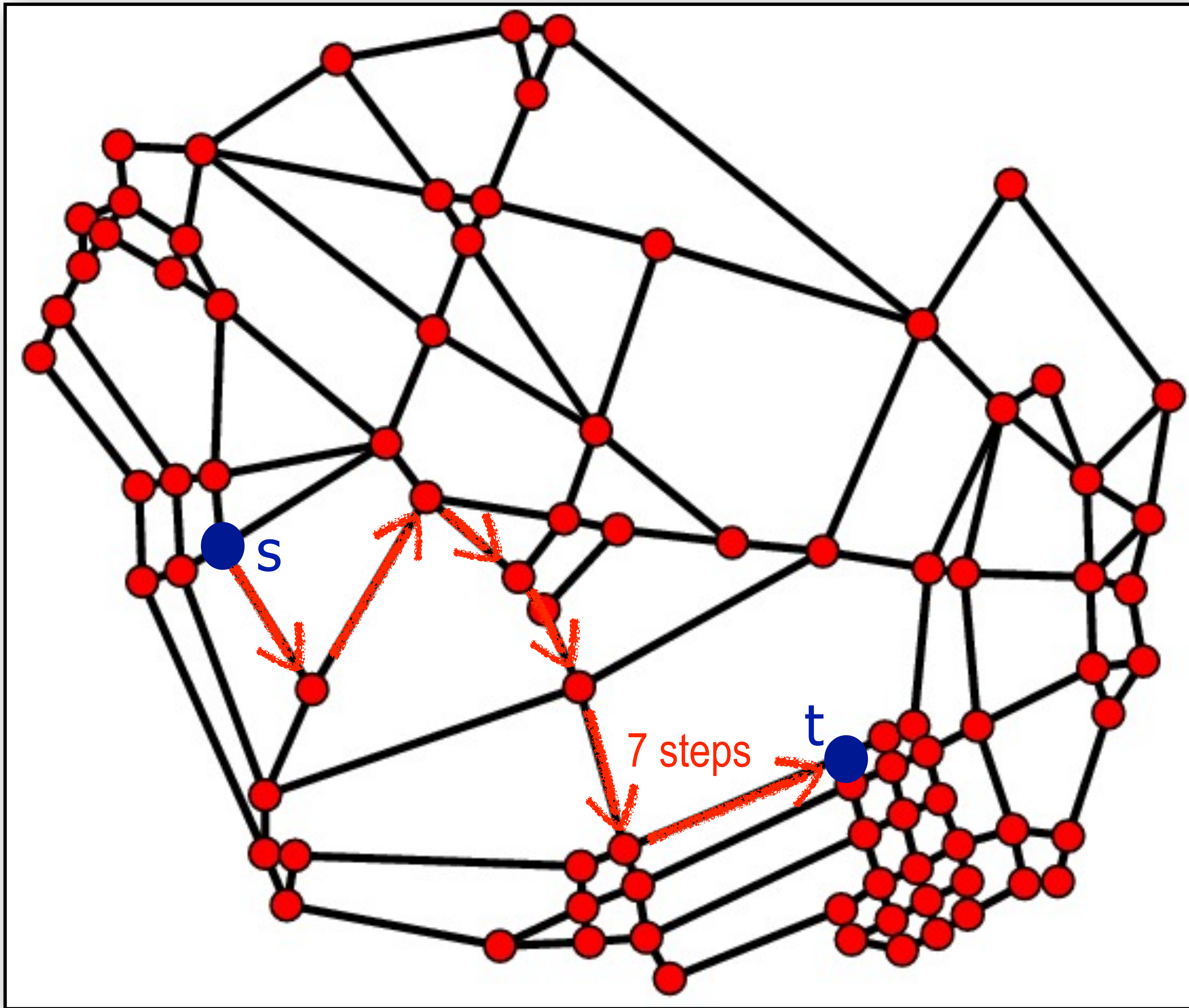


navigator centrality vs betweenness centrality: “vertex/edge profile”

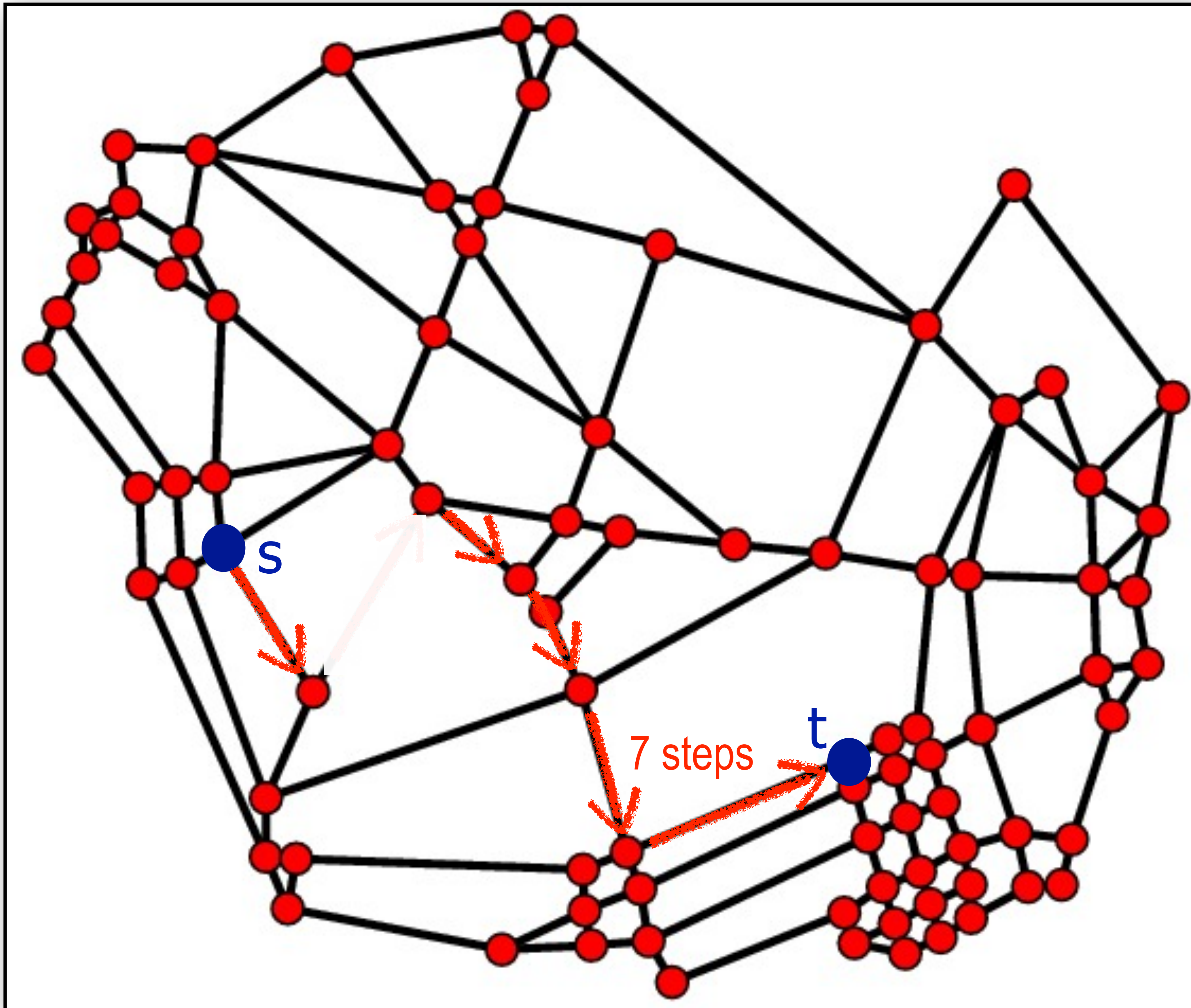
Counterintuitive phenomenon caused by greedy navigators



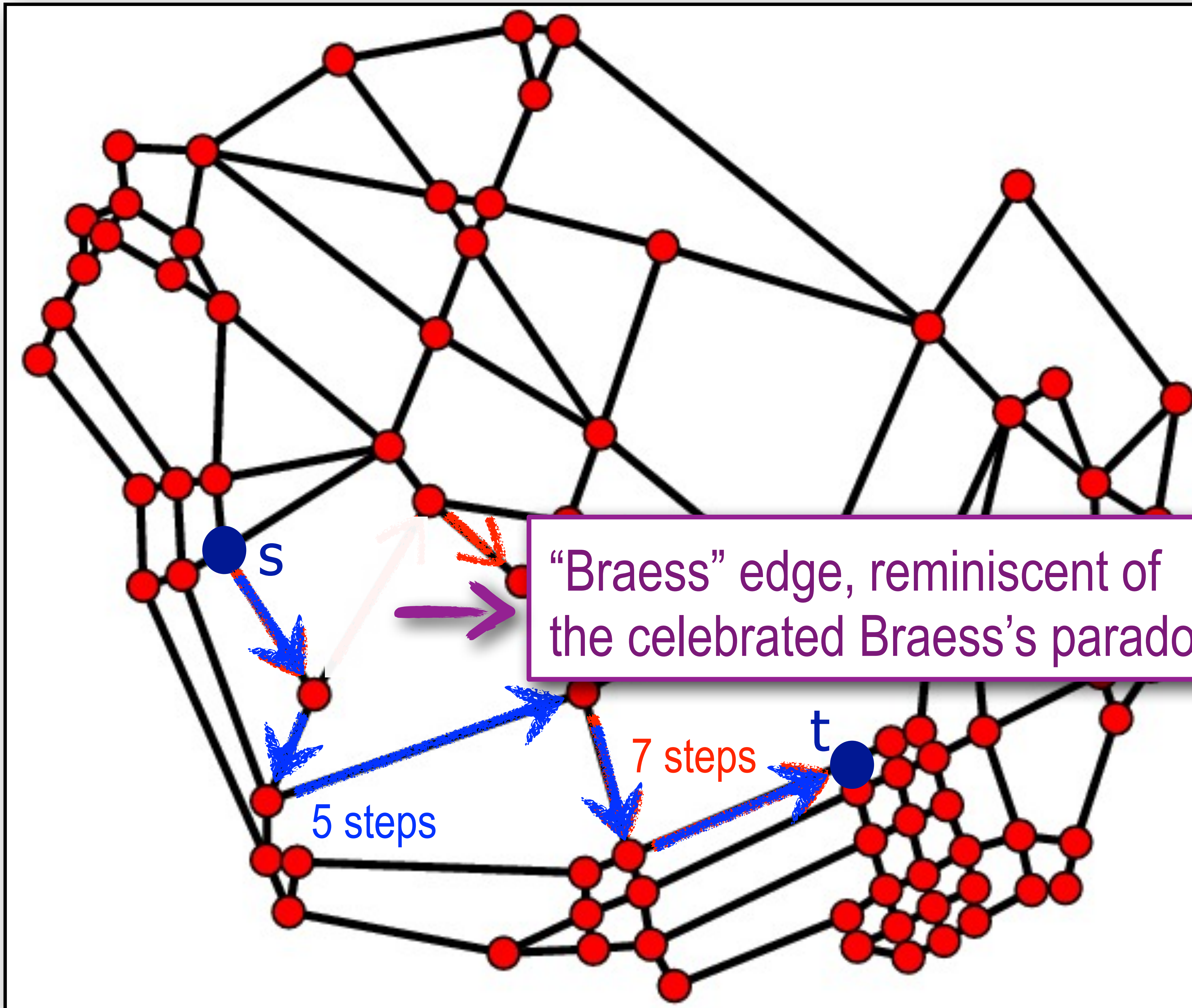
Counterintuitive phenomenon caused by greedy navigators



Counterintuitive phenomenon caused by greedy navigators



Counterintuitive phenomenon caused by greedy navigators



"Braess" edge, reminiscent of the celebrated Braess's paradox

H. Youn, M. Gastner, and H. Jeong, Phys. Rev. Lett. **101**, 128701 (2008).

Quantifying this property: **edge Essentiality** e for edge

edge **Essentiality**

$$e(l) = \mathrm{aGPL}(G \setminus \{l\}) - \mathrm{aGPL}(G)$$

$$\text{where aGPL} = \text{average GSN path length}$$

Quantifying this property: **edge Essentiality** e for edge

edge **Essentiality**

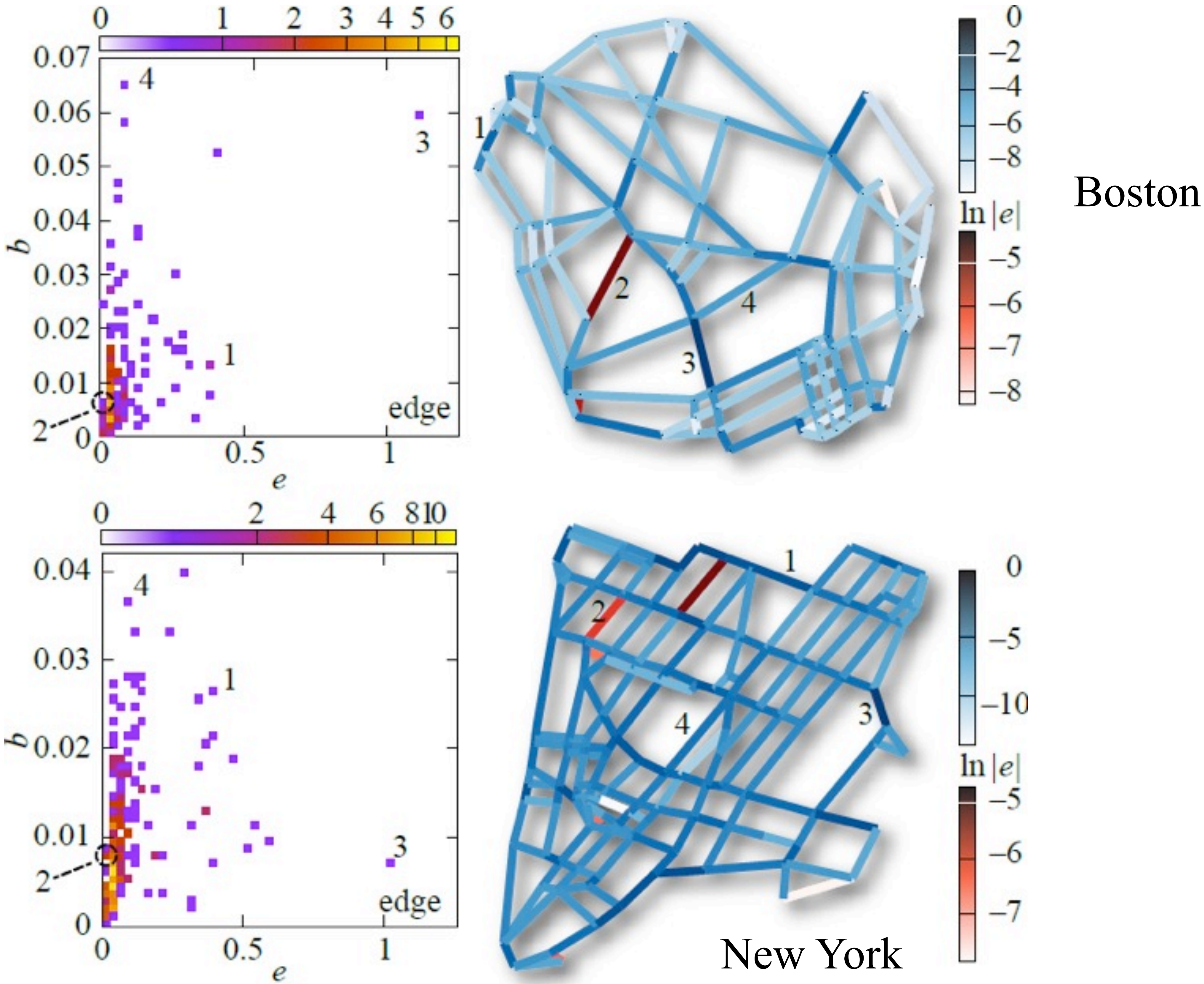
$$e(l) = \mathrm{aGPL}(G \setminus \{l\}) - \mathrm{aGPL}(G)$$

where aGPL = average GSN path length

$e(l) > 0$: the removal of l deteriorates the navigability $\longrightarrow$ "normal" edge l

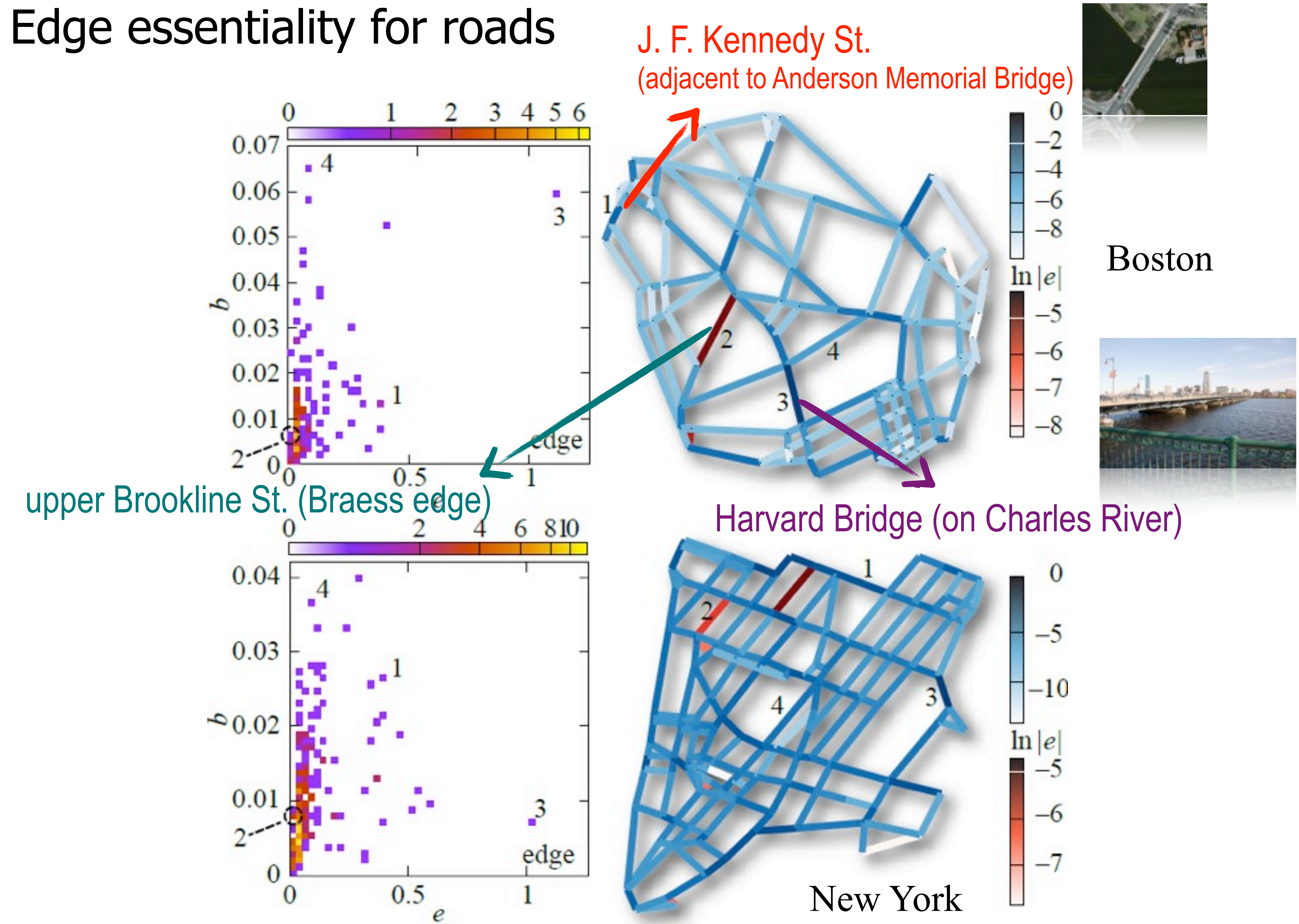
$e(l) < 0$: the removal of l **enhances** the navigability $\longrightarrow$ "Braess" edge l (from Braess's paradox)

Edge essentiality for roads



edge essentiality vs edge betweenness

Edge essentiality for roads



edge essentiality vs edge betweenness

Is it possible to predict e or "Braessiness" from other topological/geometric characteristics?

TABLE II: Coefficients for the multiple linear regression $e = m_1 b + m_2(\text{length}) + m_3 c + m_4(k_i k_j) + m_5 \theta$ for road networks, with some measures defined on edges: b (the edge betweenness), the edge length, the distance c from the midpoint of edges to the centroid of vertices, the product $k_i k_j$ of degrees of vertices attached to edges, and the angle θ between edges and principal flow direction in Eq. (6). The statistical significance codes are $\diamond$: < 0.1, $*$: < 0.05, $**$: < 0.01, and $***$: < 0.001.

road	Boston	New York
m_1	6.938^{***}	8.864^{***}
m_2	$-4.597 \times 10^{-5*}$	$-4.139 \times 10^{-5\diamond}$
m_3	-1.573×10^{-6}	$1.925 \times 10^{-5*}$
m_4	$-8.772 \times 10^{-3**}$	$-5.721 \times 10^{-3*}$
m_5	1.219×10^{-2}	$3.202 \times 10^{-2\diamond}$
multiple R^2	0.2520	0.2059
p-value	2.695×10^{-8}	2.242×10^{-9}

Is it possible to predict e or "Braessiness" from other topological/geometric characteristics?

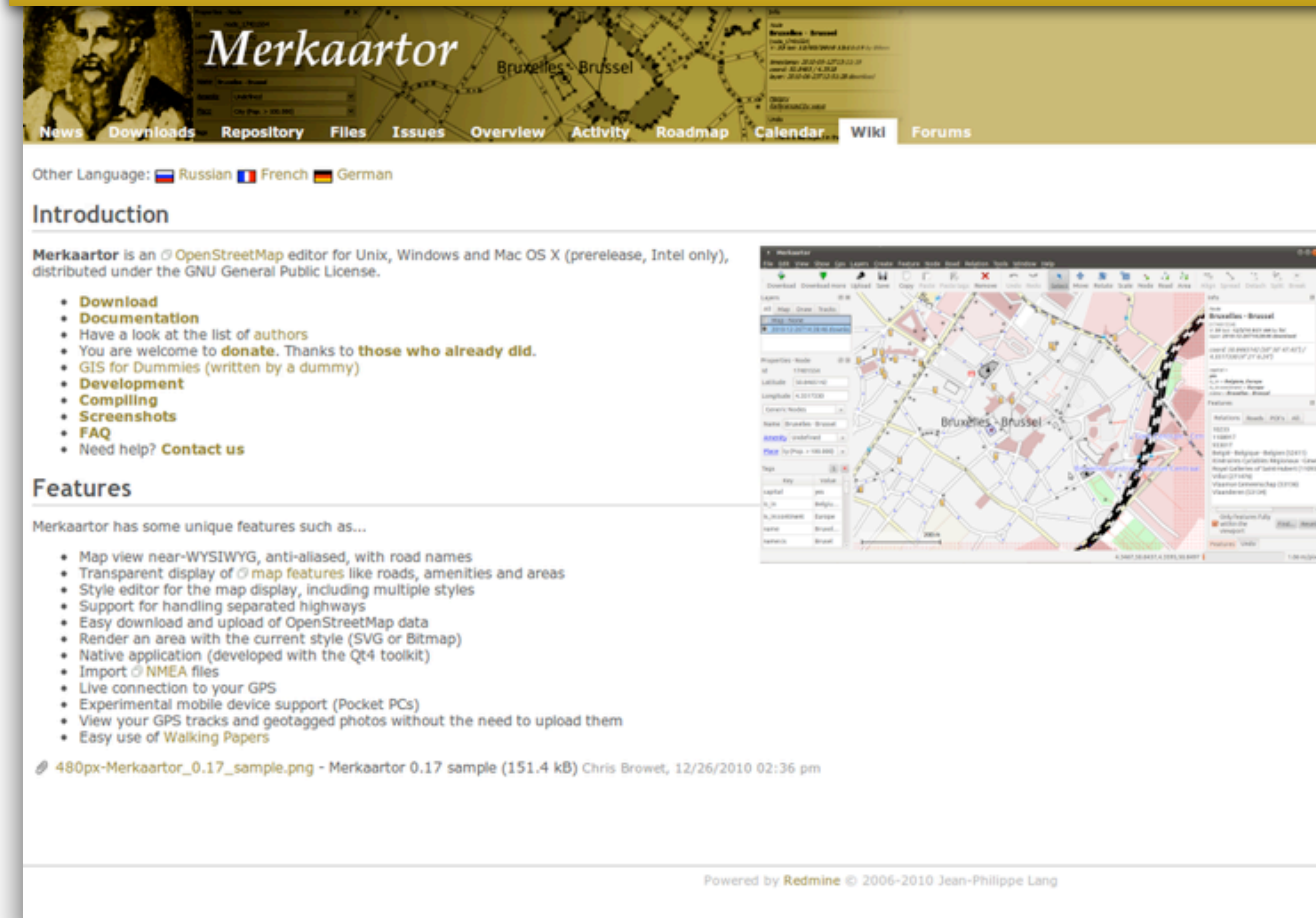
TABLE II: Coefficients for the multiple linear regression $e = m_1 b + m_2(\text{length}) + m_3 c + m_4(k_i k_j) + m_5\theta$ for road networks, with some measures defined on edges: b (the edge betweenness), the edge length, the distance c from the midpoint of edges to the centroid of vertices, the product $k_i k_j$ of degrees of vertices attached to edges, and the angle θ between edges and principal flow direction in Eq. (6). The statistical significance codes are $\diamond$: < 0.1, $*$: < 0.05, $**$: < 0.01, and $***$: < 0.001.

road	Boston	New York
m_1	6.938^{**} [illegible]	[illegible]
m_2	-4.597×1[illegible]	[illegible]
m_3	$-1.573 \times$ [illegible]	[illegible]
m_4	-8.772×1[illegible]	[illegible]
m_5	1.219×10[illegible]	[illegible]
multiple R^2	0.2520	0.2059
p-value	2.695×10^{-8}	2.242×10^{-9}

e heavily depends on very specific geometric arrangements, so it is **not plausible to predict only from other network characteristics!**

Let's get more systematic and reliable data!
(unit area, uniform criterion for selecting roads, etc.)

Merkaartor program (available on Linux, Windows, Mac OS)



Merkaartor

News Downloads Repository Files Issues Overview Activity Roadmap Calendar Wiki Forums

Other Language: Russian French German

Introduction

Merkaartor is an OpenStreetMap editor for Unix, Windows and Mac OS X (prerelease, Intel only), distributed under the GNU General Public License.

- **Download**
- **Documentation**
- Have a look at the list of authors
- You are welcome to **donate**. Thanks to **those who already did**.
- GIS for Dummies (written by a dummy)
- **Development**
- **Compiling**
- **Screenshots**
- **FAQ**
- Need help? **Contact us**

Features

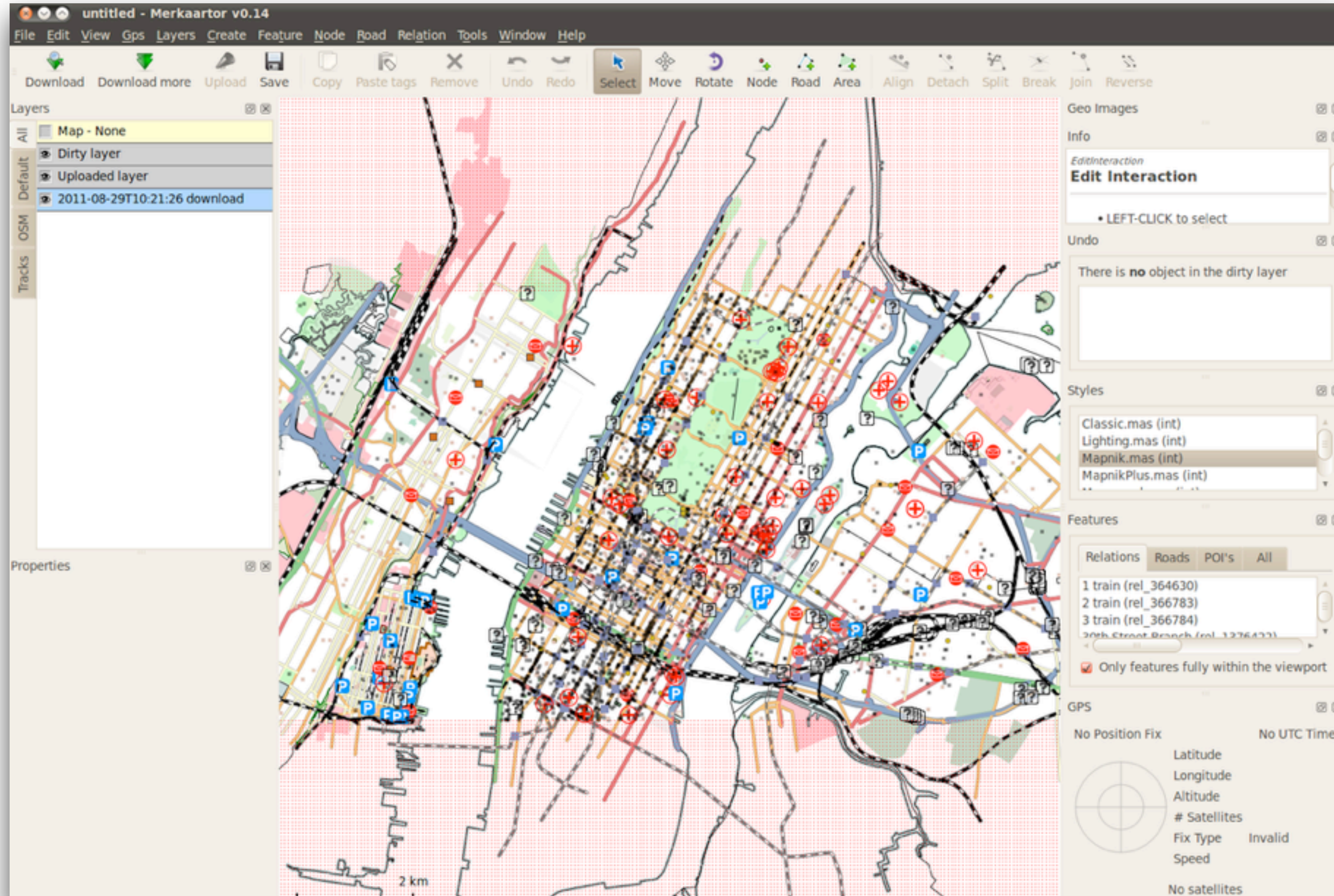
Merkaartor has some unique features such as...

- Map view near-WYSIWYG, anti-aliased, with road names
- Transparent display of map features like roads, amenities and areas
- Style editor for the map display, including multiple styles
- Support for handling separated highways
- Easy download and upload of OpenStreetMap data
- Render an area with the current style (SVG or Bitmap)
- Native application (developed with the Qt4 toolkit)
- Import NMEA files
- Live connection to your GPS
- Experimental mobile device support (Pocket PCs)
- View your GPS tracks and geotagged photos without the need to upload them
- Easy use of Walking Papers

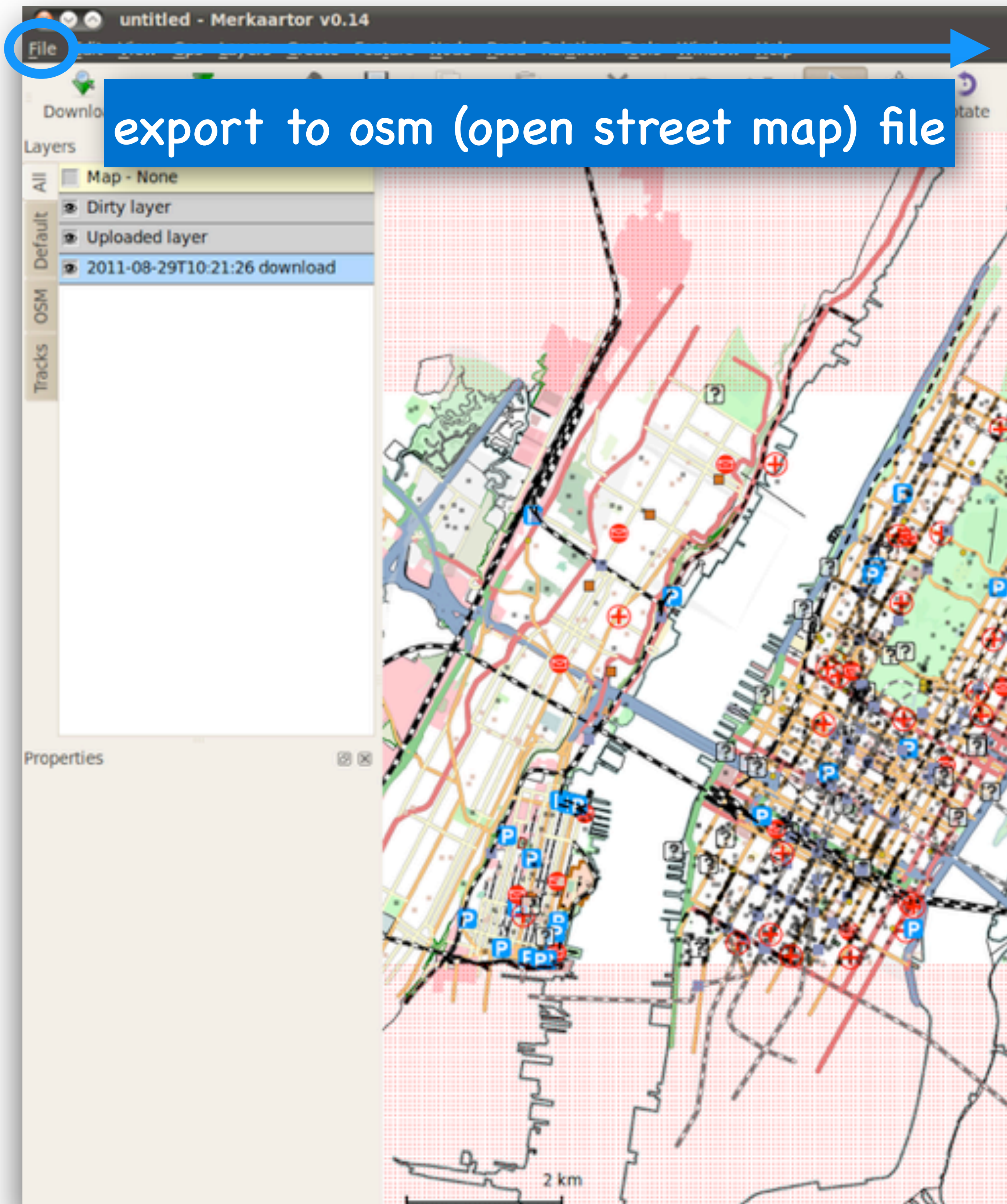
480px-Merkaartor_0.17_sample.png - Merkaartor 0.17 sample (151.4 kB) Chris Browet, 12/26/2010 02:36 pm

Powered by Redmine © 2006-2010 Jean-Philippe Lang

the first step to get the road network data with Merkaartor: New York city case



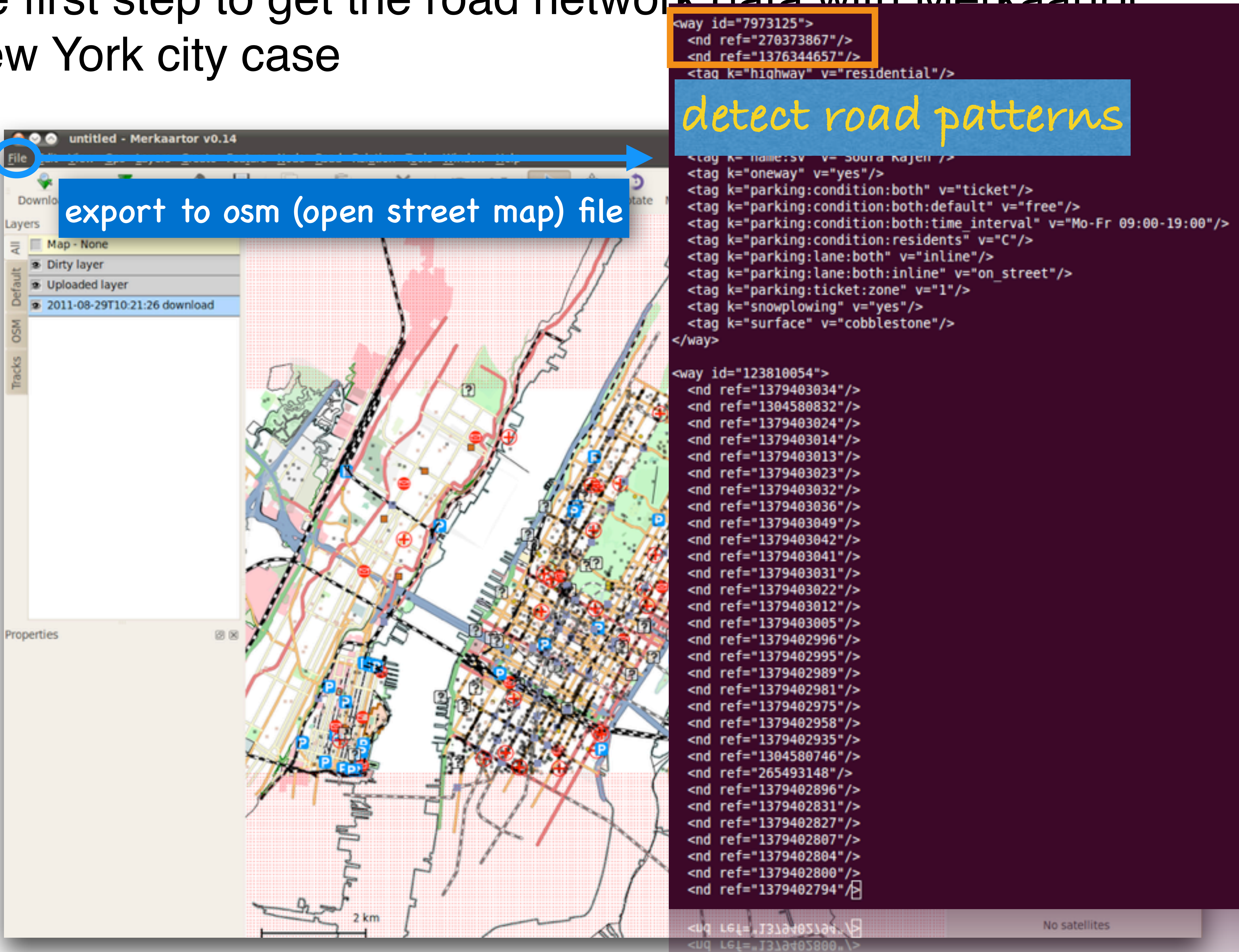
the first step to get the road network data with Merkaartor: New York city case



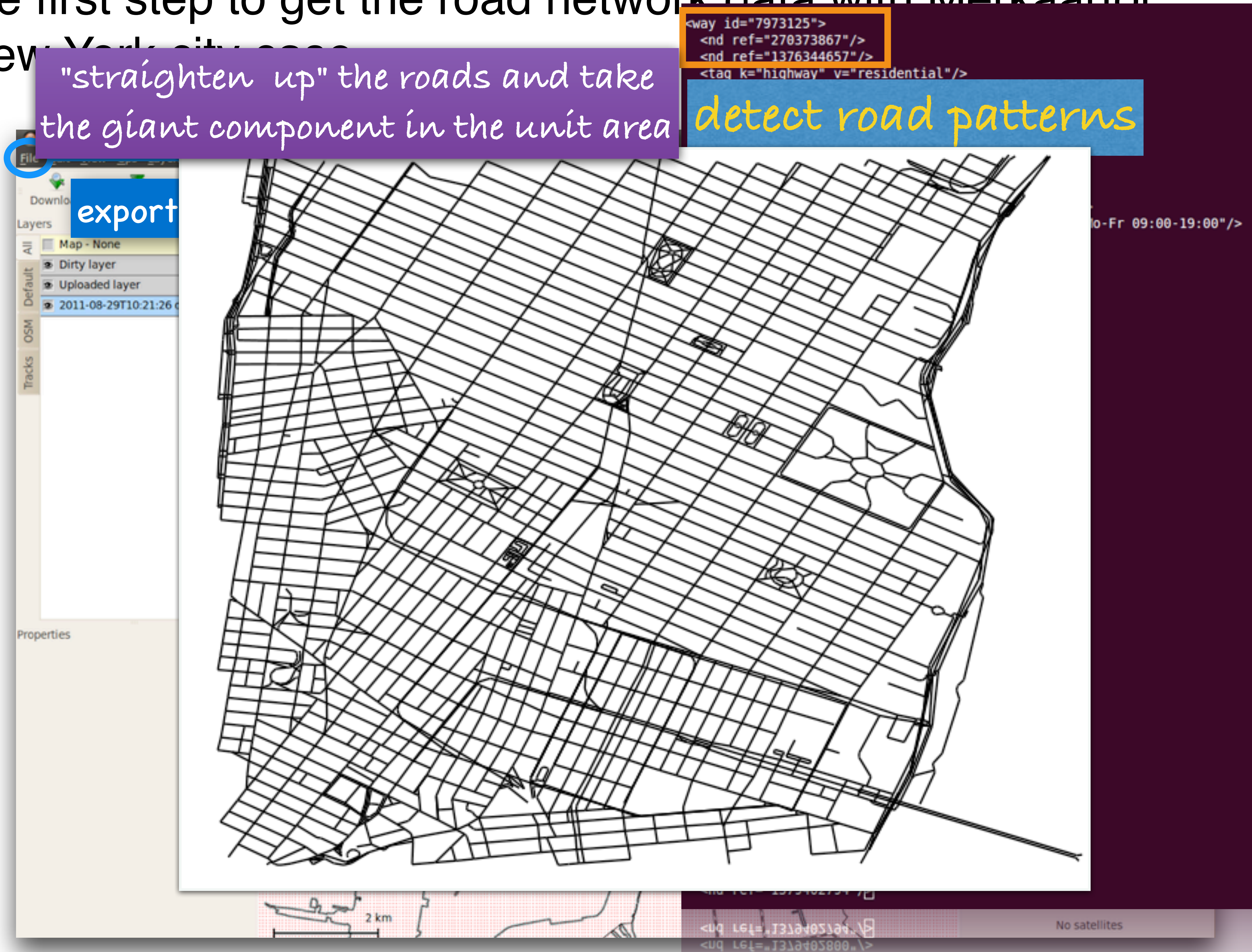

```xml
<way id="7973125">
  <nd ref="270373867"/>
  <nd ref="1376344657"/>
  <tag k="highway" v="residential"/>
  <tag k="lit" v="yes"/>
  <tag k="maxspeed" v="30"/>
  <tag k="name" v="Eteläranta"/>
  <tag k="name:fi" v="Eteläranta"/>
  <tag k="name:sv" v="Södra Kajen"/>
  <tag k="oneway" v="yes"/>
  <tag k="parking:condition:both" v="ticket"/>
  <tag k="parking:condition:both:default" v="free"/>
  <tag k="parking:condition:both:time_interval" v="Mo-Fr 09:00-19:00"/>
  <tag k="parking:condition:residents" v="C"/>
  <tag k="parking:lane:both" v="inline"/>
  <tag k="parking:lane:both:inline" v="on_street"/>
  <tag k="parking:ticket:zone" v="1"/>
  <tag k="snowplowing" v="yes"/>
  <tag k="surface" v="cobblestone"/>
</way>

<way id="123810054">
  <nd ref="1379403034"/>
  <nd ref="1304580832"/>
  <nd ref="1379403024"/>
  <nd ref="1379403014"/>
  <nd ref="1379403013"/>
  <nd ref="1379403023"/>
  <nd ref="1379403032"/>
  <nd ref="1379403036"/>
  <nd ref="1379403049"/>
  <nd ref="1379403042"/>
  <nd ref="1379403041"/>
  <nd ref="1379403031"/>
  <nd ref="1379403022"/>
  <nd ref="1379403012"/>
  <nd ref="1379403005"/>
  <nd ref="1379402996"/>
  <nd ref="1379402995"/>
  <nd ref="1379402989"/>
  <nd ref="1379402981"/>
  <nd ref="1379402975"/>
  <nd ref="1379402958"/>
  <nd ref="1379402935"/>
  <nd ref="1304580746"/>
  <nd ref="265493148"/>
  <nd ref="1379402896"/>
  <nd ref="1379402831"/>
  <nd ref="1379402827"/>
  <nd ref="1379402807"/>
  <nd ref="1379402804"/>
  <nd ref="1379402800"/>
  <nd ref="1379402794"/>
```

the first step to get the road network data with Merkaartor:
New York city case



the first step to get the road network data with Merkaartor:
New York city case



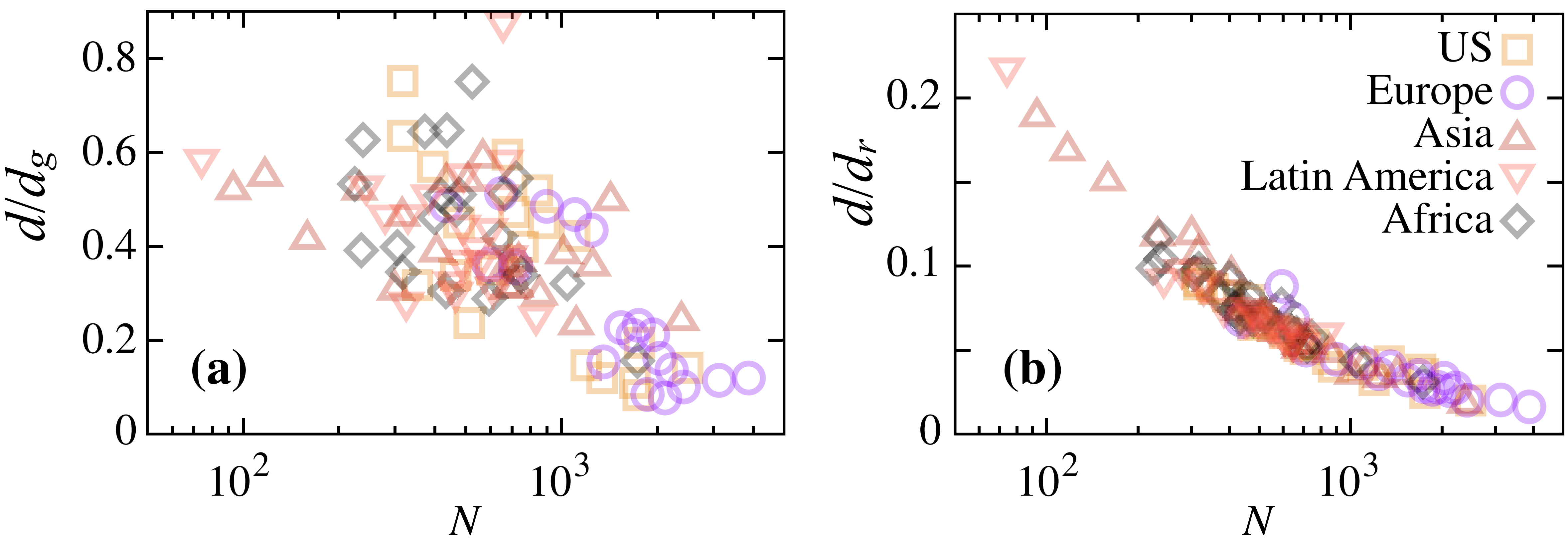
Dataset: 20 largest cities in the US, Europe, Asia, Latin America, and Africa (100 cities in total)

US/Austin	Europe/Barcelona	Asia/Bangkok	LatinAmerica/BeloHorizonte	Africa/Abidjan
US/Charlotte	Europe/Berlin	Asia/Beijing	LatinAmerica/Bogota	Africa/Accra
US/Chicago	Europe/Brussels	Asia/Delhi	LatinAmerica/Brasilia	Africa/AddisAbaba
US/Columbus	Europe/Bucharest	Asia/Dhaka	LatinAmerica/BuenosAires	Africa/Alexandria
US/Dallas	Europe/Budapest	Asia/Guangzhou	LatinAmerica/Caracas	Africa/Algiers
US/Detroit	Europe/Hamburg	Asia/HongKong	LatinAmerica/Fortaleza	Africa/Cairo
US/ElPaso	Europe/London	Asia/Jakarta	LatinAmerica/Guadalajara	Africa/CapeTown
US/FortWorth	Europe/Lyon	Asia/Karachi	LatinAmerica/Guayaquil	Africa/Casablanca
US/Houston	Europe/Madrid	Asia/Kolkata	LatinAmerica/Lima	Africa/Dakar
US/Indianapolis	Europe/Marseille	Asia/Manila	LatinAmerica/Maracaibo	Africa/DarEsSalaam
US/Jacksonville	Europe/Milan	Asia/Mumbai	LatinAmerica/Medellin	Africa/Durban
US/LosAngeles	Europe/Munich	Asia/Nagoya	LatinAmerica/MexicoCity	Africa/Ibadan
US/Memphis	Europe/Naples	Asia/Osaka	LatinAmerica/Monterrey	Africa/Johannesburg
US/NewYork	Europe/Paris	Asia/Seoul	LatinAmerica/PortoAlegre	Africa/Khartoum
US/Philadelphia	Europe/Prague	Asia/Shanghai	LatinAmerica/Recife	Africa/Kinshasa
US/Phoenix	Europe/Rome	Asia/Shenzhen	LatinAmerica/RioDeJaneiro	Africa/Lagos
US/SanAntonio	Europe/Sofia	Asia/Taipei	LatinAmerica/Salvador	Africa/Luanda
US/SanDiego	Europe/Valencia	Asia/Tehran	LatinAmerica/Santiago	Africa/Nairobi
US/SanFrancisco	Europe/Vienna	Asia/Tokyo	LatinAmerica/SantoDomingo	Africa/Pretoria
US/SanJose	Europe/Warsaw	Asia/Wuhan	LatinAmerica/SaoPaulo	Africa/Tunis

d/d_g vs d/d_r profile for 100 large cities (2 km*2 km samples)

- d: real shortest path length
- d_g: GSN path length
- d_r: random DFS path length

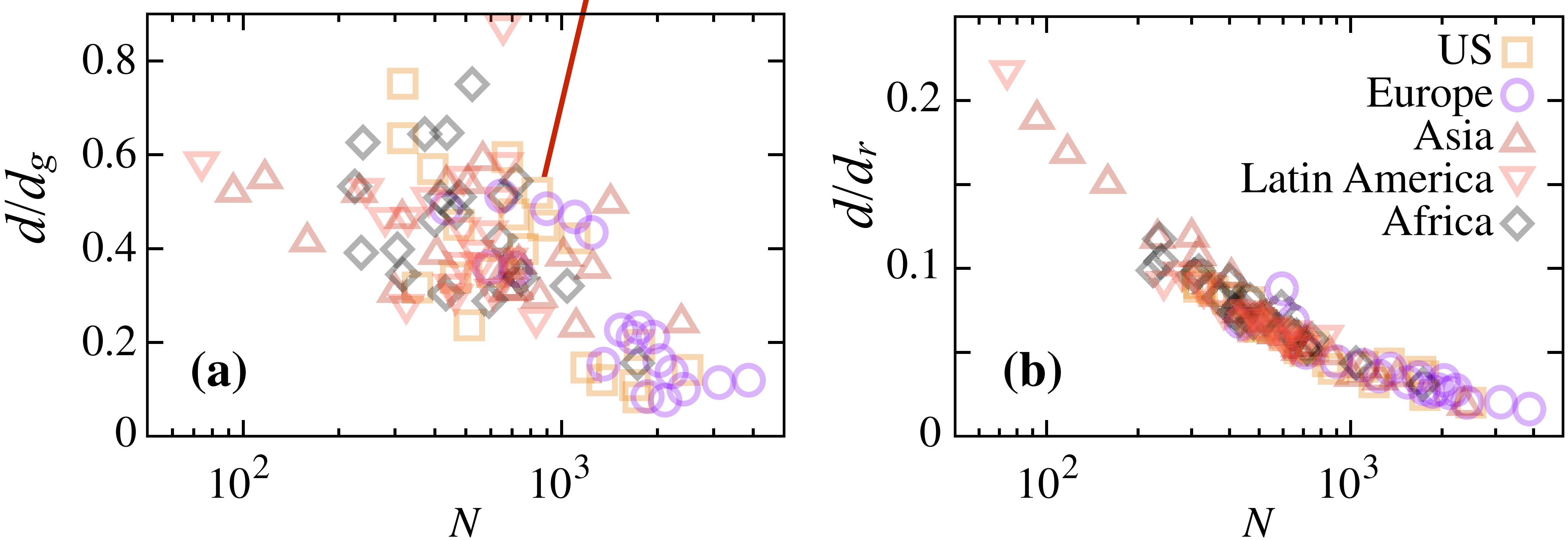
d/d_g compared to d/d_r:
measure of navigability



d/d_g vs d/d_r profile for 100 large cities (2 km*2 km samples)

- d: real shortest path length
- d_g: GSN path length
- d_r: random DFS path length

d/d_g compared to d/d_r:
measure of navigability



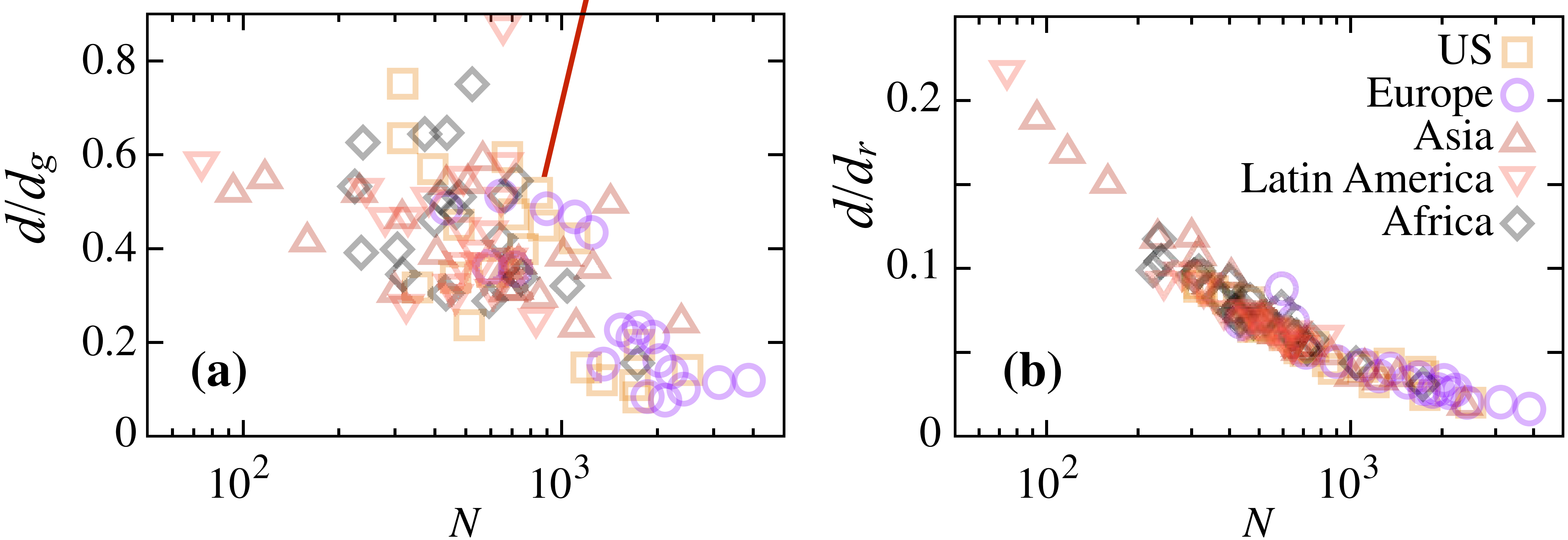
d/d_g vs d/d_r profile for 100 large cities (2 km*2 km samples)

- d: real shortest path length
- d_g: GSN path length
- d_r: random DFS path length

New York

d/d_g compared to d/d_r:

measure of navigability



diverse values for d/d_g vs clear scaling for d/d_r:

d/d_g shows the **real characteristics** of city structures

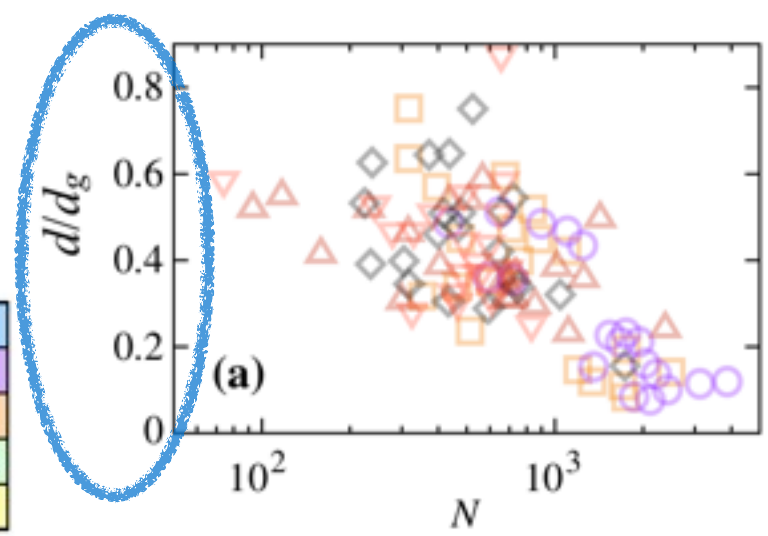
Ranked cities based on d/d_g

SHL and P. Holme, Phys. Rev. Lett. **108**, 128701 (2012).

rank	city	d/d_g
1	Guayaquil	0.87153475
2	Dallas	0.751327251
3	Khartoum	0.750326569
4	Johannesburg	0.646588605
5	Kinshasa	0.643906117
6	Los Angeles	0.635007285
7	Pretoria	0.625790442
8	New York	0.594814987
9	Nagoya	0.590465551
10	Recife	0.581970088
11	Guadalajara	0.580433974
12	Fort Worth	0.569207094
13	Wuhan	0.551891422
14	Lima	0.550886837
15	Cairo	0.544213047
16	Mumbai	0.541609353
17	Tehran	0.539319224
18	Abidjan	0.532253231
19	Maracaibo	0.525445466
20	Shenzhen	0.522871912
21	Jakarta	0.522273008
22	San Francisco	0.518447496
23	Naples	0.51226827
24	Dakar	0.511954476
25	Durban	0.509738578
26	Dar es Salaam	0.509059056
27	Sao Paulo	0.505905555
28	Medellin	0.505887074
29	Kolkata	0.499531424
30	Lyon	0.485799194
31	Budapest	0.484988332
32	Luanda	0.476971656
33	Columbus	0.472754382
34	Barcelona	0.467556084
35	Karachi	0.466714526
36	Caracas	0.463537945
37	Rio de Janeiro	0.462997416
38	Ibadan	0.461537277
39	Charlotte	0.450666431
40	Indianapolis	0.448876406
41	Santiago	0.441916028
42	Mexico City	0.434675377
43	Sofia	0.434447746
44	Philadelphia	0.421522638
45	Casablanca	0.418701767
46	Guangzhou	0.417294581
47	Houston	0.39869371
48	Lagos	0.397973175
49	Santo Domingo	0.393599977
50	Alexandria	0.391348225
51	Delhi	0.390437031
52	Taipei	0.386707203
53	Bogota	0.376506648
54	Manila	0.370664996
55	Buenos Aires	0.367060505
56	Porto Alegre	0.364351628
57	Osaka	0.360996708
58	Tokyo	0.360566814
59	Marseille	0.359403937
60	Bucharest	0.355247549
61	Jacksonville	0.350364823
62	Accra	0.344701237
63	Nairobi	0.33843625
64	El Paso	0.338130399
65	Monterrey	0.320958348
66	Cape Town	0.320270326
67	Memphis	0.317301402
68	Seoul	0.316102892
69	Beijing	0.313137426
70	Belo Horizonte	0.30973628
71	Dhaka	0.309330927
72	Addis Ababa	0.304048152
73	Bangkok	0.299254306
74	Brasilia	0.299090784
75	Algiers	0.287942921
76	Fortaleza	0.277673424
77	Salvador	0.2479511
78	Shanghai	0.244778393
79	San Jose	0.236622937
80	Hong Kong	0.235434556
81	Valencia	0.231731821
82	Vienna	0.226759618
83	Madrid	0.211601578
84	Rome	0.21040621
85	Detroit	0.194525563
86	Warsaw	0.161307412
87	Tunis	0.155626232
88	Milan	0.15375989
89	Chicago	0.146330524
90	Phoenix	0.141044953
91	London	0.136564567
92	San Diego	0.119215411
93	Munich	0.119165581
94	Brussels	0.114166163
95	San Antonio	0.108783428
96	Berlin	0.100183015
97	Hamburg	0.08422761
98	Paris	0.083574313
99	Austin	0.0826651
100	Prague	0.077063185

continent

- US
- Europe
- Asia
- Latin America
- Africa



Ranked cities based on d/d_g

SHL and P. Holme, Phys. Rev. Lett. **108**, 128701 (2012).

rank	city	d/d_g
1	Guayaquil	0.87153475
2	Dallas	0.751327251
3	Khartoum	0.750326569
4	Johannesburg	0.646588605
5	Kinshasa	0.643906117
6	Los Angeles	0.635007285
7	Pretoria	0.625790442
8	New York	0.594814987
9	Nagoya	0.590465551
10	Recife	0.581970088
11	Guadalajara	0.580433974
12	Fort Worth	0.569207094
13	Wuhan	0.551891422
14	Lima	0.550886837
15	Cairo	0.544213047
16	Mumbai	0.541609353
17	Tehran	0.539319224
18	Abidjan	0.532253231
19	Maracaibo	0.525445466
20	Shenzhen	0.522871912
21	Jakarta	0.522273008
22	San Francisco	0.518447496
23	Naples	0.51226827
24	Dakar	0.511954476
25	Durban	0.509738578
26	Dar es Salaam	0.509059056
27	Sao Paulo	0.505905555
28	Medellin	0.505887074
29	Kolkata	0.499531424
30	Lyon	0.485799194
31	Budapest	0.484988332
32	Luanda	0.476971656
33	Columbus	0.472754382
34	Barcelona	0.467556084
35	Karachi	0.466714526

Physics

spotlighting exceptional research

Home | About | Browse | APS Journals

Synopsis: Greed is Good

iStockphoto/porcorex

Exploring Maps with Greedy Navigators
Sang Hoon Lee and Petter Holme
Phys. Rev. Lett. **108**, 128701 (2012)
Published March 22, 2012

Many a tourist has, perhaps happily, gotten lost in the twists and turns along the way to Venice's Piazza San Marco. How navigable a city is—or could be with an extra footbridge or better-placed signs—is something network models try to quantify. Now, writing in *Physical Review Letters*, two scientists show how one such model could better account for the way humans actually go about reaching a destination.

Sang Hoon Lee and Petter Holme at Umeå University in Sweden focus on a type of "greedy" navigation model, where at each point on a map, a navigator heads in the direction most in line with her destination (say a tall building in the distance) and only backtracks if she can't move to a point that hasn't already been visited. The model thus assumes a navigator has more information than one making random decisions, but doesn't have at hand any "smart" technology telling her the overall shortest route.

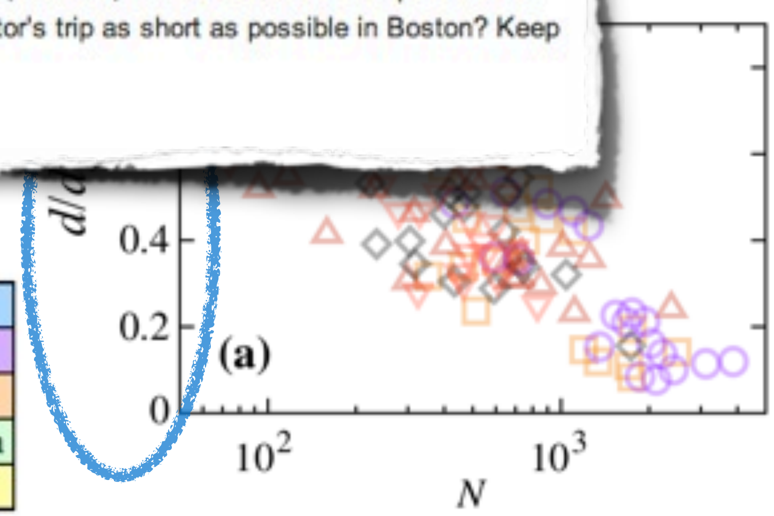
Using maps of New York, Boston, and the Swiss Rail System, as well as the maze at Leeds Castle in England, the authors compare the distance traveled by a greedy navigator with that taken by a random navigator and the actual shortest path. Not surprisingly, greedy navigators get to where they are going in a much shorter distance than random travelers, though this advantage almost vanishes in the disorienting twists and turns in a maze.

Such models could be used to figure out the impact of blocking off certain bridges, tunnels, or roads on drivers or pedestrians trying to navigate a city. What do Lee and Holme advise to keep a greedy navigator's trip as short as possible in Boston? Keep the Harvard Bridge open. – *Jessica Thomas*

Previous synopsis | Next synopsis

66	Cape Town	0.320270326
67	Memphis	0.317301402
68	Seoul	0.316102892
69	Beijing	0.313137426
70	Belo Horizonte	0.30973628
71	Dhaka	0.309330927
72	Addis Ababa	0.304048152

continent
US
Europe
Asia
Latin America
Africa



If we're designers/architects of systems … (Part I)

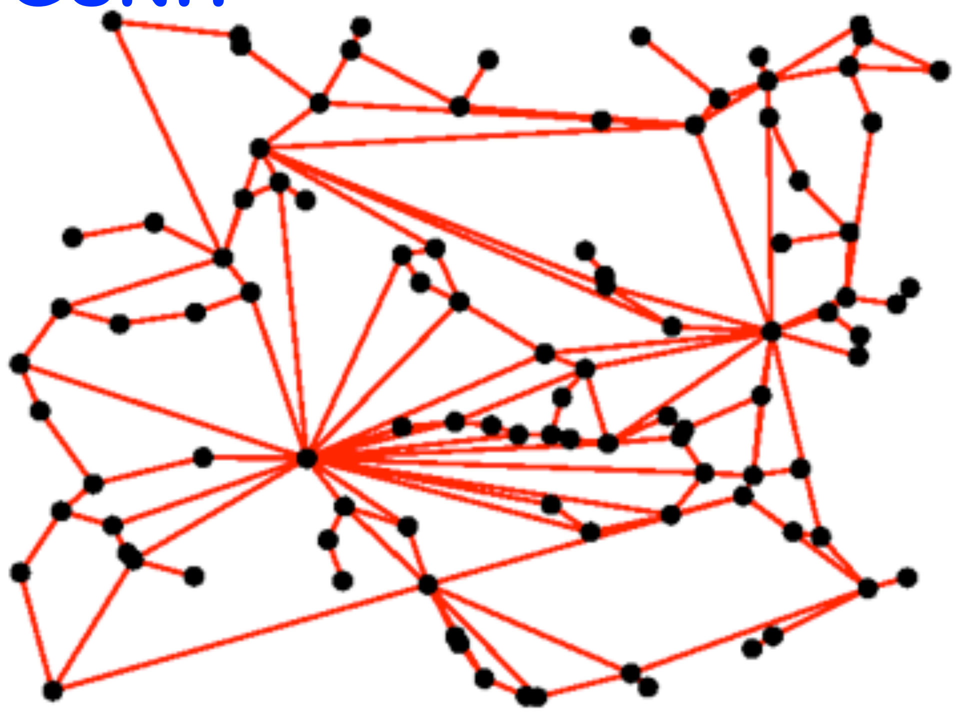
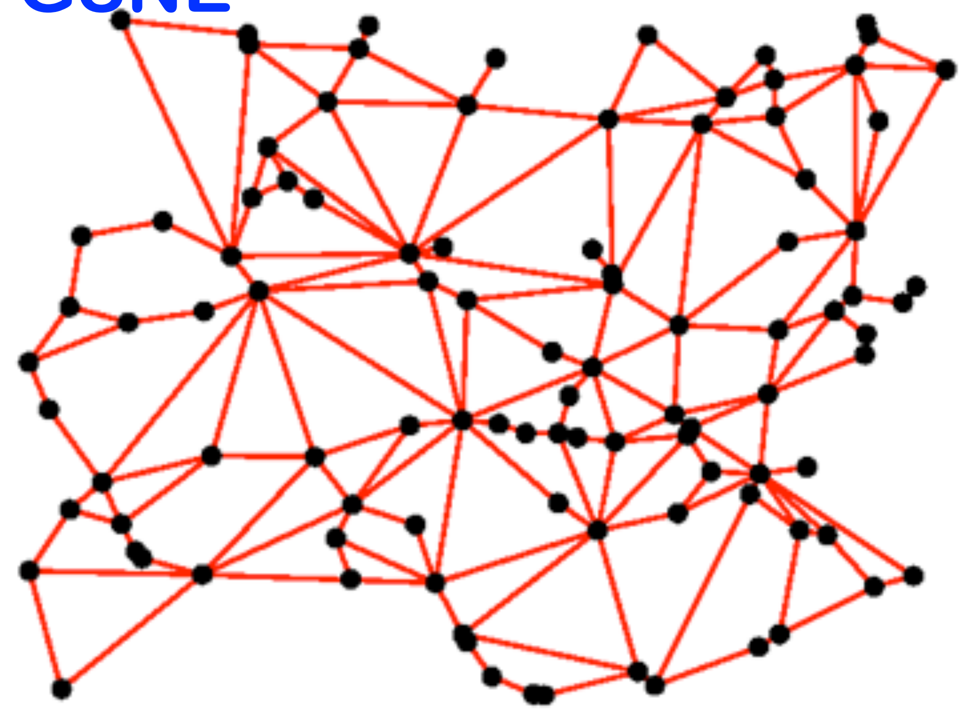
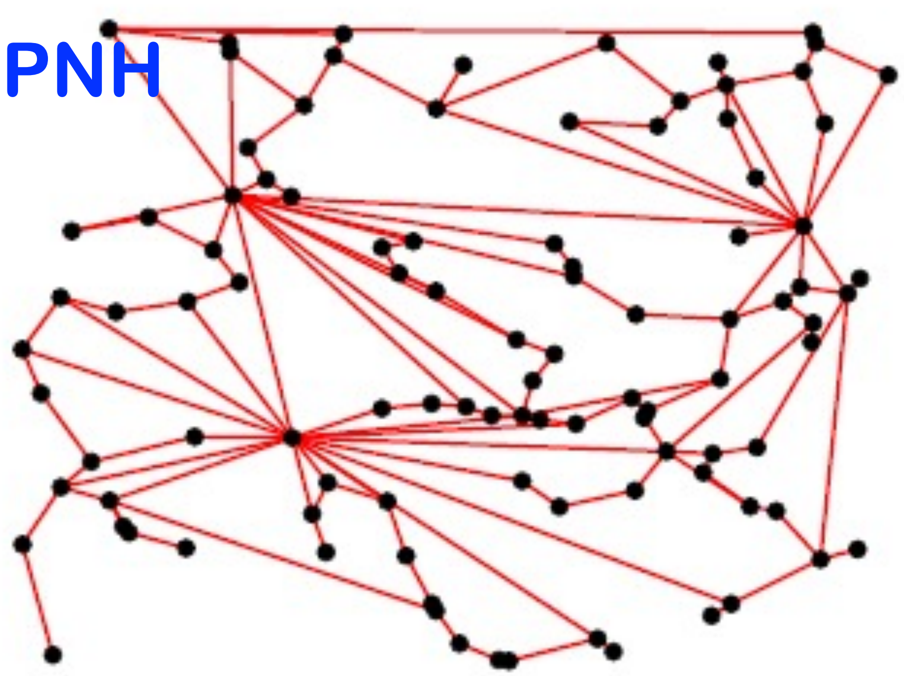
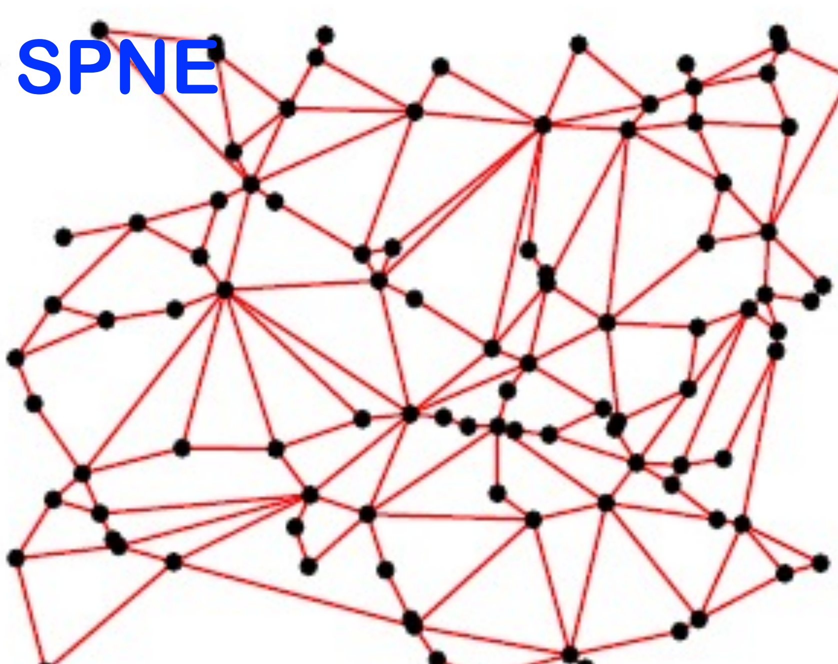
- How to **optimize the network edges** for "greedy and smart" navigator with GSN strategy?

Greedy shortcut construction model

- initial configuration: minimum spanning tree (MST) from the given **vertices on 2D space**, **minimizing the total length** of the road
- adding a shortcut which **does not cross the existing edges**, **maximizing the GSN performance** at each time step
  - repeating this as long as the sum of all the road lengths does not exceed a certain threshold $l_{\max}$ (limited resource)

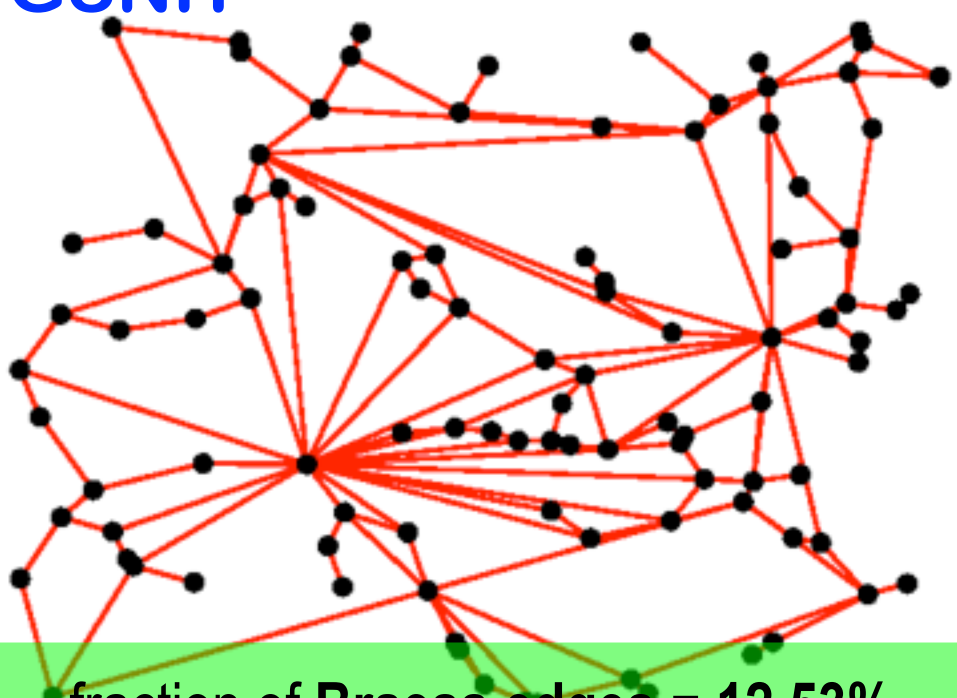
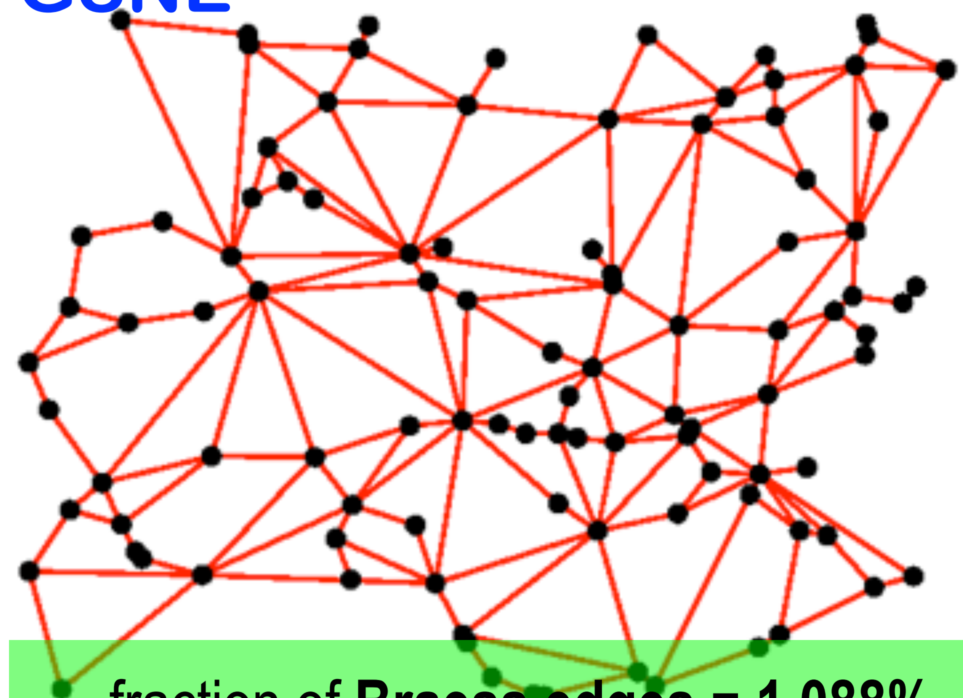
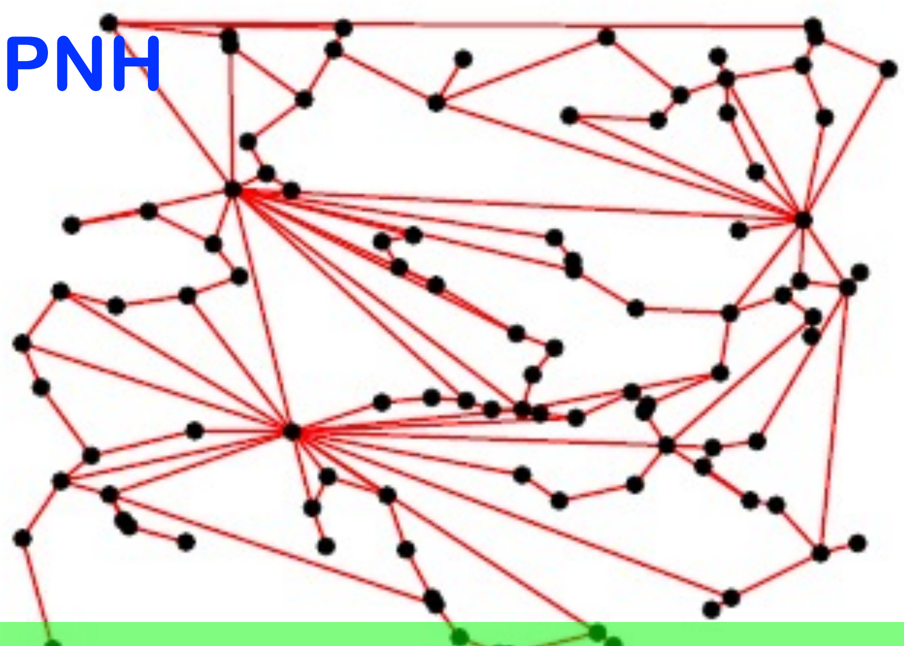
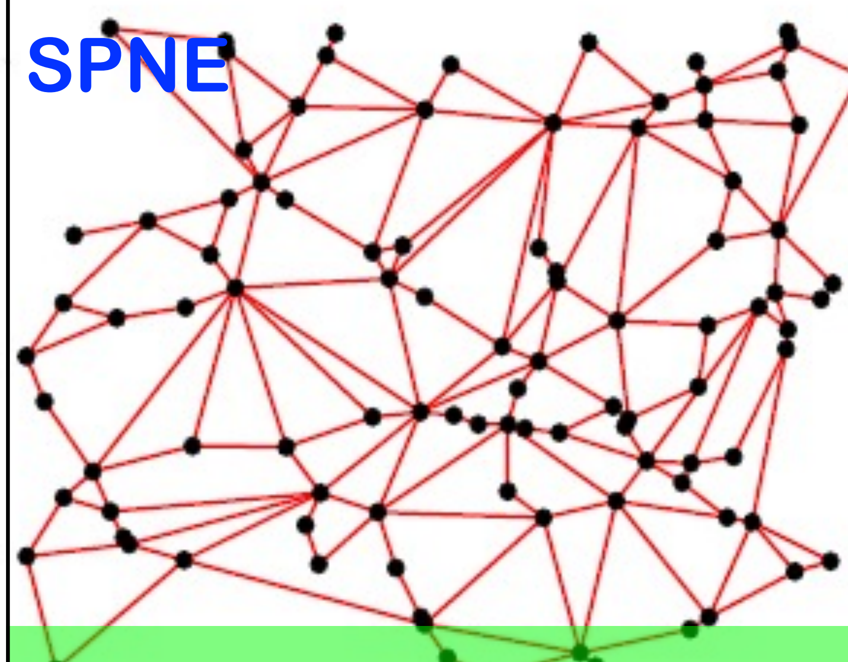
same initial locations for vertices

Structures emerged from randomly distributed vertices on unit squares

$N = 10^2$ $l_{max} = 20$	hopping-distance-based	Euclidean-distance-based
GSN (greedy navigator)	**GSNH** 	**GSNE** 
SP (real shortest path)	**SPNH** 	**SPNE** 

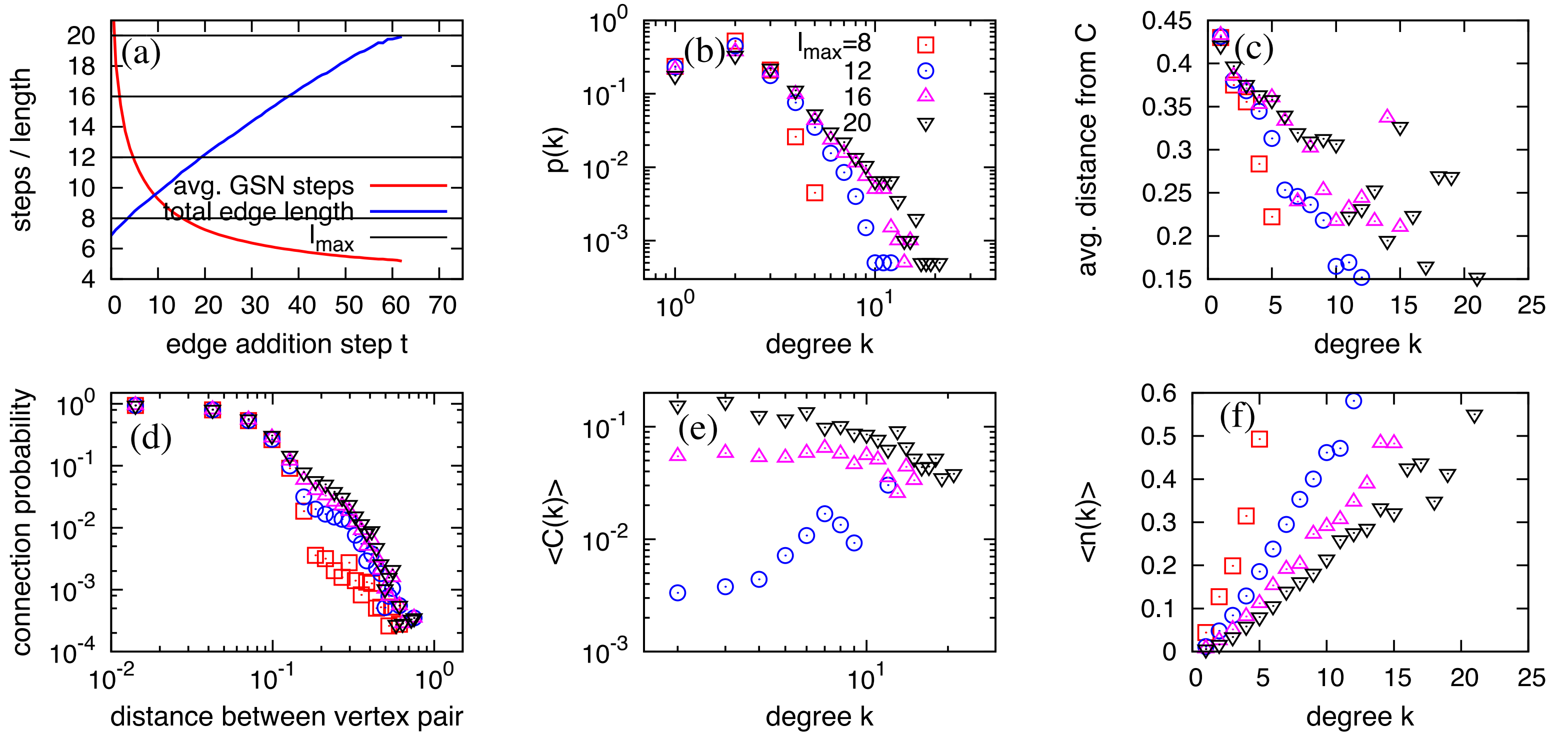
same initial locations for vertices

Structures emerged from randomly distributed vertices on unit squares

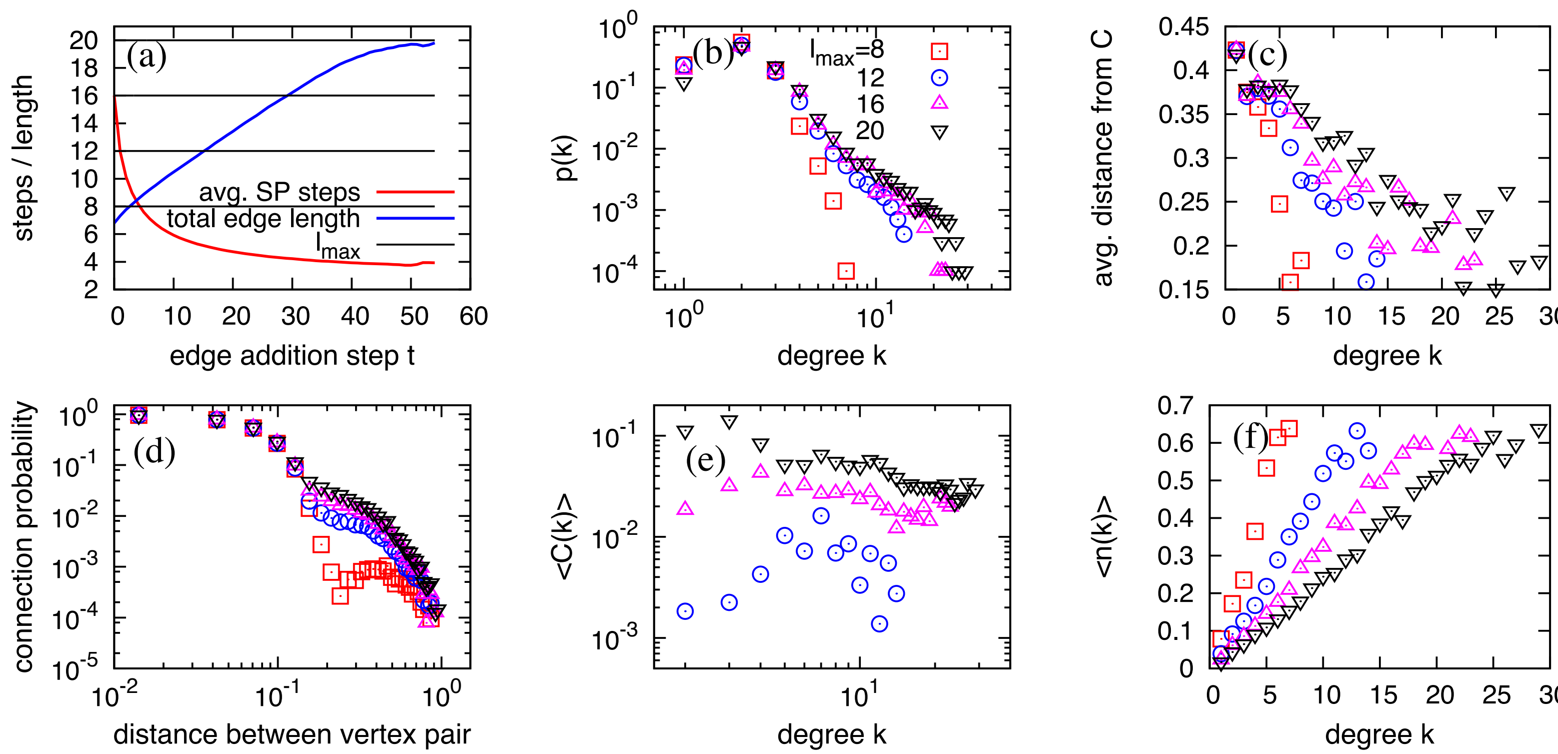
$N = 10^2$ $l_{max} = 20$	hopping-distance-based	Euclidean-distance-based
GSN (greedy navigator)	**GSNH**  fraction of **Braess edges = 12.53%**	**GSNE**  fraction of **Braess edges = 1.088%**
SP (real shortest path)	**SPNH**  fraction of **Braess edges = 20.41%**	**SPNE**  fraction of **Braess edges = 4.934%**

Structures emerged from randomly distributed vertices on unit squares

GSNH

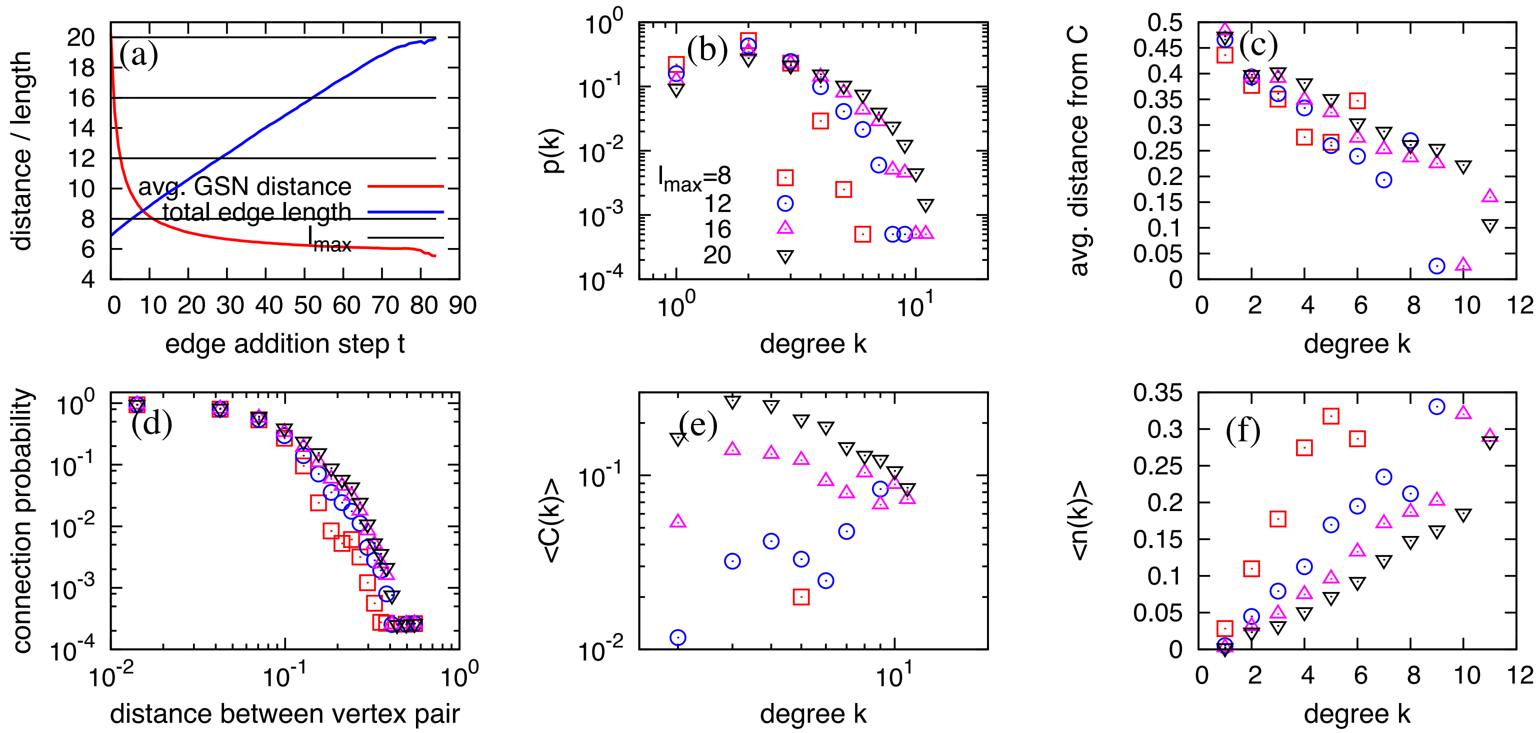


SPNH

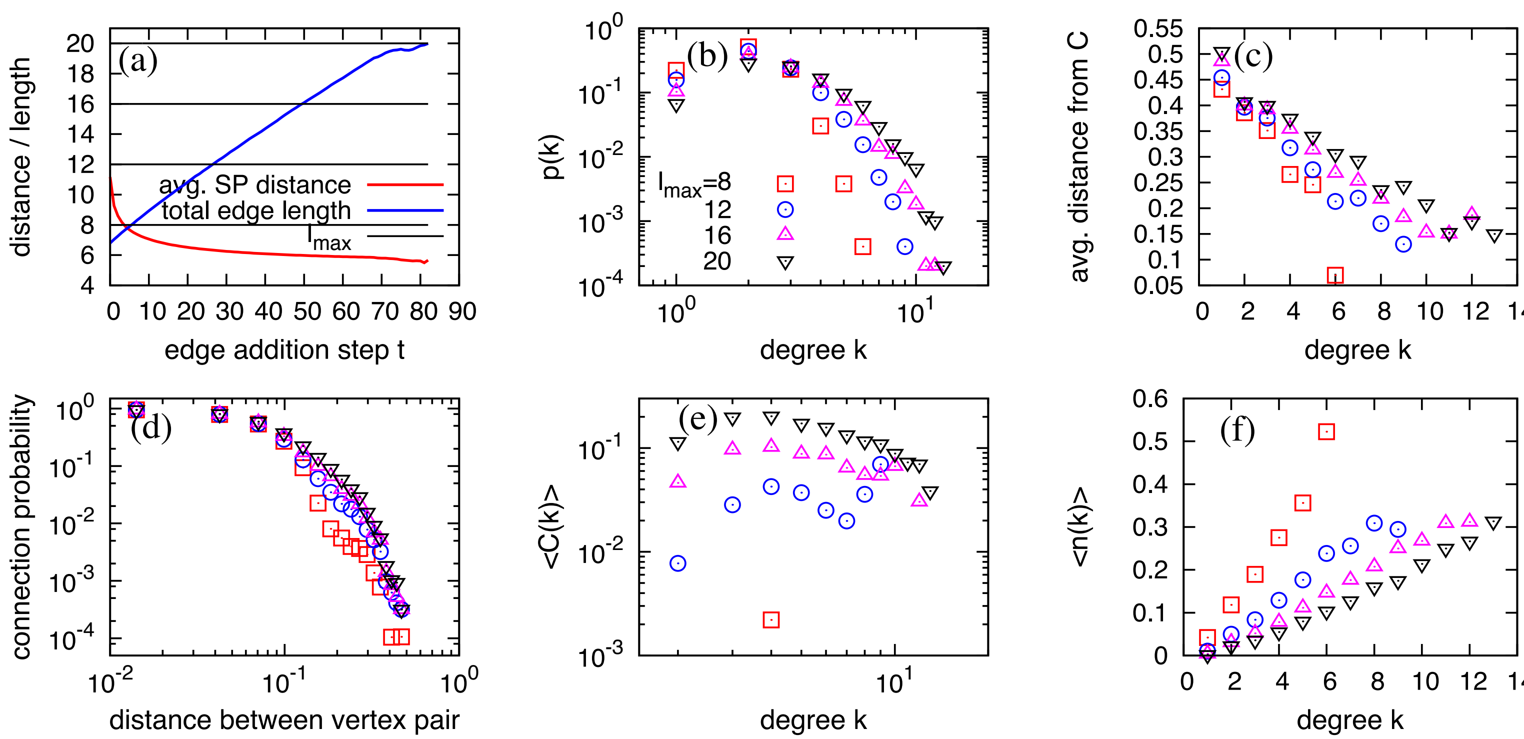


Structures emerged from randomly distributed vertices on unit squares

GSNE



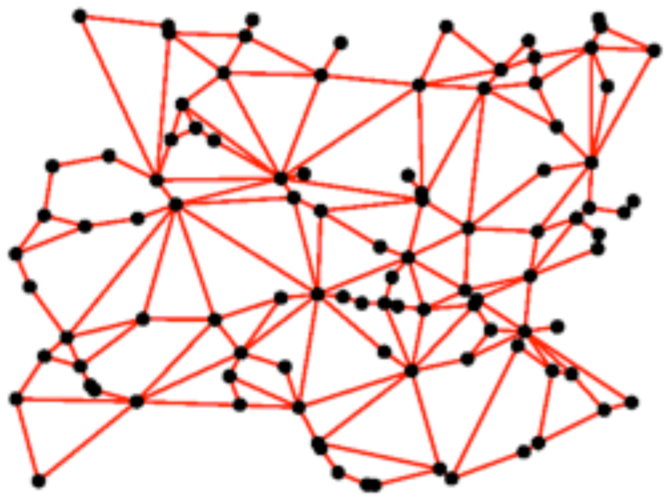
SPNE



Structures emerged from randomly distributed vertices on unit squares

TABLE III: The clustering coefficient based on the number of triangles ($C_\triangle$), compared to the random counterpart ($C_r = 2M/N^2$, where N and M are the numbers of vertices and edges, respectively).

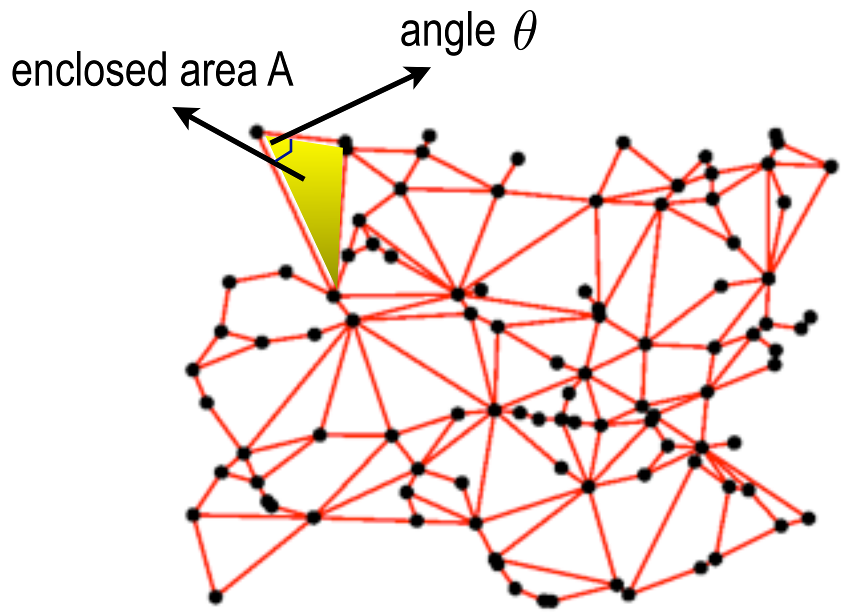
method	$C_\triangle$	C_r	$C_\triangle/C_r$
GSNH	1.04×10^{-1}	3.20×10^{-2}	3.26
GSNE	1.89×10^{-1}	3.66×10^{-2}	5.15
SPNH	6.29×10^{-2}	2.98×10^{-2}	2.11
SPNE	1.56×10^{-1}	3.44×10^{-2}	4.53

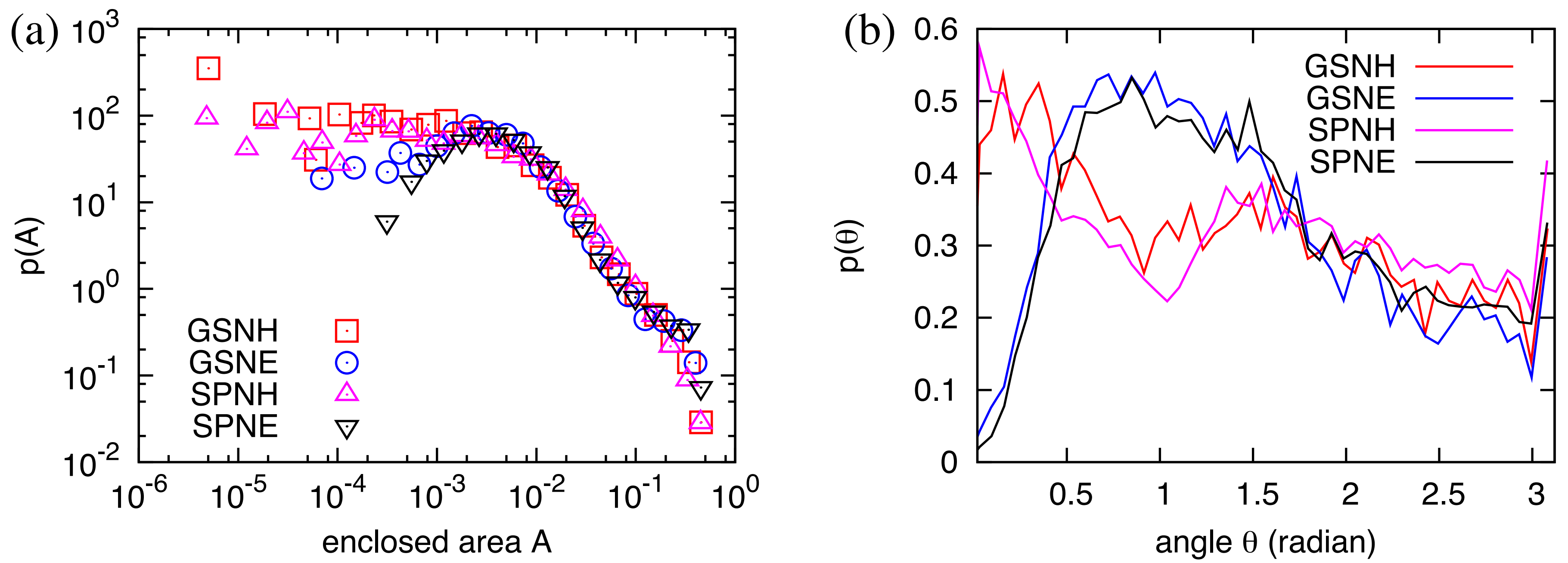


Structures emerged from randomly distributed vertices on unit squares

TABLE III: The clustering coefficient based on the number of triangles ($C_\triangle$), compared to the random counterpart ($C_r = 2M/N^2$, where N and M are the numbers of vertices and edges, respectively).

method	$C_\triangle$	C_r	$C_\triangle/C_r$
GSNH	1.04×10^{-1}	3.20×10^{-2}	3.26
GSNE	1.89×10^{-1}	3.66×10^{-2}	5.15
SPNH	6.29×10^{-2}	2.98×10^{-2}	2.11
SPNE	1.56×10^{-1}	3.44×10^{-2}	4.53



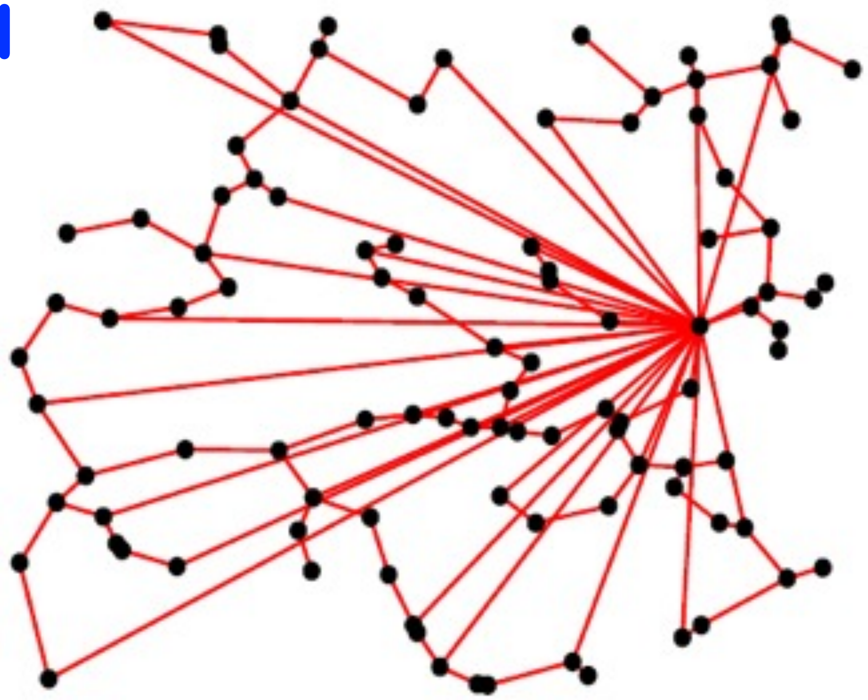


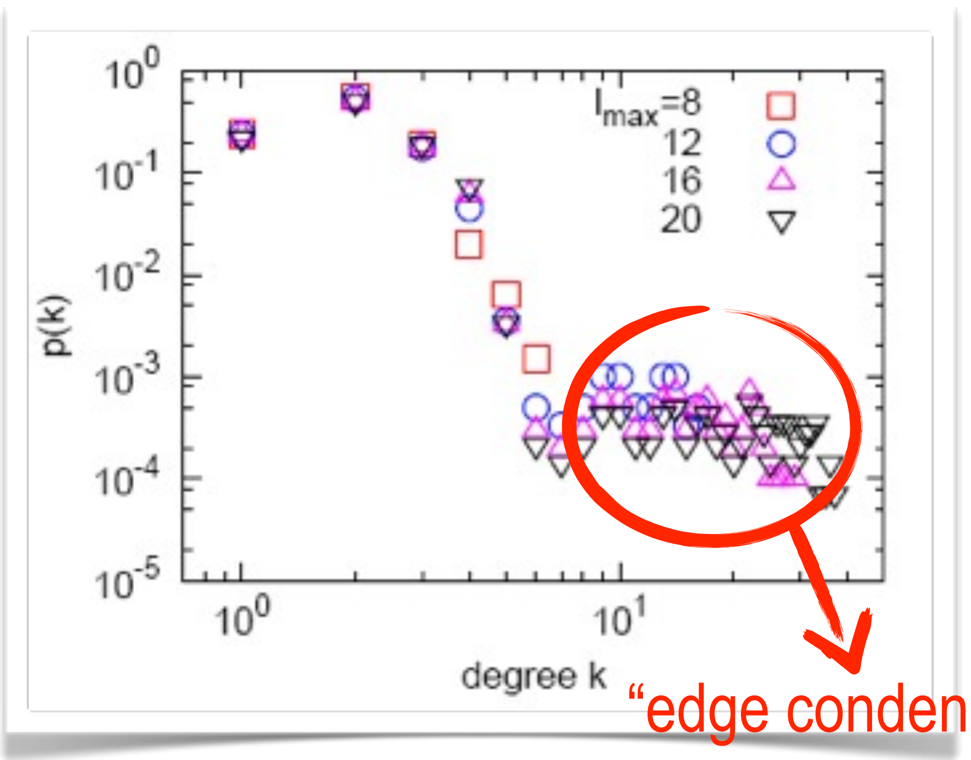
same initial locations for vertices

Role of no-crossing rule

- If "crossing" is allowed,

SPNH





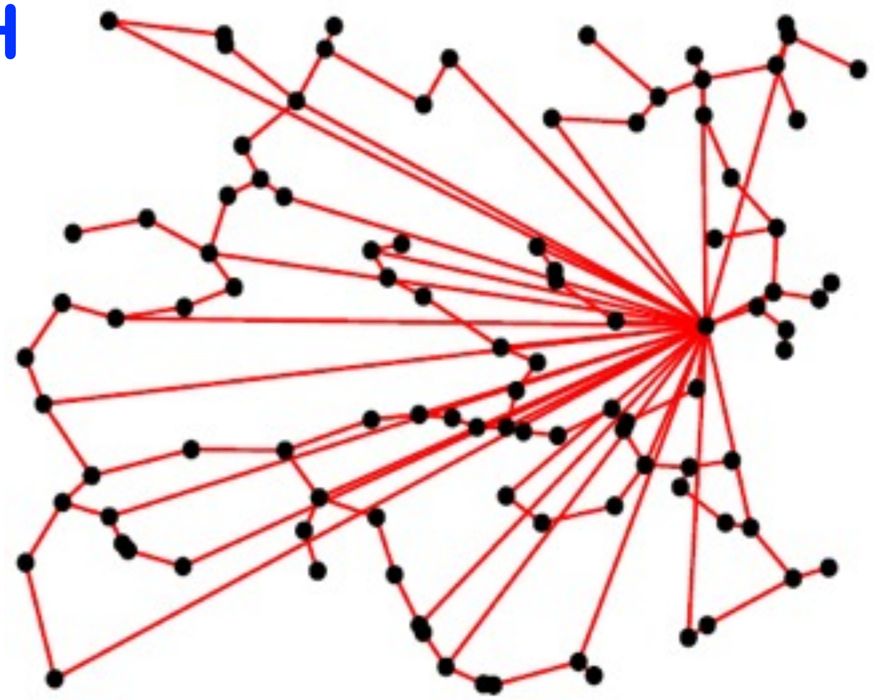
"edge condensation" for
shortest path routing!

same initial locations for vertices

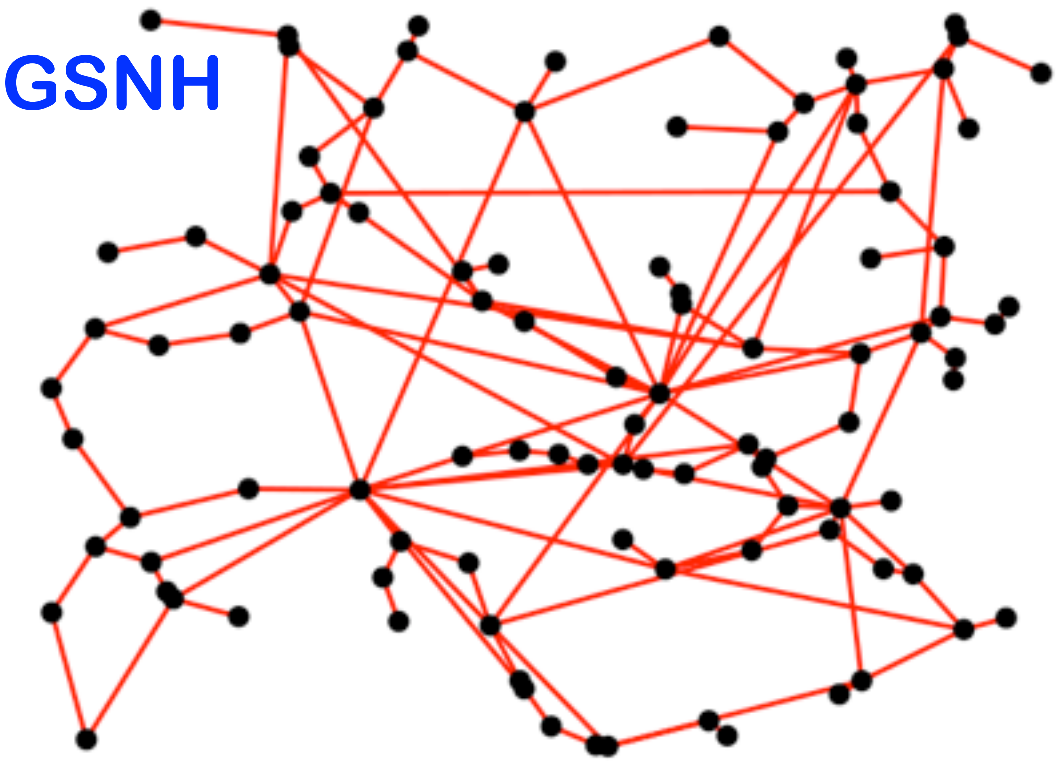
Role of no-crossing rule

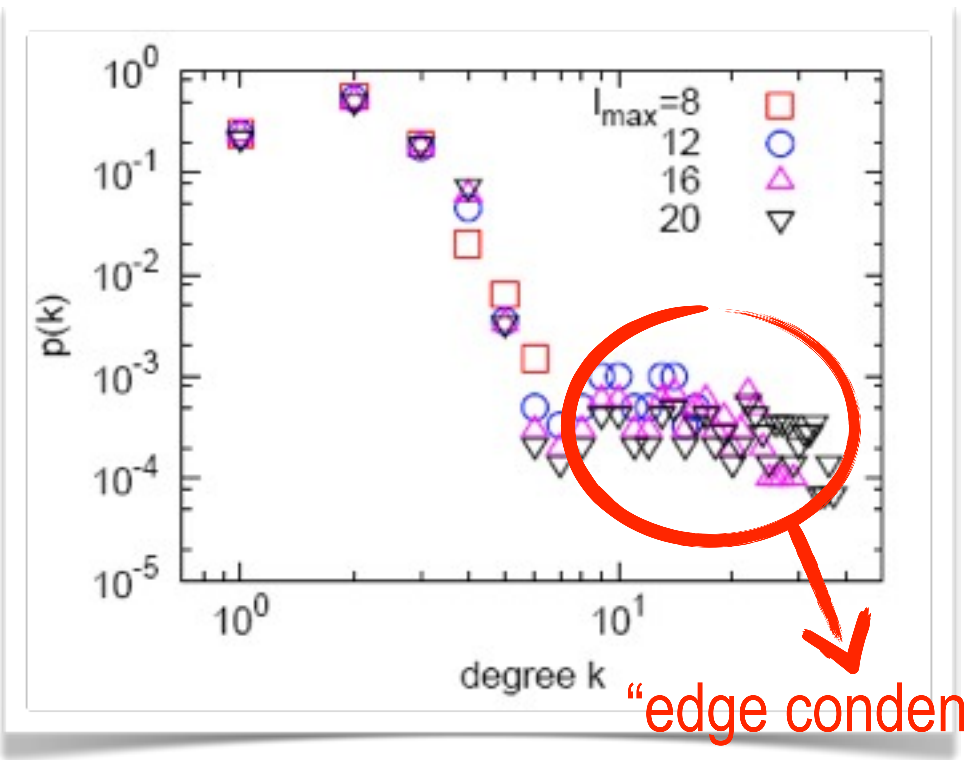
- If “crossing” is allowed,

SPNH



GSNH





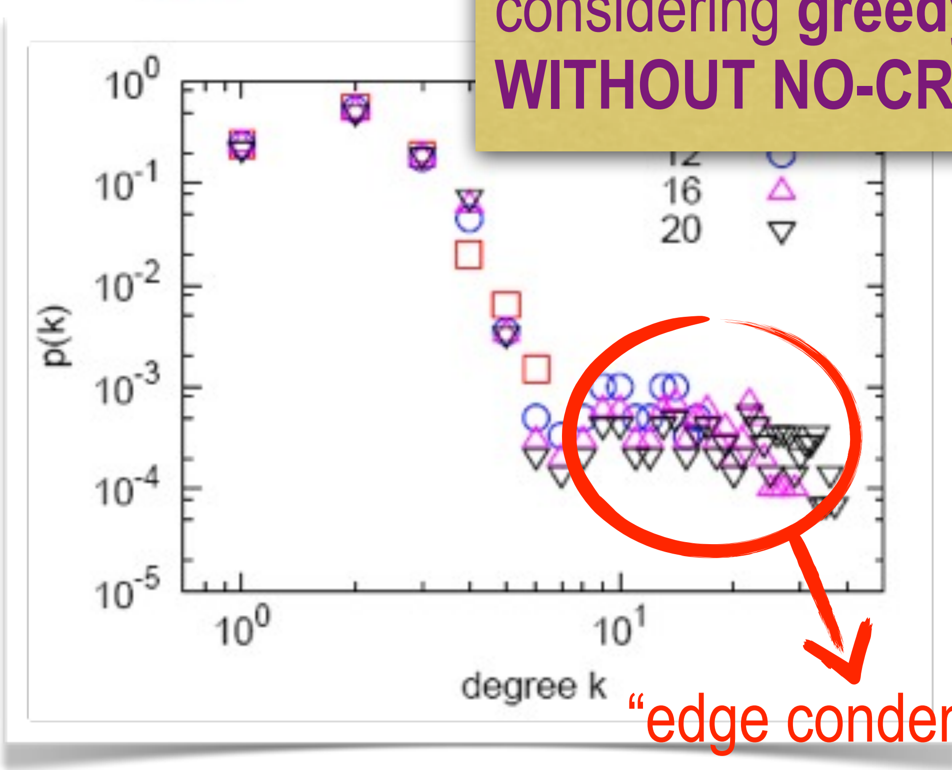
“edge condensation” for
shortest path routing!

same initial locations for vertices

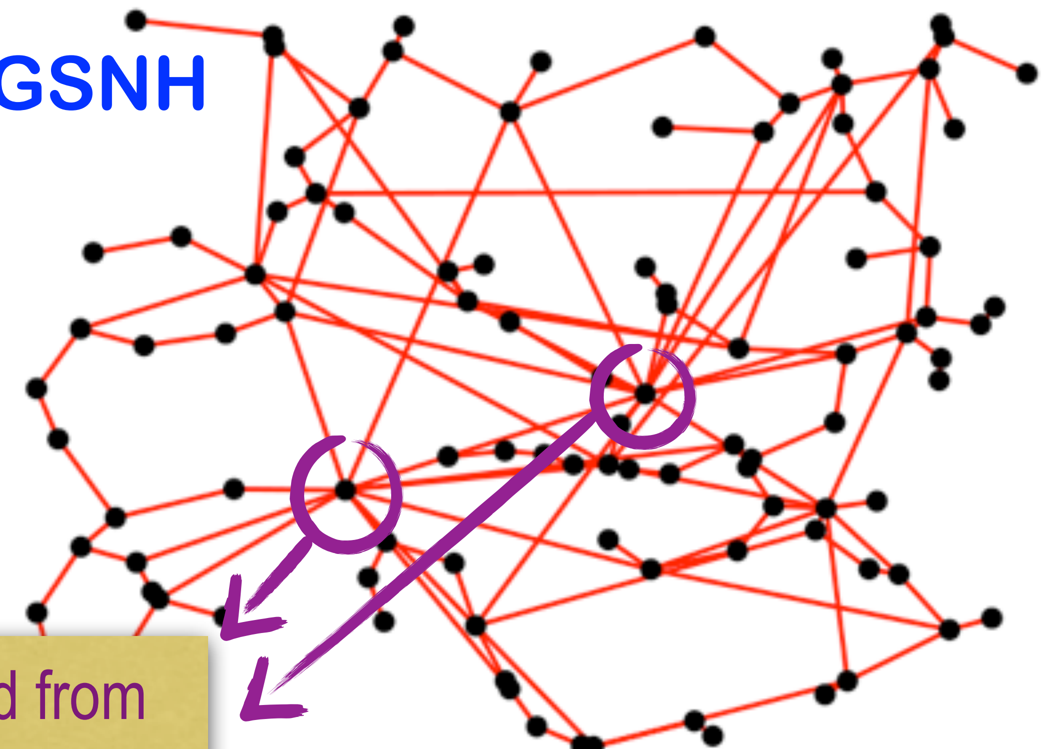
Role of no-crossing rule

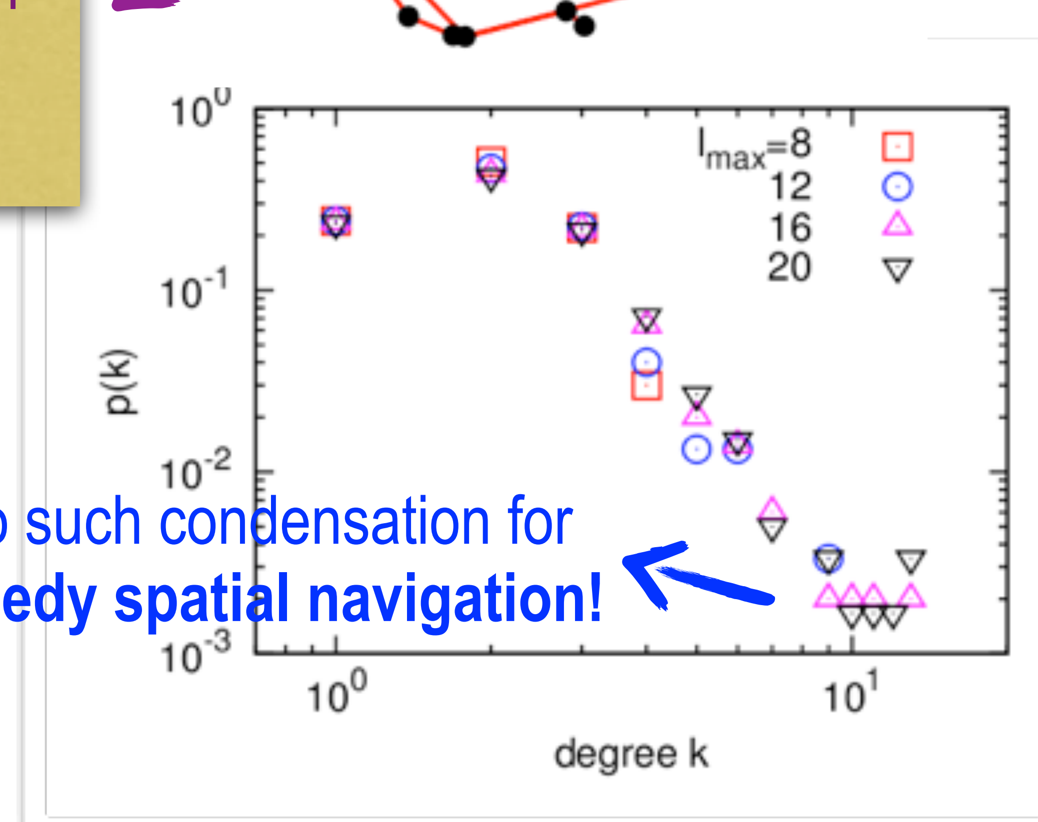
- If “crossing” is allowed,

SPNH



GSNH





“local hubs” naturally emerged from considering **greedy navigators**, **WITHOUT NO-CROSSING RULE**

“edge condensation” for **shortest path routing!**

no such condensation for **greedy spatial navigation!**

SHL and P. Holme, arXiv:1205.0537 (Eur. Phys. J.-Spec. Top., in press)

If we're designers/architects of systems … (Part II)

- How to **optimize the network layout** for "greedy and smart" navigator with GSN strategy?

Layout optimization based on simulated annealing

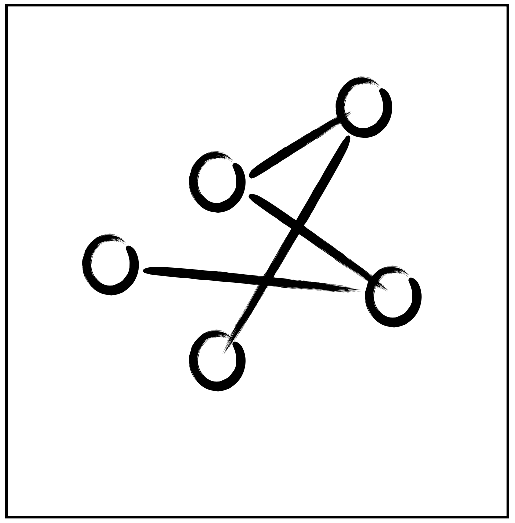
- initial configuration: randomly distributed **vertices (and edges attached to them) on 2D space** inside the unit square, for a **given** network topology

- simulated annealing
  - trial movement: choose a random vertex with the coordinates (x_0, y_0)

    $(x_0, y_0) \to (x_0 + \Delta x, y_0 + \Delta y)$ where Δx and Δy are uniformly randomly drawn from the interval $[-l, l]$

  - calculate the average (hopping-distance-based) GSN pathway d_g, which is the **object function** to be minimized
  - accept the movement if d_g is decreased, or with probability p otherwise
    - with $p = p_{\text{high}}$ (heating) & $p = p_{\text{low}}$ (quenching) repeatedly
  - record the **layout with the minimum d_g value**

$d_g\,(t = 0)$



If we're designers/architects of systems … (Part II)

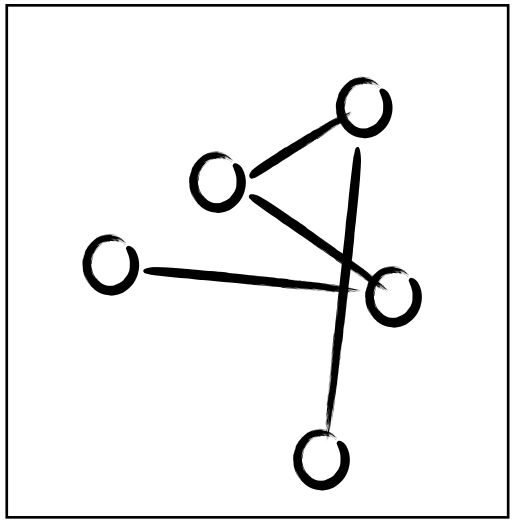
- How to **optimize the network layout** for "greedy and smart" navigator with GSN strategy?

Layout optimization based on simulated annealing

- initial configuration: randomly distributed **vertices (and edges attached to them) on 2D space** inside the unit square, for a **given** network topology
- simulated annealing
  - trial movement: choose a random vertex with the coordinates (x_0, y_0)

    $(x_0, y_0) \to (x_0 + \Delta x, y_0 + \Delta y)$ where Δx and Δy are uniformly randomly drawn from the interval $[-l, l]$
  - calculate the average (hopping-distance-based) GSN pathway d_g, which is the **object function** to be minimized
  - accept the movement if d_g is decreased, or with probability p otherwise
    - with $p = p_{\text{high}}$ (heating) & $p = p_{\text{low}}$ (quenching) repeatedly
  - record the **layout with the minimum d_g value**

$d_g (t = 1)$



If we're designers/architects of systems ... (Part II)

- How to **optimize the network layout** for "greedy and smart" navigator with GSN strategy?

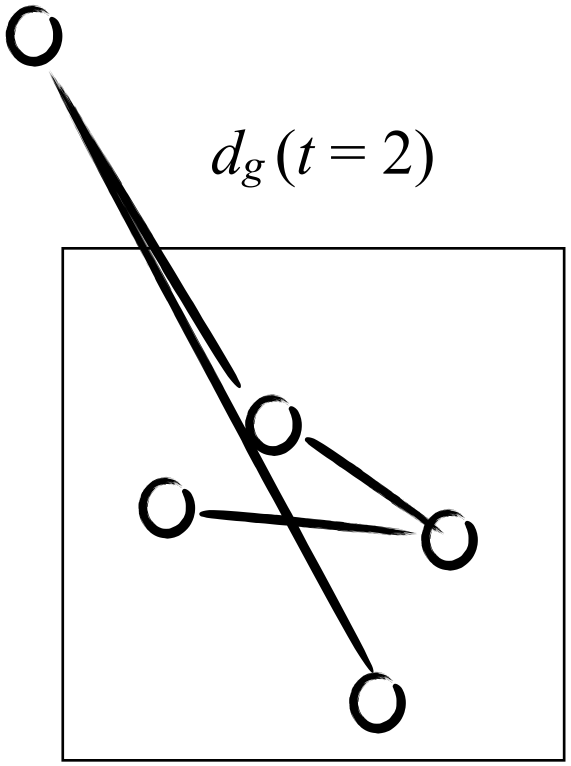
Layout optimization based on simulated annealing

- initial configuration: randomly distributed **vertices (and edges attached to them) on 2D space** inside the unit square, for a **given** network topology

- simulated annealing
  - trial movement: choose a random vertex with the coordinates (x_0, y_0)

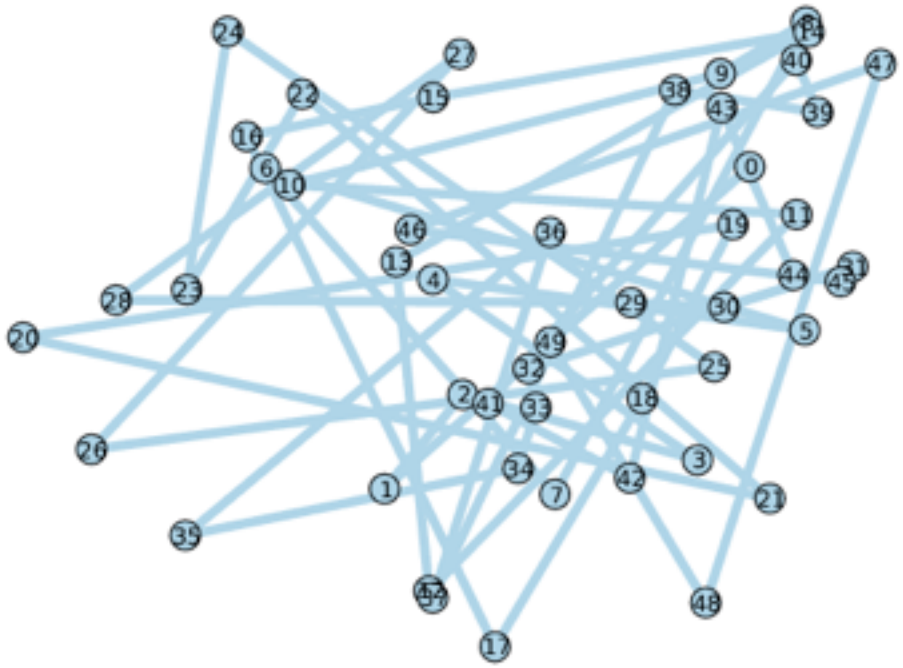
    $(x_0, y_0) \rightarrow (x_0 + \Delta x, y_0 + \Delta y)$ where Δx and Δy are uniformly randomly drawn from the interval $[-l, l]$

  - calculate the average (hopping-distance-based) GSN pathway d_g, which is the **object function** to be minimized
  - accept the movement if d_g is decreased, or with probability p otherwise
    - with $p = p_{\mathrm{high}}$ (heating) & $p = p_{\mathrm{low}}$ (quenching) repeatedly
  - record the **layout with the minimum d_g value**



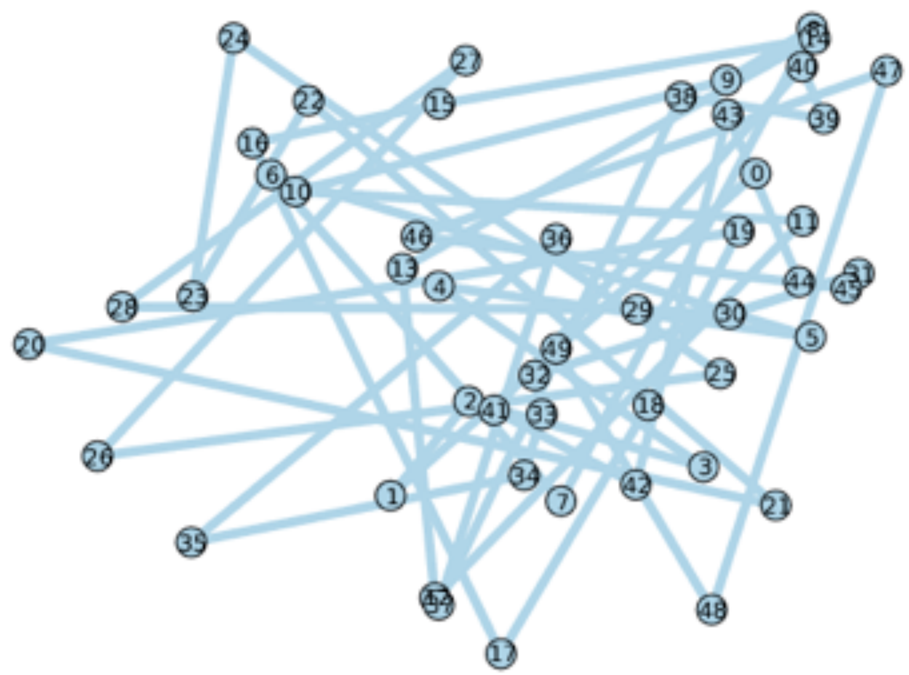
Real (but model graphs, though) examples . . .

Real (but model graphs, though) examples . . .

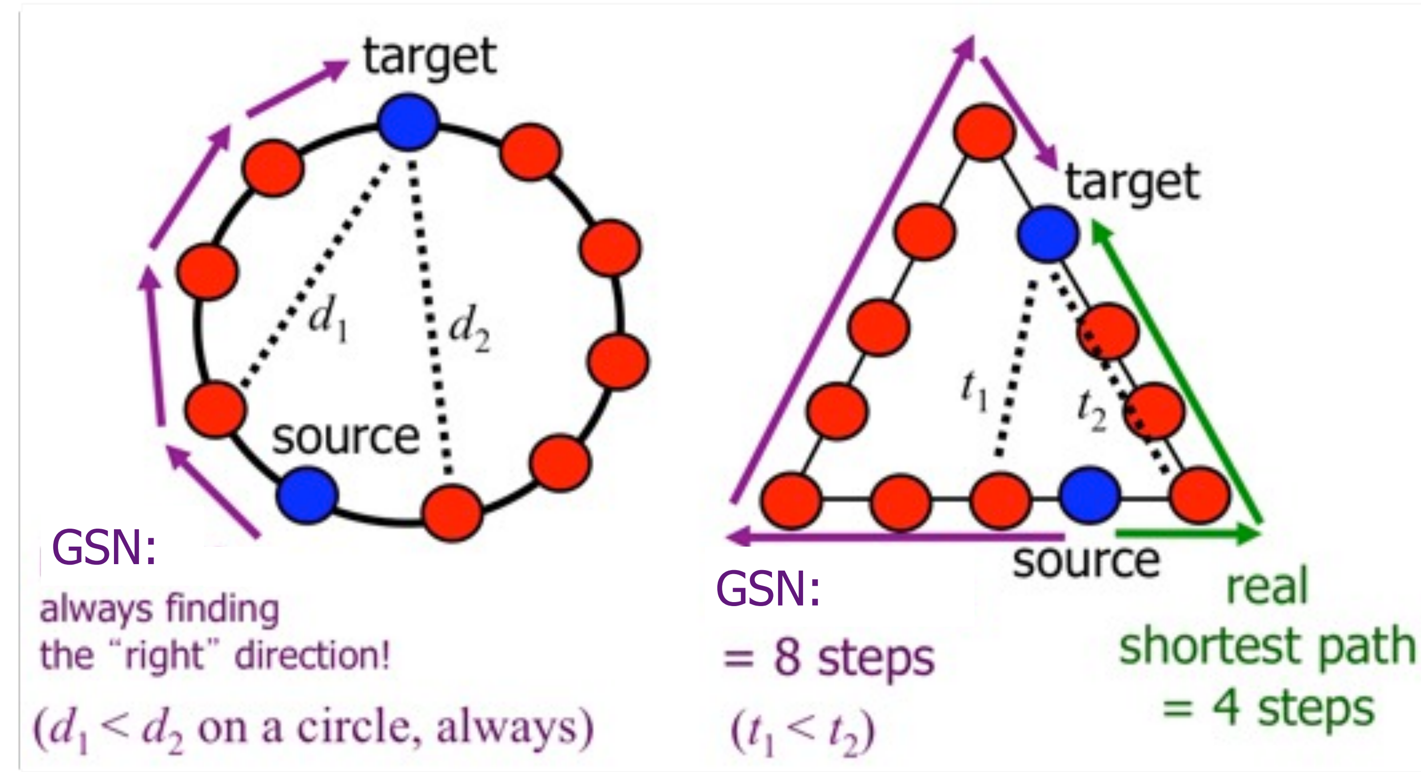


p=0.2, MC step #0: 23.9910204082 steps..

Real (but model graphs, though) examples . . .

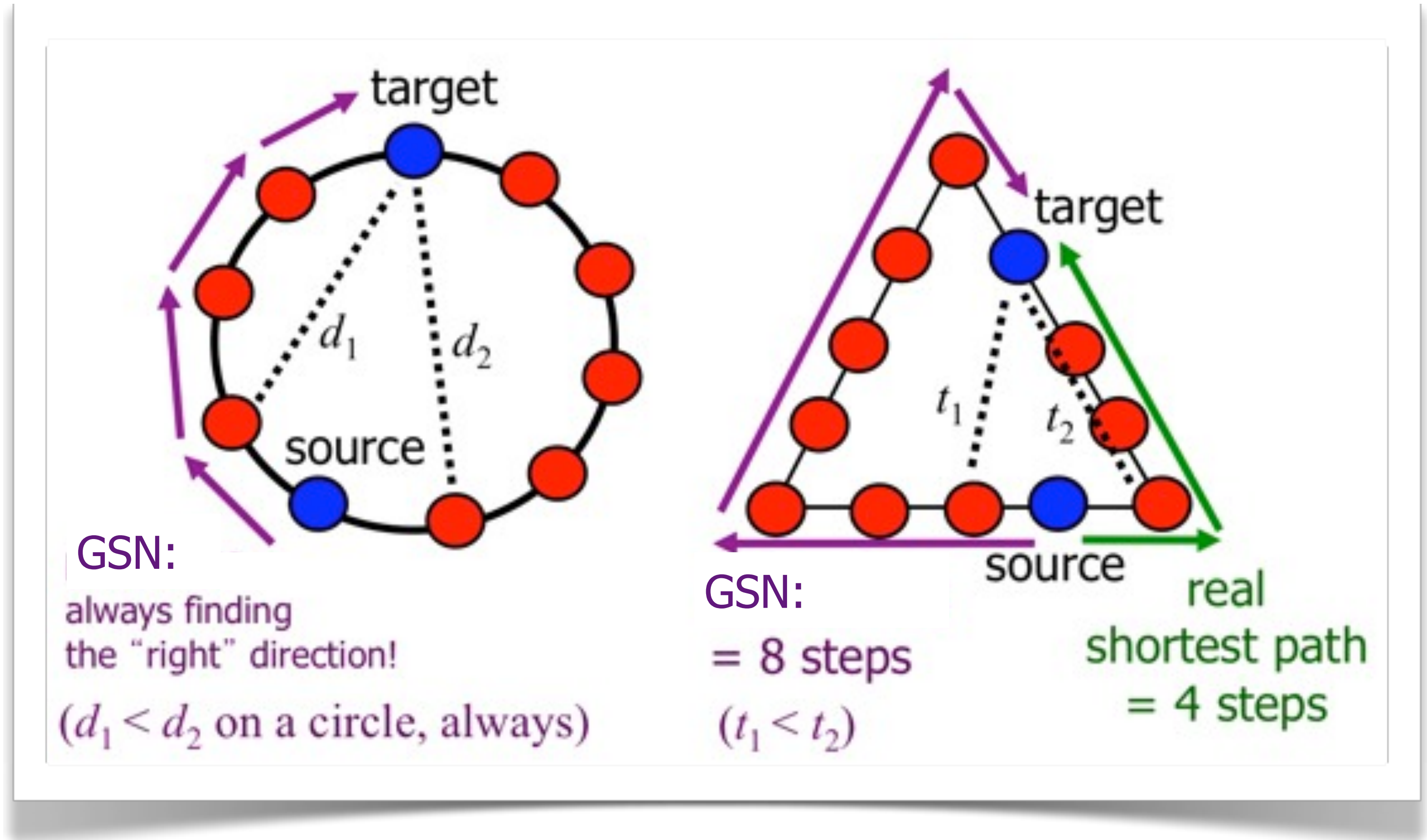


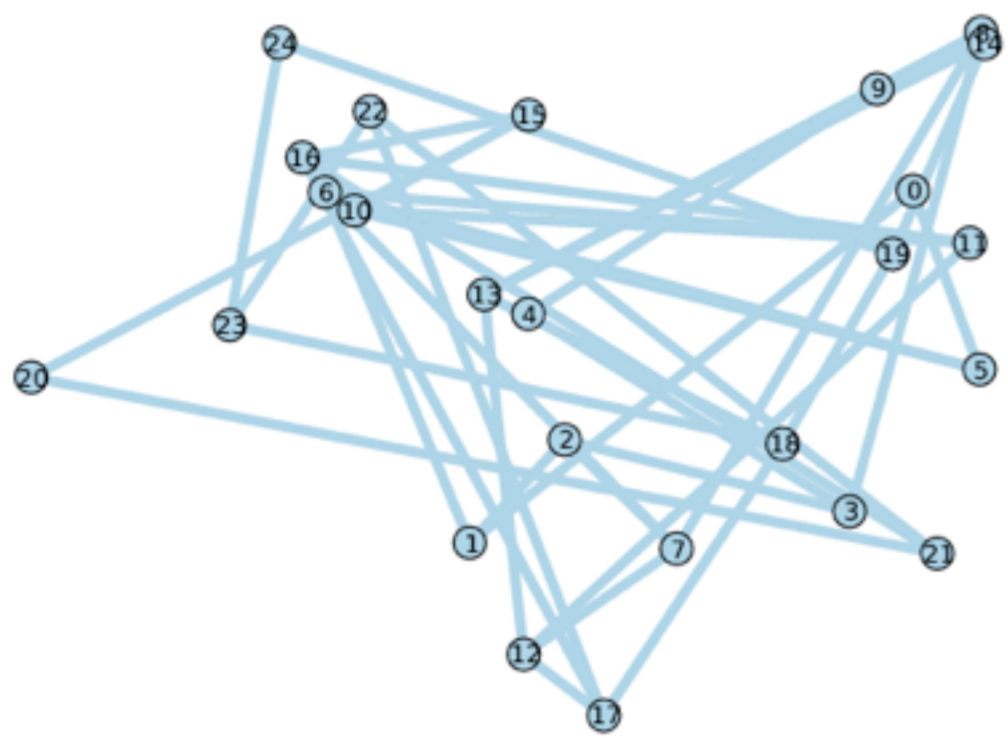
p=0.2, MC step #0: 23.9910204082 steps..



Real (but model graphs, though) examples . . .

p=0.2, MC step #0: 23.9910204082 steps..





p=0.1, MC step #0: 7.97166666667 steps..

Kamada-Kawai (KK) spring layout vs GSN-pathway-optimized layout

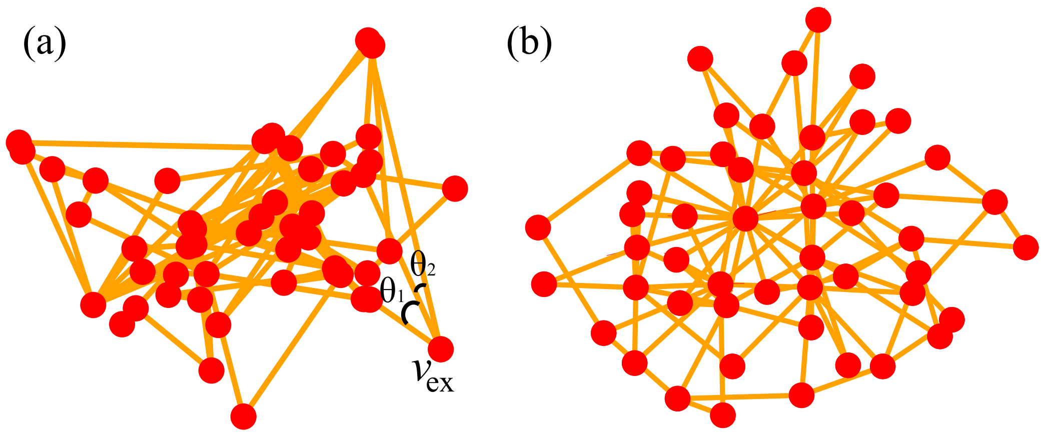


FIG. 1: (color online) Examples of the optimal (a) and KK layout (b) of the BA model. The GSN pathway is 3.85 (4.79) for the optimal (KK) layout, respectively.

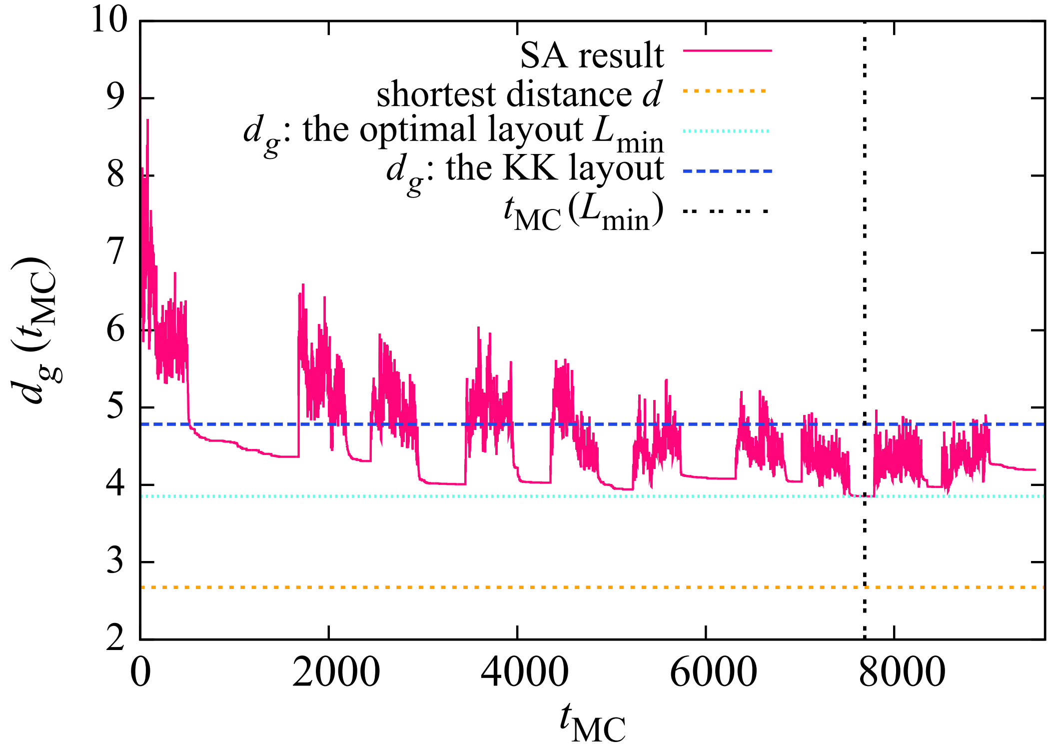


FIG. 2: (color online) A typical time series of d_g in the unit of MC steps t_{MC}, in case of the BA model used in Fig. 1, along with the real shortest path length d, d_g for the optimal layout $L_{\min}$ [Fig. 1(a)] and the KK layout [Fig. 1(b)]. The bursting part and almost flat plateau correspond to the heating (p_{high}) and quenching (p_{low}) processes, respectively. The moment of $L_{\min}$ denoted as the vertical line.

Kamada-Kawai (KK) spring layout vs GSN-pathway-optimized layout

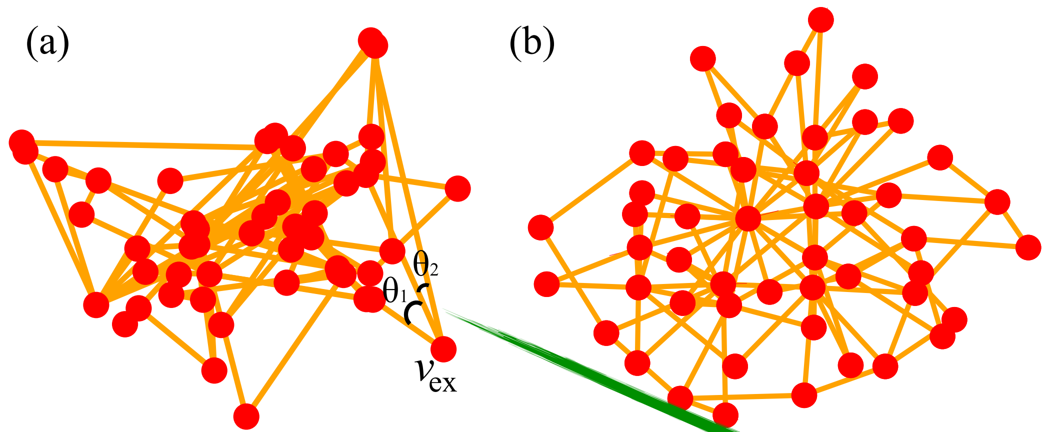


FIG. 1: (color online) Examples of the optimal (a) and KK layout (b) of the BA model. The GSN pathway is 3.85 (4.79) for the optimal (KK) layout, respectively.

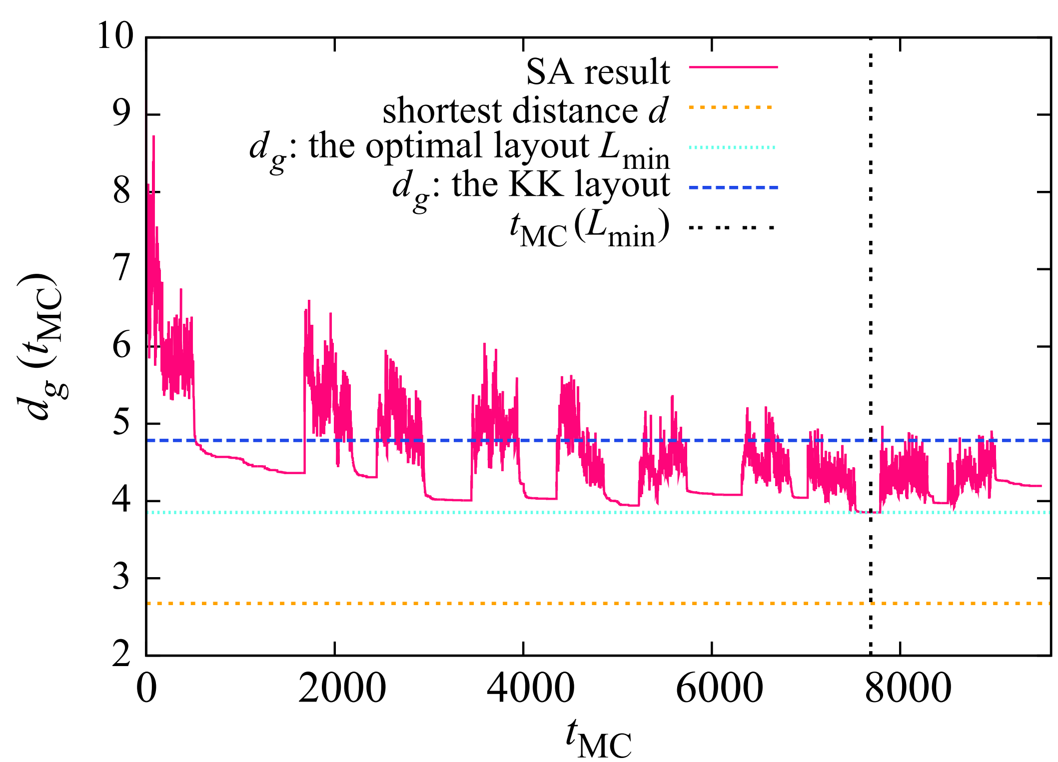


FIG. 2: (color online) A typical time series of d_g in the unit of MC steps t_{MC}, in case of the BA model used in Fig. 1, along with the real shortest path length d, d_g for the optimal layout $L_{\min}$ [Fig. 1(a)] and the KK layout [Fig. 1(b)]. The bursting part and almost flat plateau correspond to the heating (p_{high}) and quenching (p_{low}) processes, respectively. The moment of $L_{\min}$ denoted as the vertical line.

angle distribution

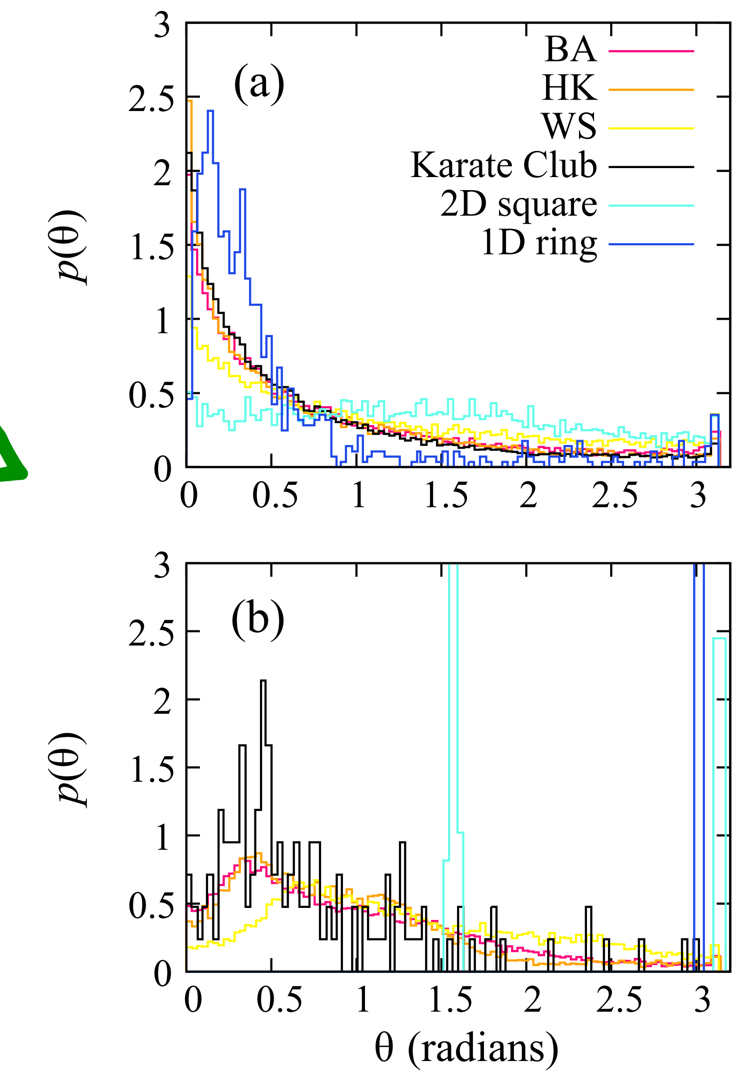


FIG. 3: (color online) Distribution of angles in the optimized (a) and KK (b) layout. At least 18 graph ensembles are used to average for all the cases.

looking at the GSN pathways in the optimized layout more closely . . .

average step decreased along the GSN pathways

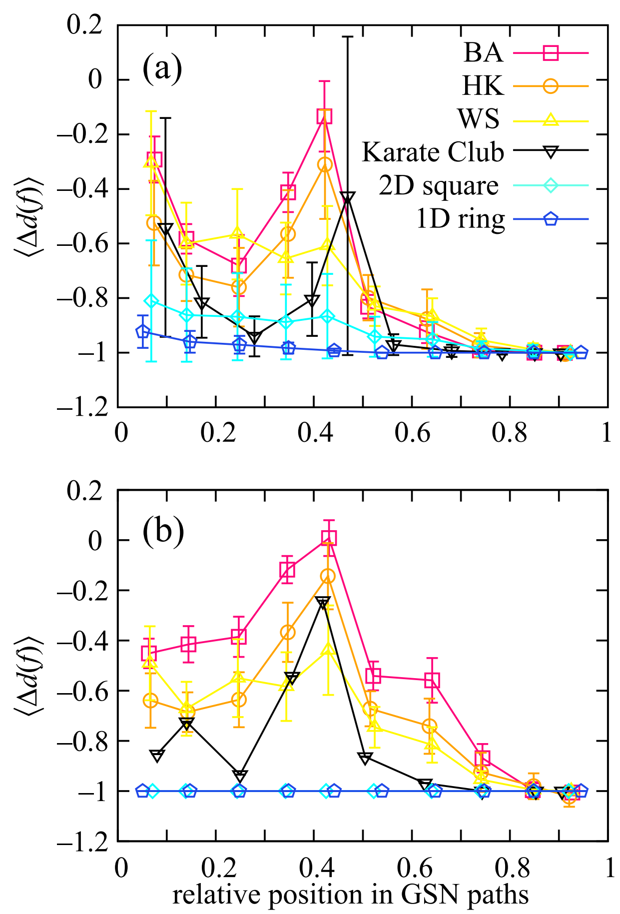


FIG. 4: (color online) Average step decreased along the GSN paths in terms of the relative position f along the pathways, in the optimized (a) and KK (b) layout. At least 18 graph ensembles are used to average for all the cases, and the horizontal axis is equipartitioned into appropriate bins. The error bars represent the standard deviation of the average values of $\Delta d(f)$ for each graph, over different graph ensembles.

looking at the GSN pathways in the optimized layout more closely . . .

average step decreased along the GSN pathways

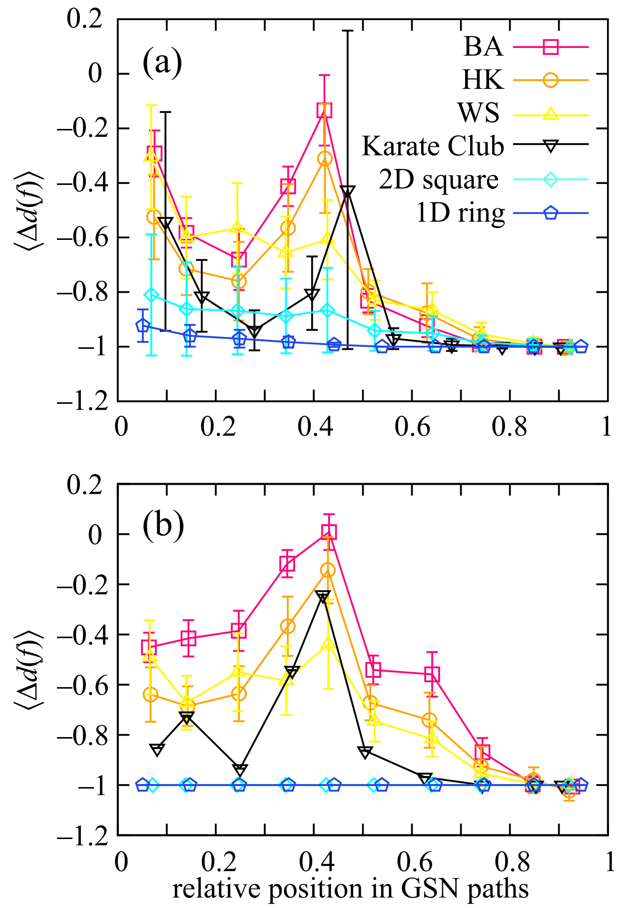


FIG. 4: (color online) Average step decreased along the GSN paths in terms of the relative position f along the pathways, in the optimized (a) and KK (b) layout. At least 18 graph ensembles are used to average for all the cases, and the horizontal axis is equipartitioned into appropriate bins. The error bars represent the standard deviation of the average values of $\Delta d(f)$ for each graph, over different graph ensembles.

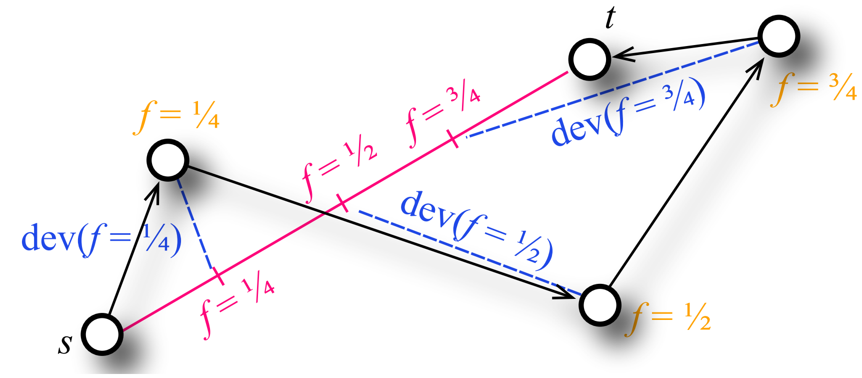


FIG. 5: (color online) An illustration of the definition of deviation from the straight line for a s–t pair. Each intermediate vertex's location in the GSN pathway of length 4 (arrows) is compared to the corresponding intermediate points ($f = 1/4, 1/2$ and $3/4$) equally spaced in the straight line (red line). The deviation for each intermediate point is defined as the Euclidean distance between the two points (blue dashed lines), in the unit of the straight line.

looking at the GSN pathways in the optimized layout more closely . . .

average step decreased along the GSN pathways

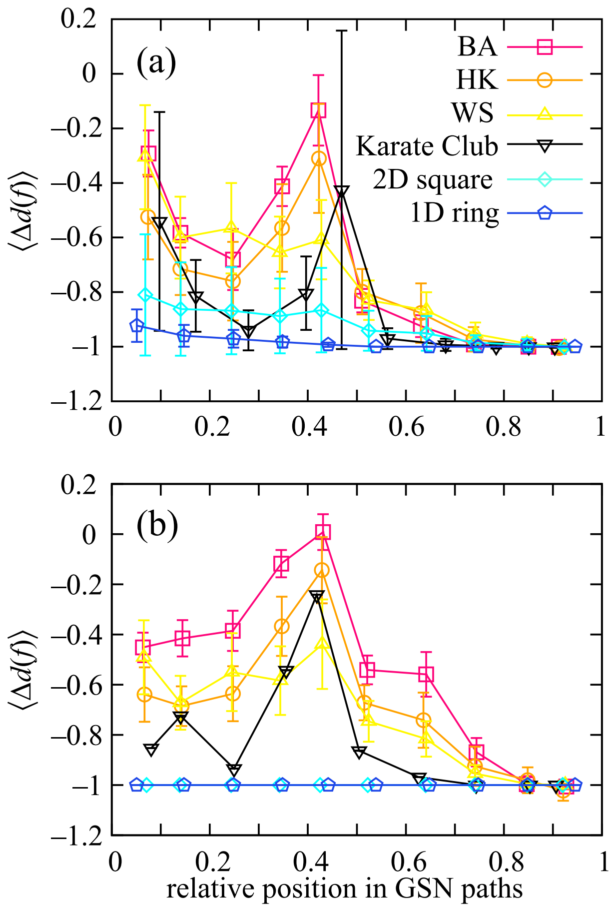


FIG. 4: (color online) Average step decreased along the GSN paths in terms of the relative position f along the pathways, in the optimized (a) and KK (b) layout. At least 18 graph ensembles are used to average for all the cases, and the horizontal axis is equipartitioned into appropriate bins. The error bars represent the standard deviation of the average values of $\Delta d(f)$ for each graph, over different graph ensembles.

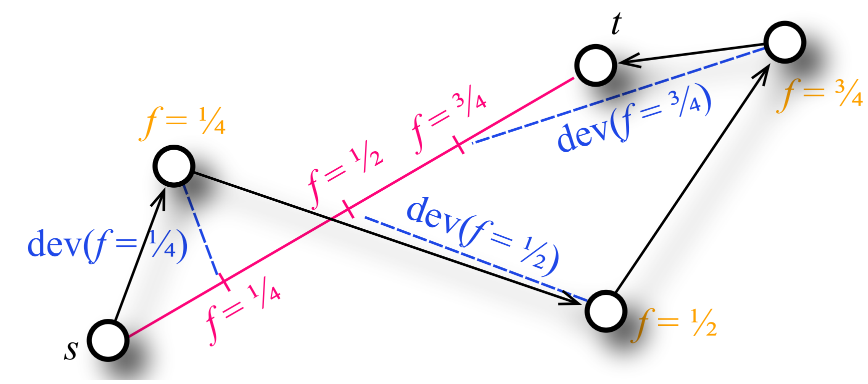


FIG. 5: (color online) An illustration of the definition of deviation from the straight line for a s–t pair. Each intermediate vertex's location in the GSN pathway of length 4 (arrows) is compared to the corresponding intermediate points ($f = 1/4, 1/2$ and $3/4$) equally spaced in the straight line (red line). The deviation for each intermediate point is defined as the Euclidean distance between the two points (blue dashed lines), in the unit of the straight line.

average deviation from the straight line connecting s-t pairs

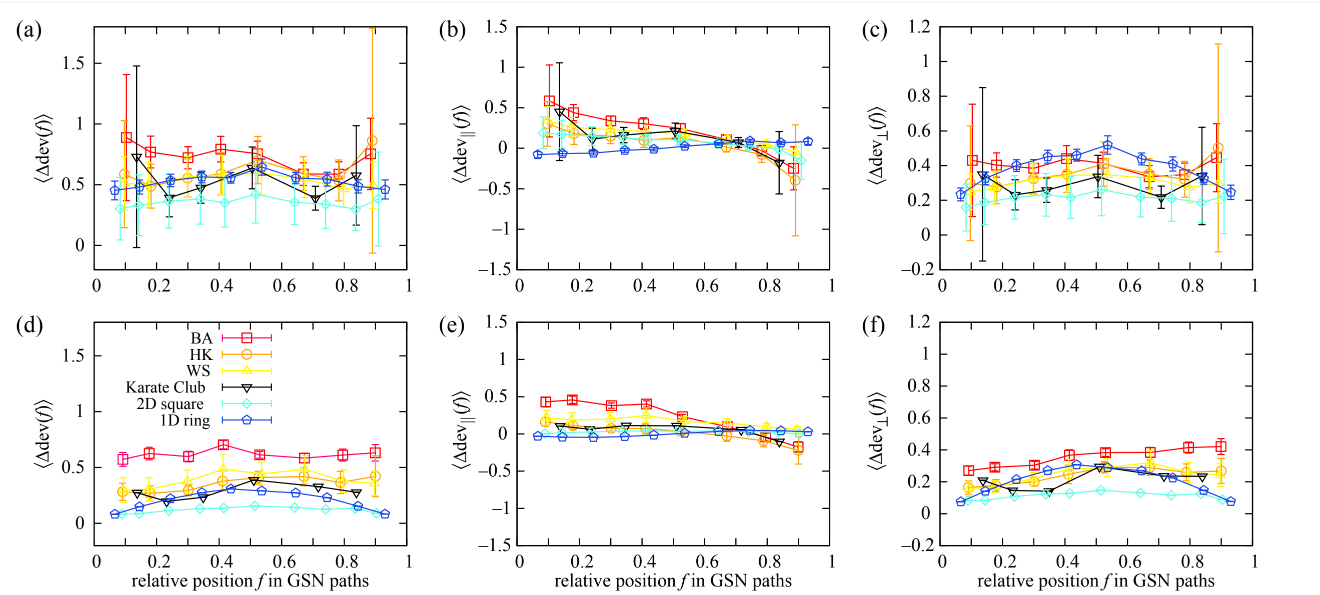


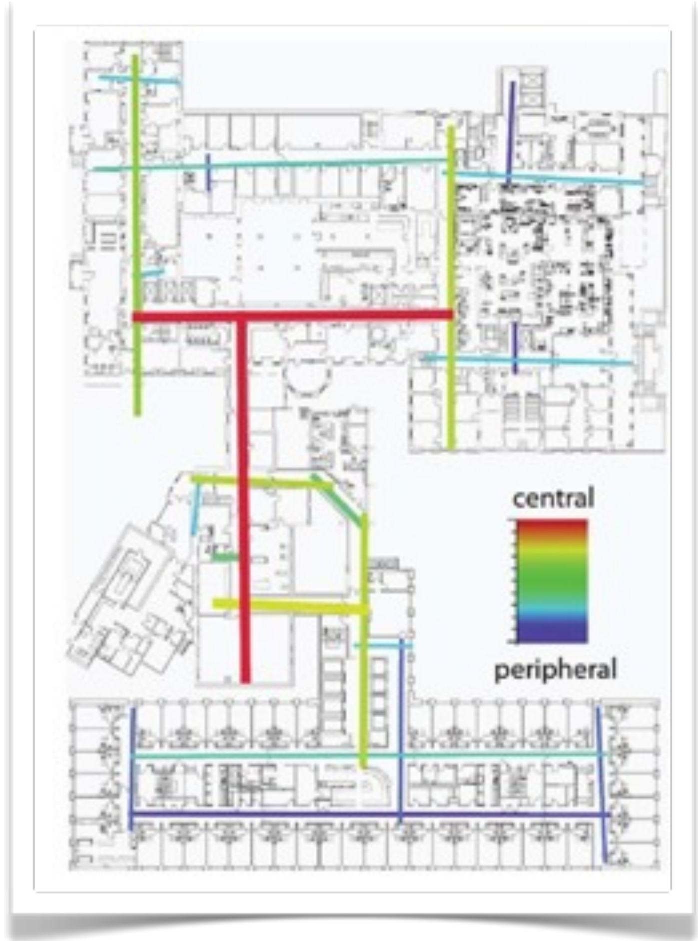
FIG. 6: (color online) Average deviation from the straight line connecting s–t pairs, for the relative position x in the GSN pathways and straight lines. Upper (lower) panels correspond to the optimized (KK) layouts, respectively. $\langle \mathrm{dev}(f) \rangle$ [(a) and (d)] is decomposed into $\langle \mathrm{dev}_{\parallel}(f) \rangle$ [(b) and (e)] and $\langle \mathrm{dev}_{\perp}(f) \rangle$ [(c) and (f)] with respect to the straight line. At least 18 graph ensembles are used to average for all the cases. The horizontal axis is binned into appropriate equidistant intervals, and the error bars represent the standard deviation of the average values of measures for each graph, over different graph ensembles.

SHL and P. Holme, arXiv:1206.6651.

User-centric approach in architecture

- Lost in buildings? Why are some buildings hard to “navigate?”

Seattle Central Library



the study of man

Human Movement and Architecture

ROBERT B. BECHTEL

"What other monumental interior in America produces such an overwhelming effect?" critic Lewis Mumford has asked of New York's Guggenheim Museum. "You may go to this building to see Kandinsky or Jackson Pollock; you remain to see Frank Lloyd Wright."

A construction worker on architect Wright's gigantic spiral expressed other ideas when the now-famous building was going up in 1957:

The way I figure it is that this is the screwiest project I ever got tied up in. The whole joint goes round and round and round and where it comes out nobody knows. (*The New Yorker*)

Architects have long been interested in knowing how their designs affect the traffic pattern within a building and

MAY

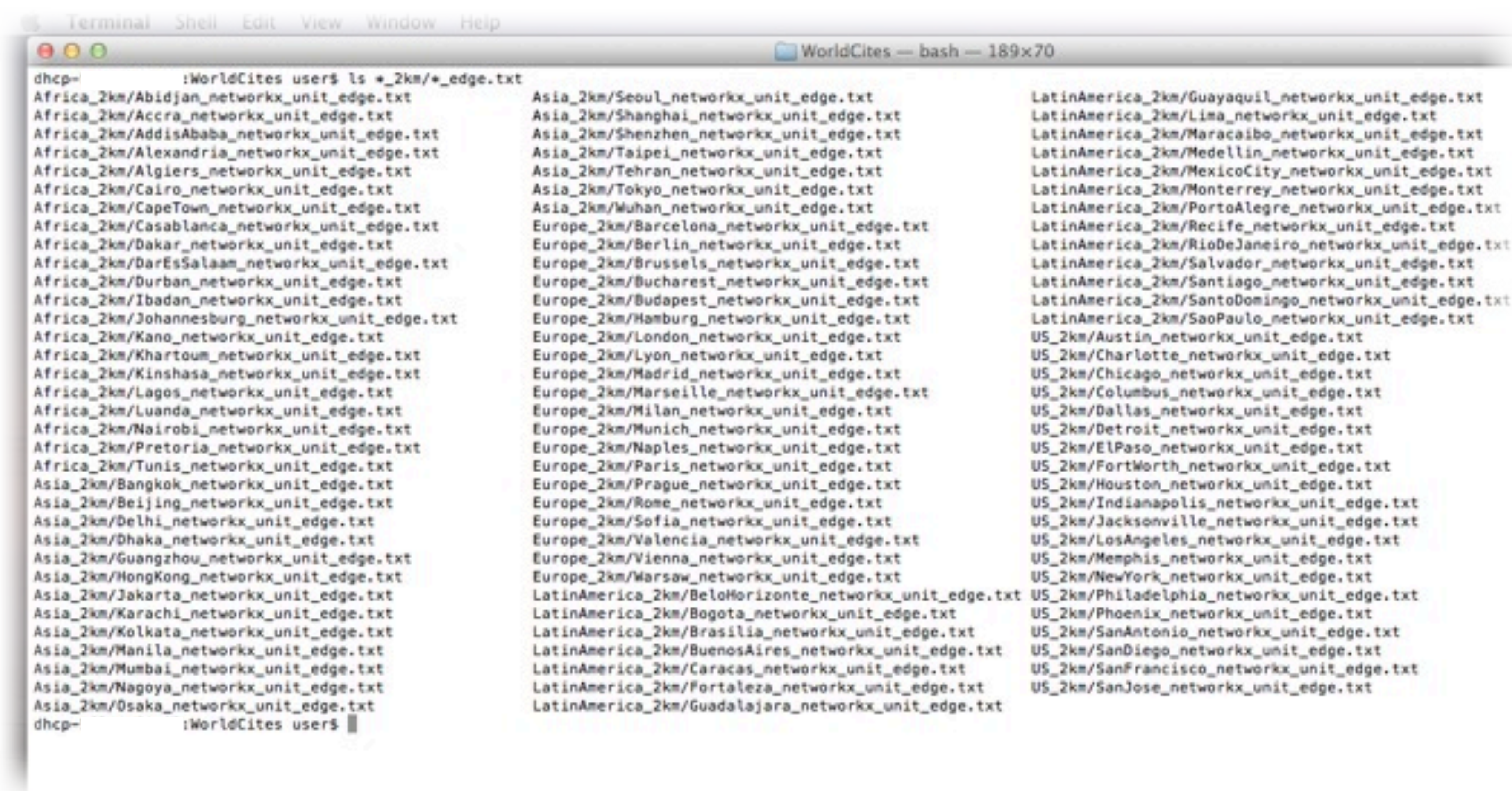
how utilization of avail
fective by proper desig
scientists have also bec
respond to architectural
some buildings more co
stimulating than others?
ently in these different

Perception and Locom

When a person mov
responding to the build
definition of interior s
drawing board or a mo
viewed from the street.

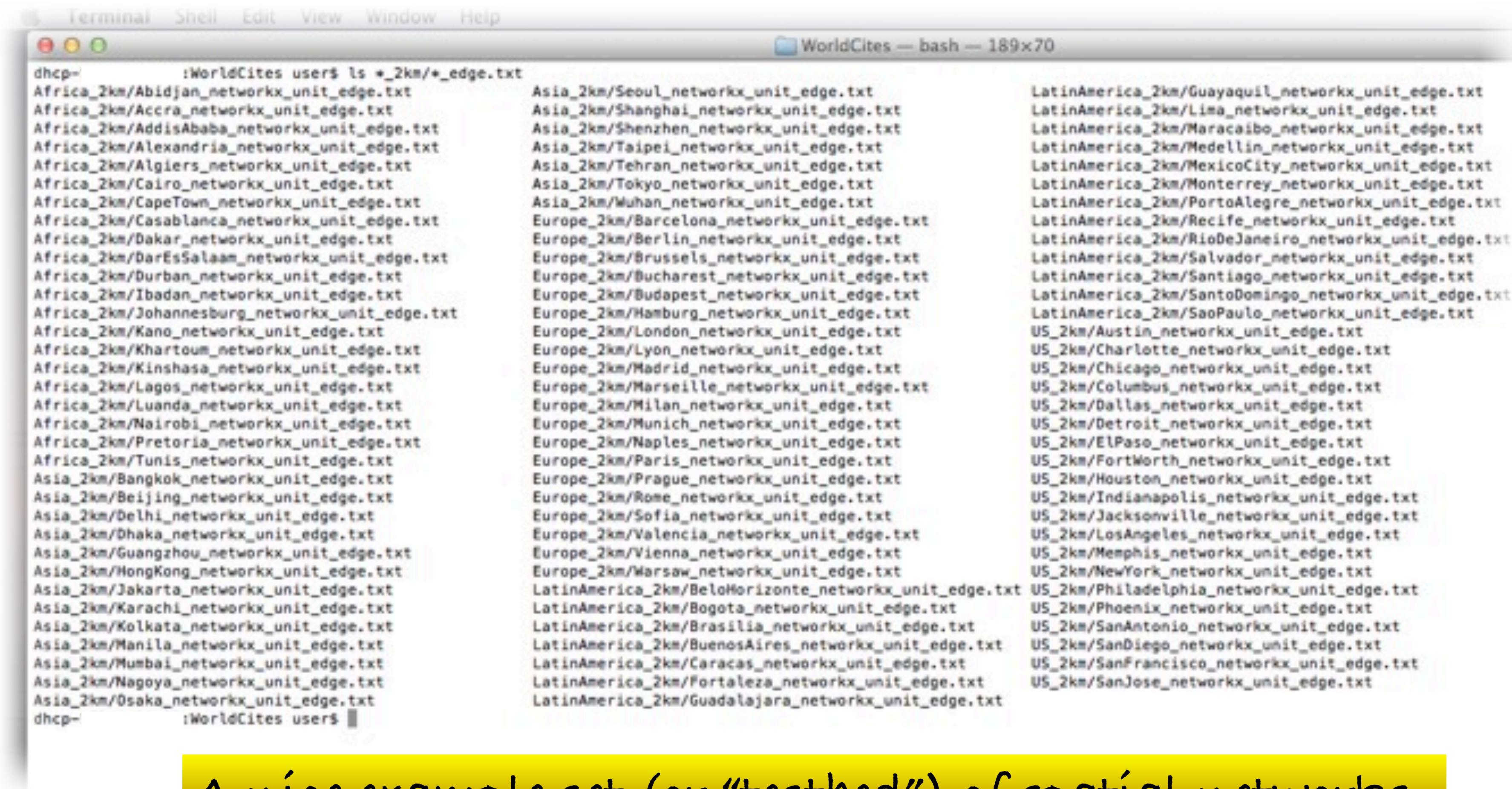
It might look very cool, but
it might feel like a maze for
general public/greedy navigators!

Now, I’m here in Oxford, with this road network data from 100 cities . . .




```
dhcp-          :WorldCites user$ ls *_2km/*_edge.txt
Africa_2km/Abidjan_networkx_unit_edge.txt
Africa_2km/Accra_networkx_unit_edge.txt
Africa_2km/AddisAbaba_networkx_unit_edge.txt
Africa_2km/Alexandria_networkx_unit_edge.txt
Africa_2km/Algiers_networkx_unit_edge.txt
Africa_2km/Cairo_networkx_unit_edge.txt
Africa_2km/CapeTown_networkx_unit_edge.txt
Africa_2km/Casablanca_networkx_unit_edge.txt
Africa_2km/Dakar_networkx_unit_edge.txt
Africa_2km/DarEsSalaam_networkx_unit_edge.txt
Africa_2km/Durban_networkx_unit_edge.txt
Africa_2km/Ibadan_networkx_unit_edge.txt
Africa_2km/Johannesburg_networkx_unit_edge.txt
Africa_2km/Kano_networkx_unit_edge.txt
Africa_2km/Khartoum_networkx_unit_edge.txt
Africa_2km/Kinshasa_networkx_unit_edge.txt
Africa_2km/Lagos_networkx_unit_edge.txt
Africa_2km/Luanda_networkx_unit_edge.txt
Africa_2km/Nairobi_networkx_unit_edge.txt
Africa_2km/Pretoria_networkx_unit_edge.txt
Africa_2km/Tunis_networkx_unit_edge.txt
Asia_2km/Bangkok_networkx_unit_edge.txt
Asia_2km/Beijing_networkx_unit_edge.txt
Asia_2km/Delhi_networkx_unit_edge.txt
Asia_2km/Dhaka_networkx_unit_edge.txt
Asia_2km/Guangzhou_networkx_unit_edge.txt
Asia_2km/HongKong_networkx_unit_edge.txt
Asia_2km/Jakarta_networkx_unit_edge.txt
Asia_2km/Karachi_networkx_unit_edge.txt
Asia_2km/Kolkata_networkx_unit_edge.txt
Asia_2km/Manila_networkx_unit_edge.txt
Asia_2km/Mumbai_networkx_unit_edge.txt
Asia_2km/Nagoya_networkx_unit_edge.txt
Asia_2km/Osaka_networkx_unit_edge.txt
Asia_2km/Seoul_networkx_unit_edge.txt
Asia_2km/Shanghai_networkx_unit_edge.txt
Asia_2km/Shenzhen_networkx_unit_edge.txt
Asia_2km/Taipei_networkx_unit_edge.txt
Asia_2km/Tehran_networkx_unit_edge.txt
Asia_2km/Tokyo_networkx_unit_edge.txt
Asia_2km/Wuhan_networkx_unit_edge.txt
Europe_2km/Barcelona_networkx_unit_edge.txt
Europe_2km/Berlin_networkx_unit_edge.txt
Europe_2km/Brussels_networkx_unit_edge.txt
Europe_2km/Bucharest_networkx_unit_edge.txt
Europe_2km/Budapest_networkx_unit_edge.txt
Europe_2km/Hamburg_networkx_unit_edge.txt
Europe_2km/London_networkx_unit_edge.txt
Europe_2km/Lyon_networkx_unit_edge.txt
Europe_2km/Madrid_networkx_unit_edge.txt
Europe_2km/Marseille_networkx_unit_edge.txt
Europe_2km/Milan_networkx_unit_edge.txt
Europe_2km/Munich_networkx_unit_edge.txt
Europe_2km/Naples_networkx_unit_edge.txt
Europe_2km/Paris_networkx_unit_edge.txt
Europe_2km/Prague_networkx_unit_edge.txt
Europe_2km/Rome_networkx_unit_edge.txt
Europe_2km/Sofia_networkx_unit_edge.txt
Europe_2km/Valencia_networkx_unit_edge.txt
Europe_2km/Vienna_networkx_unit_edge.txt
Europe_2km/Warsaw_networkx_unit_edge.txt
LatinAmerica_2km/BeloHorizonte_networkx_unit_edge.txt
LatinAmerica_2km/Bogota_networkx_unit_edge.txt
LatinAmerica_2km/Brasilia_networkx_unit_edge.txt
LatinAmerica_2km/BuenosAires_networkx_unit_edge.txt
LatinAmerica_2km/Caracas_networkx_unit_edge.txt
LatinAmerica_2km/Fortaleza_networkx_unit_edge.txt
LatinAmerica_2km/Guadalajara_networkx_unit_edge.txt
LatinAmerica_2km/Guayaquil_networkx_unit_edge.txt
LatinAmerica_2km/Lima_networkx_unit_edge.txt
LatinAmerica_2km/Maracaibo_networkx_unit_edge.txt
LatinAmerica_2km/Medellin_networkx_unit_edge.txt
LatinAmerica_2km/MexicoCity_networkx_unit_edge.txt
LatinAmerica_2km/Monterrey_networkx_unit_edge.txt
LatinAmerica_2km/PortoAlegre_networkx_unit_edge.txt
LatinAmerica_2km/Recife_networkx_unit_edge.txt
LatinAmerica_2km/RioDeJaneiro_networkx_unit_edge.txt
LatinAmerica_2km/Salvador_networkx_unit_edge.txt
LatinAmerica_2km/Santiago_networkx_unit_edge.txt
LatinAmerica_2km/SantoDomingo_networkx_unit_edge.txt
LatinAmerica_2km/SaoPaulo_networkx_unit_edge.txt
US_2km/Austin_networkx_unit_edge.txt
US_2km/Charlotte_networkx_unit_edge.txt
US_2km/Chicago_networkx_unit_edge.txt
US_2km/Columbus_networkx_unit_edge.txt
US_2km/Dallas_networkx_unit_edge.txt
US_2km/Detroit_networkx_unit_edge.txt
US_2km/ElPaso_networkx_unit_edge.txt
US_2km/FortWorth_networkx_unit_edge.txt
US_2km/Houston_networkx_unit_edge.txt
US_2km/Indianapolis_networkx_unit_edge.txt
US_2km/Jacksonville_networkx_unit_edge.txt
US_2km/LosAngeles_networkx_unit_edge.txt
US_2km/Memphis_networkx_unit_edge.txt
US_2km/NewYork_networkx_unit_edge.txt
US_2km/Philadelphia_networkx_unit_edge.txt
US_2km/Phoenix_networkx_unit_edge.txt
US_2km/SanAntonio_networkx_unit_edge.txt
US_2km/SanDiego_networkx_unit_edge.txt
US_2km/SanFrancisco_networkx_unit_edge.txt
US_2km/SanJose_networkx_unit_edge.txt
dhcp-          :WorldCites user$
```

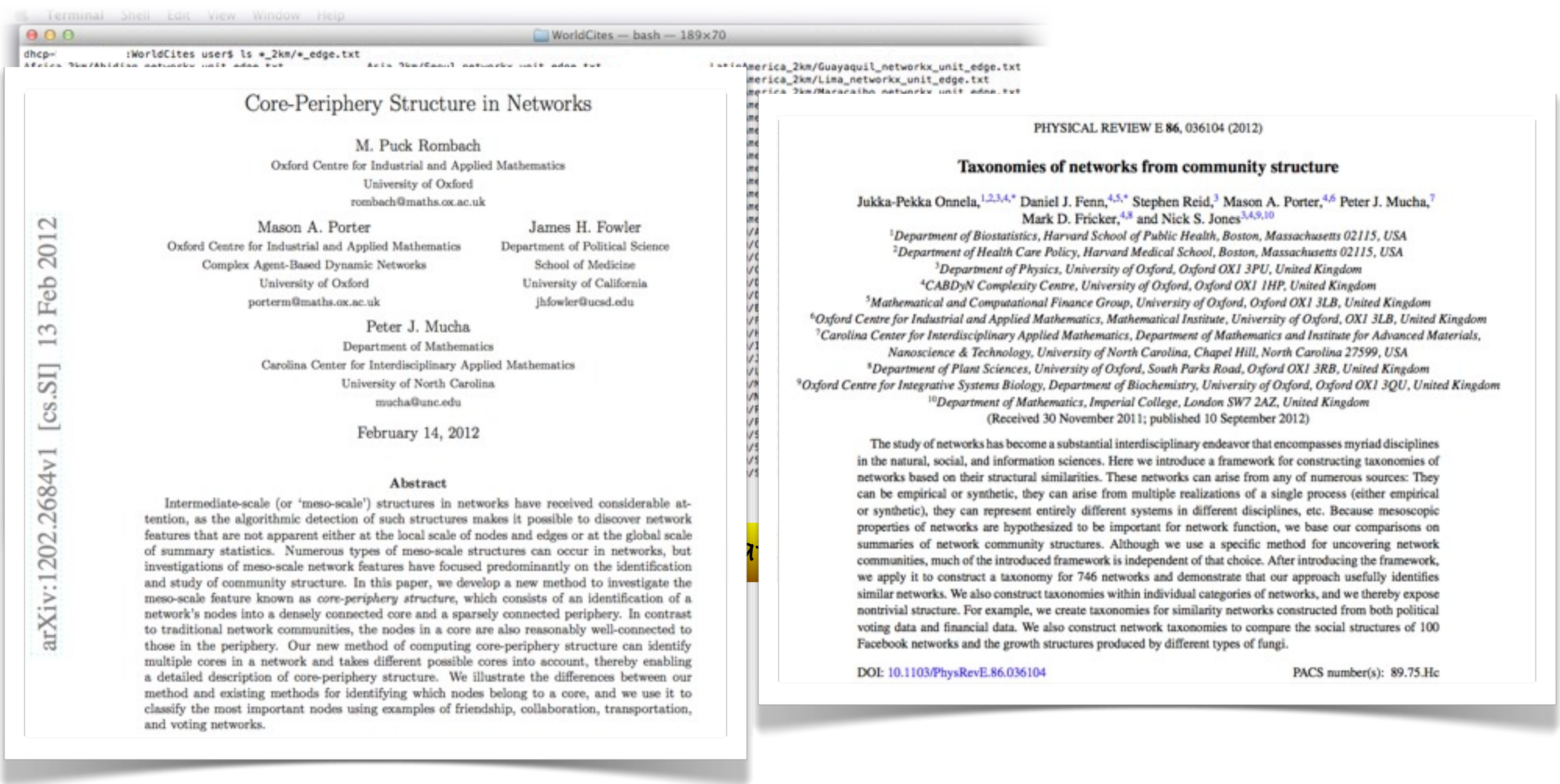
Now, I'm here in Oxford, with this road network data from 100 cities . . .



```
dhcp-        :WorldCites users$ ls *_2km/*_edge.txt
Africa_2km/Abidjan_networkx_unit_edge.txt
Africa_2km/Accra_networkx_unit_edge.txt
Africa_2km/AddisAbaba_networkx_unit_edge.txt
Africa_2km/Alexandria_networkx_unit_edge.txt
Africa_2km/Algiers_networkx_unit_edge.txt
Africa_2km/Cairo_networkx_unit_edge.txt
Africa_2km/CapeTown_networkx_unit_edge.txt
Africa_2km/Casablanca_networkx_unit_edge.txt
Africa_2km/Dakar_networkx_unit_edge.txt
Africa_2km/DarEsSalaam_networkx_unit_edge.txt
Africa_2km/Durban_networkx_unit_edge.txt
Africa_2km/Ibadan_networkx_unit_edge.txt
Africa_2km/Johannesburg_networkx_unit_edge.txt
Africa_2km/Kano_networkx_unit_edge.txt
Africa_2km/Khartoum_networkx_unit_edge.txt
Africa_2km/Kinshasa_networkx_unit_edge.txt
Africa_2km/Lagos_networkx_unit_edge.txt
Africa_2km/Luanda_networkx_unit_edge.txt
Africa_2km/Nairobi_networkx_unit_edge.txt
Africa_2km/Pretoria_networkx_unit_edge.txt
Africa_2km/Tunis_networkx_unit_edge.txt
Asia_2km/Bangkok_networkx_unit_edge.txt
Asia_2km/Beijing_networkx_unit_edge.txt
Asia_2km/Delhi_networkx_unit_edge.txt
Asia_2km/Dhaka_networkx_unit_edge.txt
Asia_2km/Guangzhou_networkx_unit_edge.txt
Asia_2km/HongKong_networkx_unit_edge.txt
Asia_2km/Jakarta_networkx_unit_edge.txt
Asia_2km/Karachi_networkx_unit_edge.txt
Asia_2km/Kolkata_networkx_unit_edge.txt
Asia_2km/Manila_networkx_unit_edge.txt
Asia_2km/Mumbai_networkx_unit_edge.txt
Asia_2km/Nagoya_networkx_unit_edge.txt
Asia_2km/Osaka_networkx_unit_edge.txt
Asia_2km/Seoul_networkx_unit_edge.txt
Asia_2km/Shanghai_networkx_unit_edge.txt
Asia_2km/Shenzhen_networkx_unit_edge.txt
Asia_2km/Taipei_networkx_unit_edge.txt
Asia_2km/Tehran_networkx_unit_edge.txt
Asia_2km/Tokyo_networkx_unit_edge.txt
Asia_2km/Wuhan_networkx_unit_edge.txt
Europe_2km/Barcelona_networkx_unit_edge.txt
Europe_2km/Berlin_networkx_unit_edge.txt
Europe_2km/Brussels_networkx_unit_edge.txt
Europe_2km/Bucharest_networkx_unit_edge.txt
Europe_2km/Budapest_networkx_unit_edge.txt
Europe_2km/Hamburg_networkx_unit_edge.txt
Europe_2km/London_networkx_unit_edge.txt
Europe_2km/Lyon_networkx_unit_edge.txt
Europe_2km/Madrid_networkx_unit_edge.txt
Europe_2km/Marseille_networkx_unit_edge.txt
Europe_2km/Milan_networkx_unit_edge.txt
Europe_2km/Munich_networkx_unit_edge.txt
Europe_2km/Naples_networkx_unit_edge.txt
Europe_2km/Paris_networkx_unit_edge.txt
Europe_2km/Prague_networkx_unit_edge.txt
Europe_2km/Rome_networkx_unit_edge.txt
Europe_2km/Sofia_networkx_unit_edge.txt
Europe_2km/Valencia_networkx_unit_edge.txt
Europe_2km/Vienna_networkx_unit_edge.txt
Europe_2km/Warsaw_networkx_unit_edge.txt
LatinAmerica_2km/BeloHorizonte_networkx_unit_edge.txt
LatinAmerica_2km/Bogota_networkx_unit_edge.txt
LatinAmerica_2km/Brasilia_networkx_unit_edge.txt
LatinAmerica_2km/BuenosAires_networkx_unit_edge.txt
LatinAmerica_2km/Caracas_networkx_unit_edge.txt
LatinAmerica_2km/Fortaleza_networkx_unit_edge.txt
LatinAmerica_2km/Guadalajara_networkx_unit_edge.txt
LatinAmerica_2km/Guayaquil_networkx_unit_edge.txt
LatinAmerica_2km/Lima_networkx_unit_edge.txt
LatinAmerica_2km/Maracaibo_networkx_unit_edge.txt
LatinAmerica_2km/Medellin_networkx_unit_edge.txt
LatinAmerica_2km/MexicoCity_networkx_unit_edge.txt
LatinAmerica_2km/Monterrey_networkx_unit_edge.txt
LatinAmerica_2km/PortoAlegre_networkx_unit_edge.txt
LatinAmerica_2km/Recife_networkx_unit_edge.txt
LatinAmerica_2km/RioDeJaneiro_networkx_unit_edge.txt
LatinAmerica_2km/Salvador_networkx_unit_edge.txt
LatinAmerica_2km/Santiago_networkx_unit_edge.txt
LatinAmerica_2km/SantoDomingo_networkx_unit_edge.txt
LatinAmerica_2km/SaoPaulo_networkx_unit_edge.txt
US_2km/Austin_networkx_unit_edge.txt
US_2km/Charlotte_networkx_unit_edge.txt
US_2km/Chicago_networkx_unit_edge.txt
US_2km/Columbus_networkx_unit_edge.txt
US_2km/Dallas_networkx_unit_edge.txt
US_2km/Detroit_networkx_unit_edge.txt
US_2km/ElPaso_networkx_unit_edge.txt
US_2km/FortWorth_networkx_unit_edge.txt
US_2km/Houston_networkx_unit_edge.txt
US_2km/Indianapolis_networkx_unit_edge.txt
US_2km/Jacksonville_networkx_unit_edge.txt
US_2km/LosAngeles_networkx_unit_edge.txt
US_2km/Memphis_networkx_unit_edge.txt
US_2km/NewYork_networkx_unit_edge.txt
US_2km/Philadelphia_networkx_unit_edge.txt
US_2km/Phoenix_networkx_unit_edge.txt
US_2km/SanAntonio_networkx_unit_edge.txt
US_2km/SanDiego_networkx_unit_edge.txt
US_2km/SanFrancisco_networkx_unit_edge.txt
US_2km/SanJose_networkx_unit_edge.txt
dhcp-        :WorldCites users$
```

A nice example set (or "testbed") of spatial networks

Now, I'm here in Oxford, with this road network data from 100 cities . . .



arXiv:1202.2684v1 [cs.SI] 13 Feb 2012

Core-Periphery Structure in Networks

M. Puck Rombach
Oxford Centre for Industrial and Applied Mathematics
University of Oxford
rombach@maths.ox.ac.uk

Mason A. Porter
Oxford Centre for Industrial and Applied Mathematics
Complex Agent-Based Dynamic Networks
University of Oxford
porterm@maths.ox.ac.uk

James H. Fowler
Department of Political Science
School of Medicine
University of California
jhfowler@ucsd.edu

Peter J. Mucha
Department of Mathematics
Carolina Center for Interdisciplinary Applied Mathematics
University of North Carolina
mucha@unc.edu

February 14, 2012

Abstract

Intermediate-scale (or 'meso-scale') structures in networks have received considerable attention, as the algorithmic detection of such structures makes it possible to discover network features that are not apparent either at the local scale of nodes and edges or at the global scale of summary statistics. Numerous types of meso-scale structures can occur in networks, but investigations of meso-scale network features have focused predominantly on the identification and study of community structure. In this paper, we develop a new method to investigate the meso-scale feature known as *core-periphery structure*, which consists of an identification of a network's nodes into a densely connected core and a sparsely connected periphery. In contrast to traditional network communities, the nodes in a core are also reasonably well-connected to those in the periphery. Our new method of computing core-periphery structure can identify multiple cores in a network and takes different possible cores into account, thereby enabling a detailed description of core-periphery structure. We illustrate the differences between our method and existing methods for identifying which nodes belong to a core, and we use it to classify the most important nodes using examples of friendship, collaboration, transportation, and voting networks.

PHYSICAL REVIEW E **86**, 036104 (2012)

Taxonomies of networks from community structure

Jukka-Pekka Onnela,[1,2,3,4,*] Daniel J. Fenn,[4,5,*] Stephen Reid,[3] Mason A. Porter,[4,6] Peter J. Mucha,[7] Mark D. Fricker,[4,8] and Nick S. Jones[3,4,9,10]

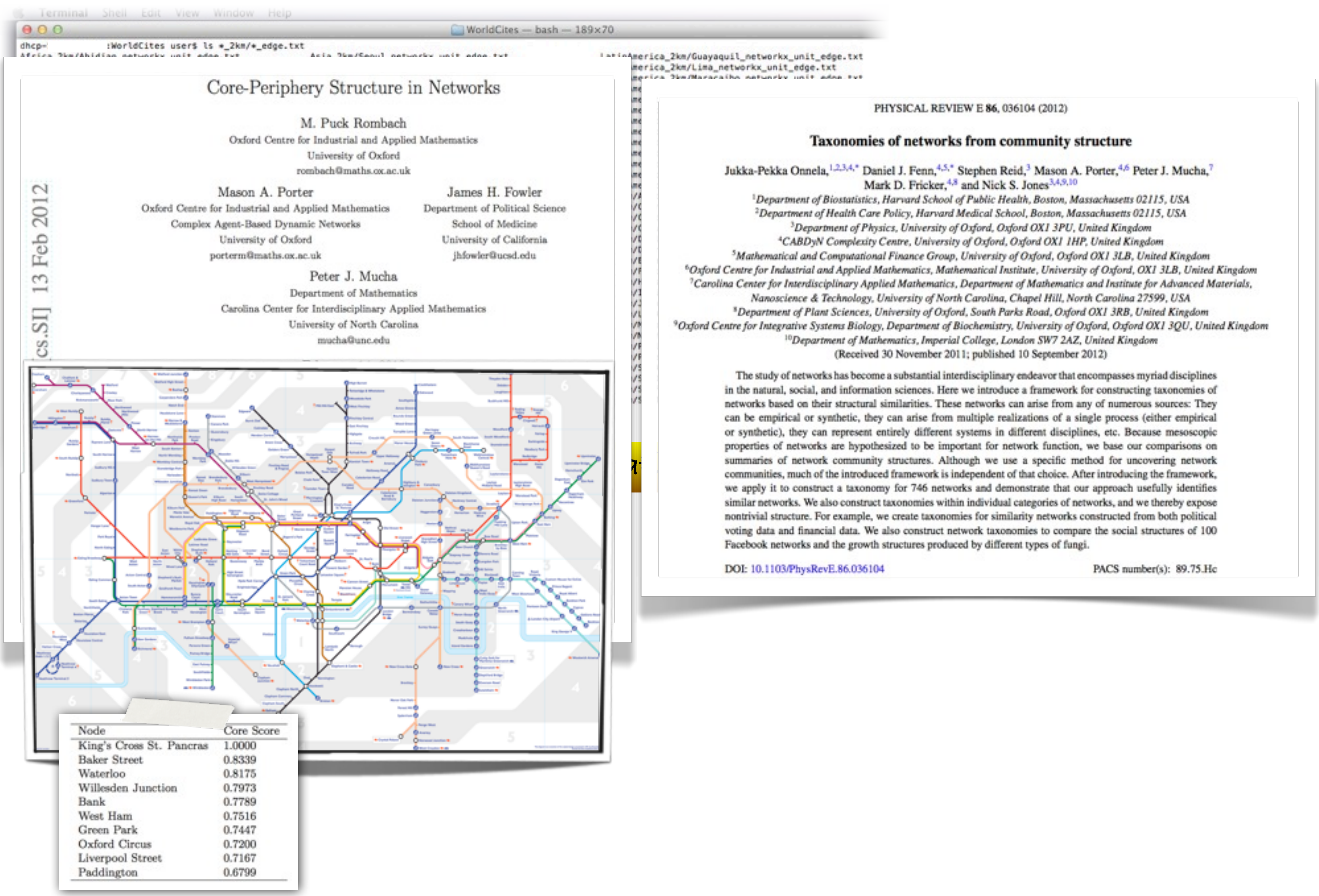
[1] *Department of Biostatistics, Harvard School of Public Health, Boston, Massachusetts 02115, USA*
[2] *Department of Health Care Policy, Harvard Medical School, Boston, Massachusetts 02115, USA*
[3] *Department of Physics, University of Oxford, Oxford OX1 3PU, United Kingdom*
[4] *CABDyN Complexity Centre, University of Oxford, Oxford OX1 1HP, United Kingdom*
[5] *Mathematical and Computational Finance Group, University of Oxford, Oxford OX1 3LB, United Kingdom*
[6] *Oxford Centre for Industrial and Applied Mathematics, Mathematical Institute, University of Oxford, OX1 3LB, United Kingdom*
[7] *Carolina Center for Interdisciplinary Applied Mathematics, Department of Mathematics and Institute for Advanced Materials, Nanoscience & Technology, University of North Carolina, Chapel Hill, North Carolina 27599, USA*
[8] *Department of Plant Sciences, University of Oxford, South Parks Road, Oxford OX1 3RB, United Kingdom*
[9] *Oxford Centre for Integrative Systems Biology, Department of Biochemistry, University of Oxford, Oxford OX1 3QU, United Kingdom*
[10] *Department of Mathematics, Imperial College, London SW7 2AZ, United Kingdom*

(Received 30 November 2011; published 10 September 2012)

The study of networks has become a substantial interdisciplinary endeavor that encompasses myriad disciplines in the natural, social, and information sciences. Here we introduce a framework for constructing taxonomies of networks based on their structural similarities. These networks can arise from any of numerous sources: They can be empirical or synthetic, they can arise from multiple realizations of a single process (either empirical or synthetic), they can represent entirely different systems in different disciplines, etc. Because mesoscopic properties of networks are hypothesized to be important for network function, we base our comparisons on summaries of network community structures. Although we use a specific method for uncovering network communities, much of the introduced framework is independent of that choice. After introducing the framework, we apply it to construct a taxonomy for 746 networks and demonstrate that our approach usefully identifies similar networks. We also construct taxonomies within individual categories of networks, and we thereby expose nontrivial structure. For example, we create taxonomies for similarity networks constructed from both political voting data and financial data. We also construct network taxonomies to compare the social structures of 100 Facebook networks and the growth structures produced by different types of fungi.

DOI: 10.1103/PhysRevE.86.036104 PACS number(s): 89.75.Hc

Now, I’m here in Oxford, with this road network data from 100 cities . . .



Core-Periphery Structure in Networks

M. Puck Rombach
Oxford Centre for Industrial and Applied Mathematics
University of Oxford
rombach@maths.ox.ac.uk

Mason A. Porter
Oxford Centre for Industrial and Applied Mathematics
Complex Agent-Based Dynamic Networks
University of Oxford
porterm@maths.ox.ac.uk

James H. Fowler
Department of Political Science
School of Medicine
University of California
jhfowler@ucsd.edu

Peter J. Mucha
Department of Mathematics
Carolina Center for Interdisciplinary Applied Mathematics
University of North Carolina
mucha@unc.edu

cs.SI] 13 Feb 2012

Node	Core Score
King’s Cross St. Pancras	1.0000
Baker Street	0.8339
Waterloo	0.8175
Willesden Junction	0.7973
Bank	0.7789
West Ham	0.7516
Green Park	0.7447
Oxford Circus	0.7200
Liverpool Street	0.7167
Paddington	0.6799

PHYSICAL REVIEW E **86**, 036104 (2012)

Taxonomies of networks from community structure

Jukka-Pekka Onnela,[1,2,3,4,*] Daniel J. Fenn,[4,5,*] Stephen Reid,[3] Mason A. Porter,[4,6] Peter J. Mucha,[7] Mark D. Fricker,[4,8] and Nick S. Jones[3,4,9,10]

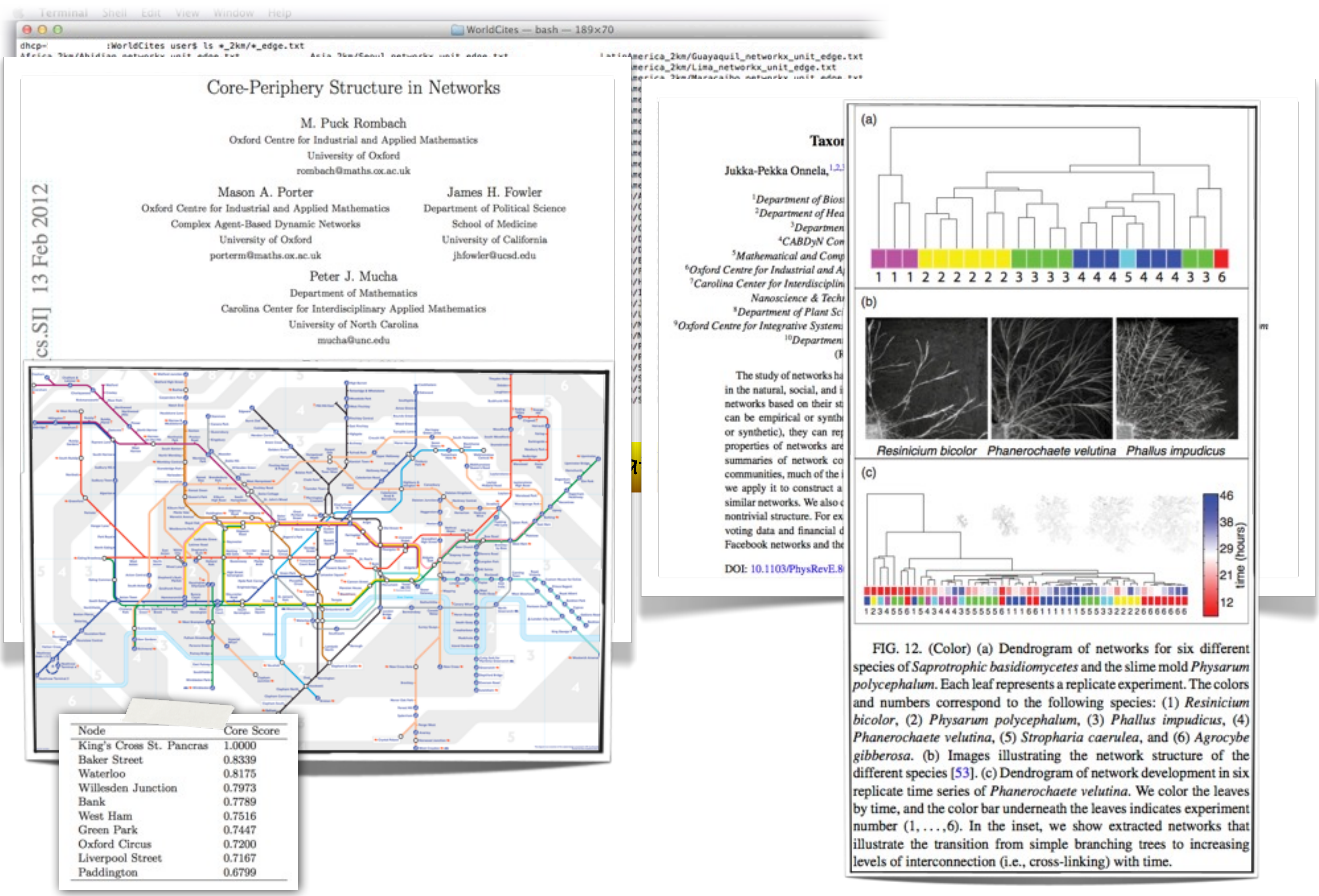
[1]*Department of Biostatistics, Harvard School of Public Health, Boston, Massachusetts 02115, USA*
[2]*Department of Health Care Policy, Harvard Medical School, Boston, Massachusetts 02115, USA*
[3]*Department of Physics, University of Oxford, Oxford OX1 3PU, United Kingdom*
[4]*CABDyN Complexity Centre, University of Oxford, Oxford OX1 1HP, United Kingdom*
[5]*Mathematical and Computational Finance Group, University of Oxford, Oxford OX1 3LB, United Kingdom*
[6]*Oxford Centre for Industrial and Applied Mathematics, Mathematical Institute, University of Oxford, OX1 3LB, United Kingdom*
[7]*Carolina Center for Interdisciplinary Applied Mathematics, Department of Mathematics and Institute for Advanced Materials, Nanoscience & Technology, University of North Carolina, Chapel Hill, North Carolina 27599, USA*
[8]*Department of Plant Sciences, University of Oxford, South Parks Road, Oxford OX1 3RB, United Kingdom*
[9]*Oxford Centre for Integrative Systems Biology, Department of Biochemistry, University of Oxford, Oxford OX1 3QU, United Kingdom*
[10]*Department of Mathematics, Imperial College, London SW7 2AZ, United Kingdom*

(Received 30 November 2011; published 10 September 2012)

The study of networks has become a substantial interdisciplinary endeavor that encompasses myriad disciplines in the natural, social, and information sciences. Here we introduce a framework for constructing taxonomies of networks based on their structural similarities. These networks can arise from any of numerous sources: They can be empirical or synthetic, they can arise from multiple realizations of a single process (either empirical or synthetic), they can represent entirely different systems in different disciplines, etc. Because mesoscopic properties of networks are hypothesized to be important for network function, we base our comparisons on summaries of network community structures. Although we use a specific method for uncovering network communities, much of the introduced framework is independent of that choice. After introducing the framework, we apply it to construct a taxonomy for 746 networks and demonstrate that our approach usefully identifies similar networks. We also construct taxonomies within individual categories of networks, and we thereby expose nontrivial structure. For example, we create taxonomies for similarity networks constructed from both political voting data and financial data. We also construct network taxonomies to compare the social structures of 100 Facebook networks and the growth structures produced by different types of fungi.

DOI: 10.1103/PhysRevE.86.036104   PACS number(s): 89.75.Hc

Now, I'm here in Oxford, with this road network data from 100 cities . . .



Core-periphery structure of networks

- **structural** core-periphery
  - M. P. Rombach *et al*., arXiv:1202.2684 (and references therein).
  - based on the structural definition: “core nodes tend to be connected to core nodes, and peripheral nodes also tend to be connected to core nodes.”

(a) (b) (c) (d)

Figure 1: Different network block models. (a) Community structure, (b) core-periphery structure, (c) global core-periphery structure with local community structure, and (d) global community structure with local core-periphery structure. Note that (c) and (d) are equivalent.

- core-periphery in terms of **transportation**
  - M. Cucuringu *et al*., in preparation.
  - backup-pathway-based definition:

Objective B1: Develop a novel computationally efficient core-periphery detection algorithm. The approach we propose to investigate in this direction is reminiscent of the algorithm for computing a measure of betweenness centrality in networks based on random walks [46]. Our approach aims at developing a scoring method for nodes, based on computing shortest paths in a graph, which reflects the likelihood of that node being in the core, and hence the name of PATH-SCORE (P-SCORE). In what follows, we restrict our attention to undirected unweighted graphs, although we have experimented our approach on weighted graphs and plan on considering directed networks. For each edge (i, j) of a graph G, we compute the shortest path in G between nodes i and j, with edge (i, j) temporarily removed, and all nodes on this shortest path increase their path-score value by $+1$. After repeating this procedure for all edges in G, each node will have a P-Score that reflects the likelihood of that node being in the core. The intuition behind our algorithm is that nodes that are in the core will be on many shortest path in the graph, while nodes in the core will rarely be so.

Core-periphery structure of networks

- **structural** core-periphery
  - M. P. Rombach *et al.*, arXiv:1202.2684 (and references therein).
  - based on the structural definition: "core nodes tend to be connected to core nodes, and peripheral nodes also tend to be connected to core nodes."

Figure 1: Different network block models. (a) Community structure, (b) core-periphery structure, (c) global core-periphery structure with local community structure, and (d) global community structure with local core-periphery structure. Note that (c) and (d) are equivalent.

more suitable for road networks!

- core-periphery in terms of **transportation**
  - M. Cucuringu *et al.*, in preparation.
  - backup-pathway-based definition:

Objective B1: Develop a novel computationally efficient core-periphery detection algorithm. The approach we propose to investigate in this direction is reminiscent of the algorithm for computing a measure of betweenness centrality in networks based on random walks [46]. Our approach aims at developing a scoring method for nodes, based on computing shortest paths in a graph, which reflects the likelihood of that node being in the core, and hence the name of PATH-SCORE (P-SCORE). In what follows, we restrict our attention to undirected unweighted graphs, although we have experimented our approach on weighted graphs and plan on considering directed networks. For each edge (i, j) of a graph G, we compute the shortest path in G between nodes i and j, with edge (i, j) temporarily removed, and all nodes on this shortest path increase their path-score value by $+1$. After repeating this procedure for all edges in G, each node will have a P-Score that reflects the likelihood of that node being in the core. The intuition behind our algorithm is that nodes that are in the core will be on many shortest path in the graph, while nodes in the core will rarely be so.

Core-periphery structure of networks

- **structural** core-periphery
  - M. P. Rombach *et al.*, arXiv:1202.2684 (and references therein).
  - based on the structural definition: "core nodes tend to be connected to core nodes, and peripheral nodes also tend to be connected to core nodes."

Figure 1: Different network block models. (a) Community structure, (b) core-periphery structure, (c) global core-periphery structure with local community structure, and (d) global community structure with local core-periphery structure. Note that (c) and (d) are equivalent.

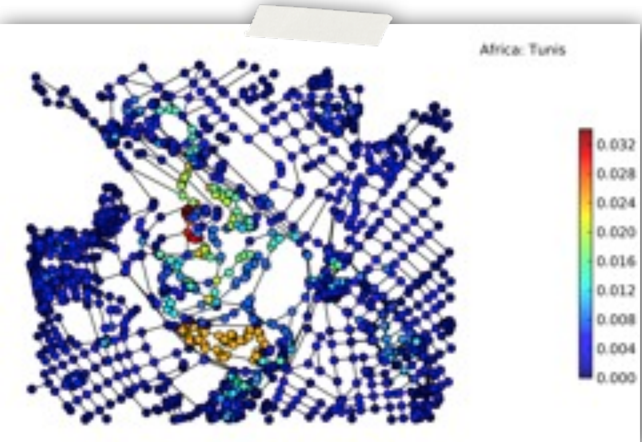
more suitable for road networks!

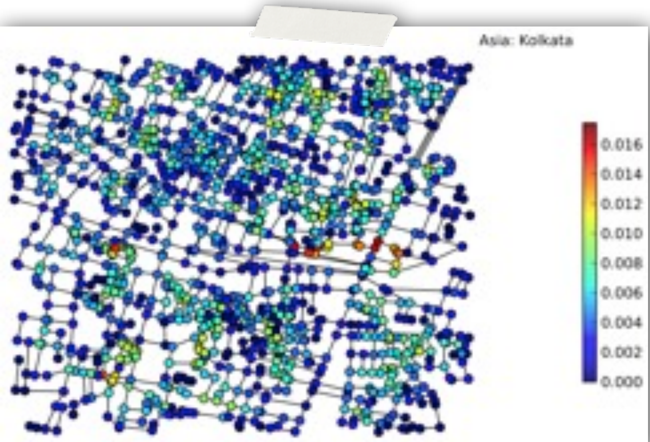
- core-periphery in terms of **transportation**
  - M. Cucuringu *et al.*, in preparation.
  - backup-pathway-based definition:

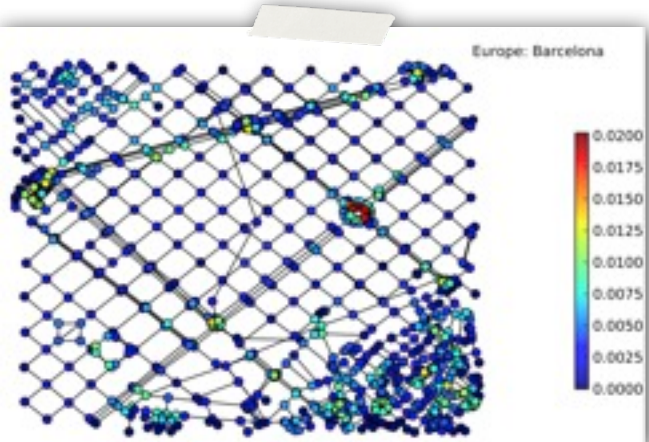
could be edges

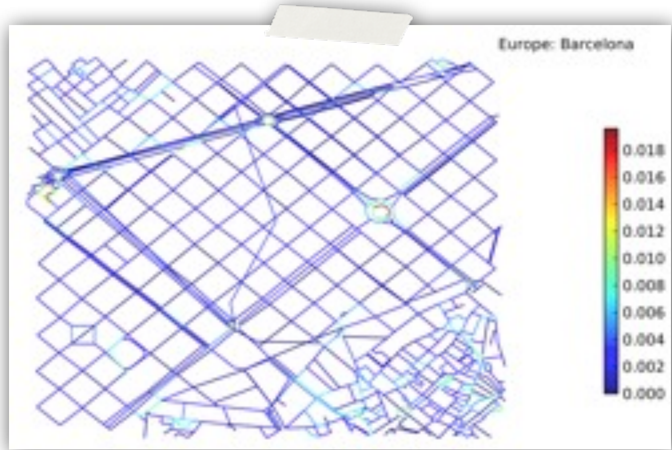
Objective B1: Develop a novel computationally efficient core-periphery detection algorithm. The approach we propose could be "optimal path" considering the weighted edges thm for computing a measure of betweenness centrality in networks based on random walks [46]. Our approach aims at developing a scoring method for nodes, based on computing shortest paths in a graph, which reflects the likelihood of that node being in the core, and hence the name of PATH-SCORE (P-SCORE). In what follows, we restrict our attention to undirected unweighted graphs, although we have experimented our approach on weighted graphs and plan on considering directed networks. For each edge (i, j) of a graph G, we compute the shortest path in G between nodes i and j, with edge (i, j) temporarily removed, and all nodes on this shortest path increase their path-score value by $+1$. After repeating this procedure for all edges in G, each node will have a P-Score that reflects the likelihood of that node being in the core. The intuition behind our algorithm is that nodes that are in the core will be on many shortest path in the graph, while nodes in the core will rarely be so.

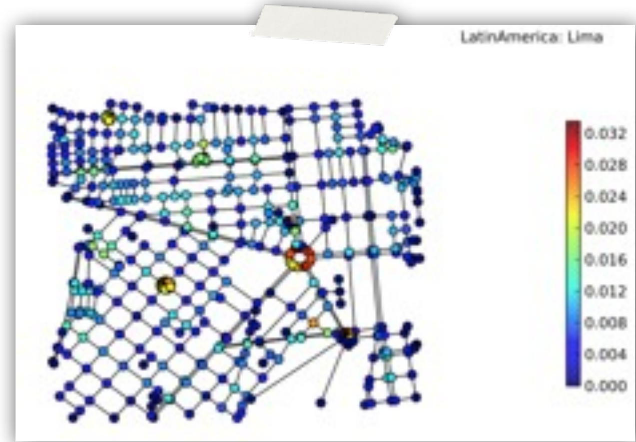
P-SCORE for nodes and edges, considering the shortest path minimizing the sum of Euclidean distances

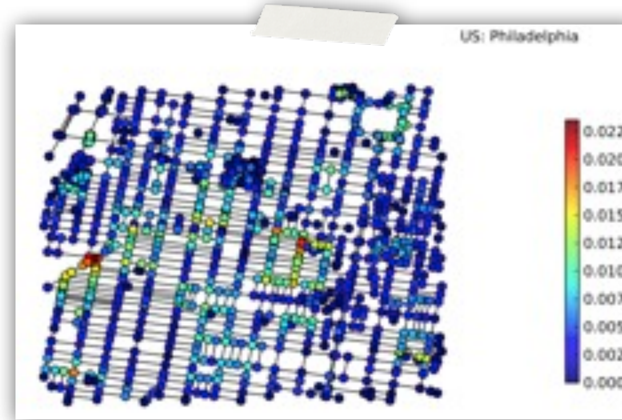


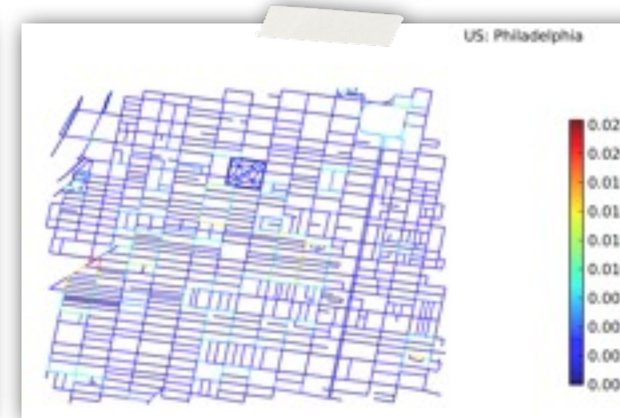




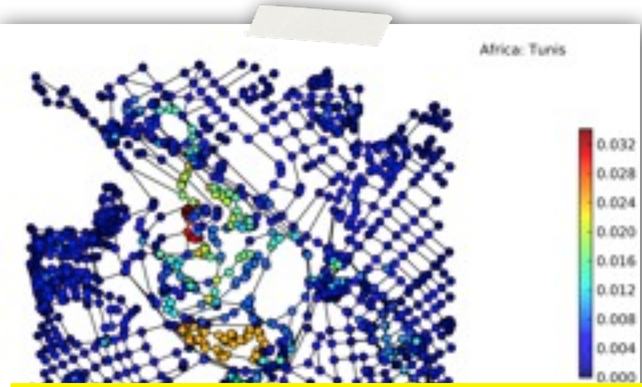




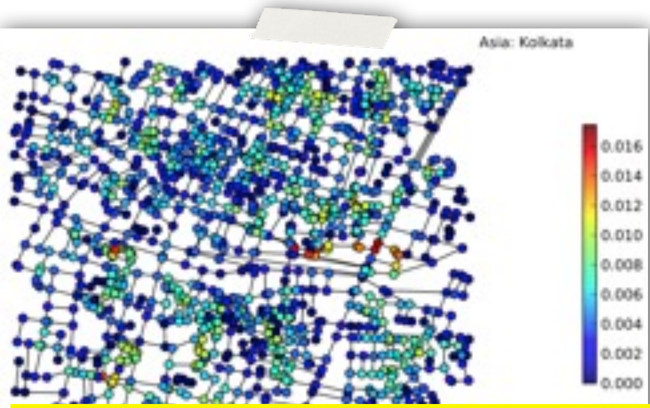




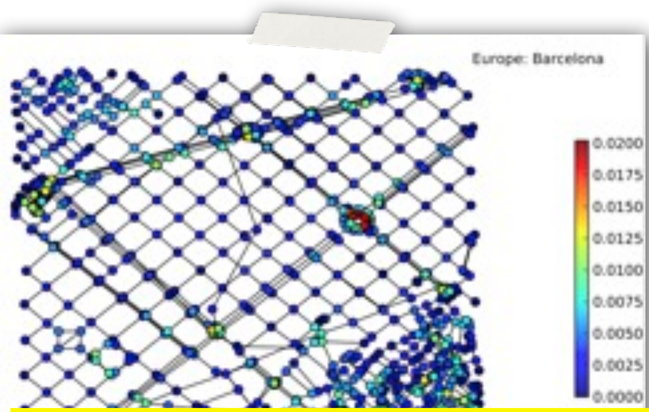
P-SCORE for nodes and edges, considering the shortest path minimizing the sum of Euclidean distances



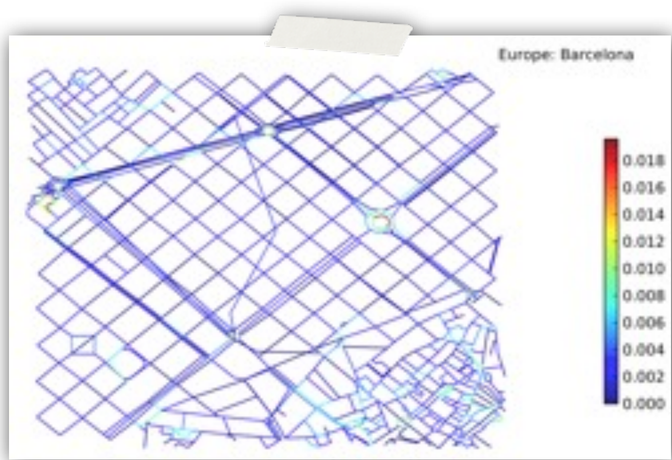
correlation btw P-SCORE and betweenness
= 0.168 (Pearson), 0.374 (Spearman)

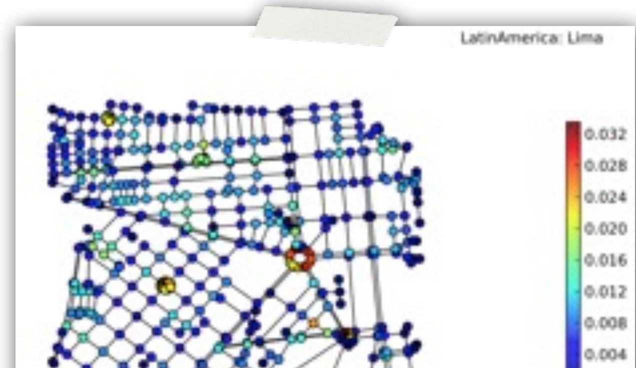


correlation btw P-SCORE and betweenness
= 0.278 (Pearson), 0.478 (Spearman)

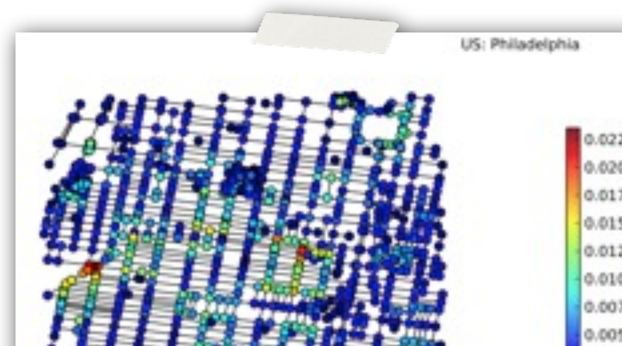


correlation btw P-SCORE and betweenness
= 0.139 (Pearson), 0.322 (Spearman)

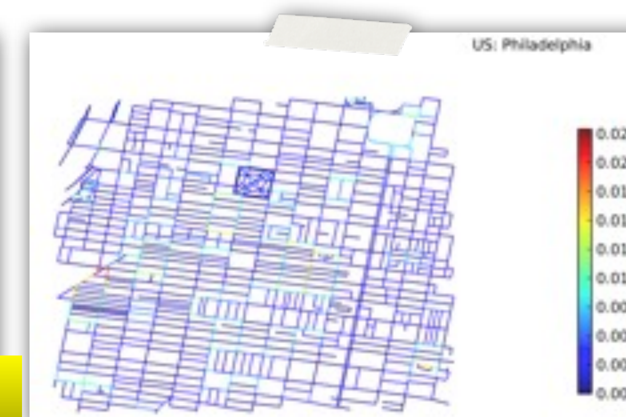




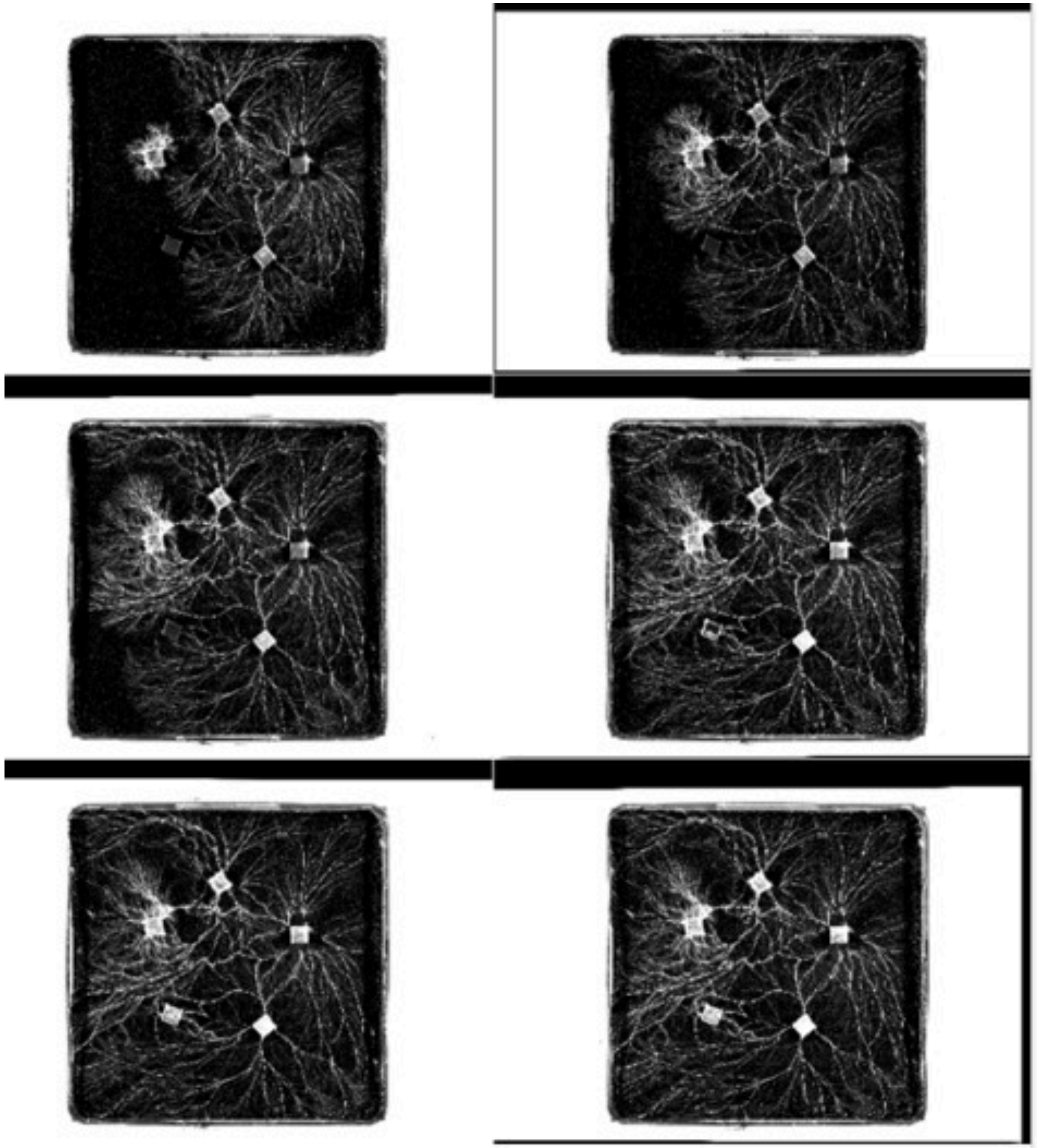
correlation btw P-SCORE and betweenness
= 0.247 (Pearson), 0.435 (Spearman)



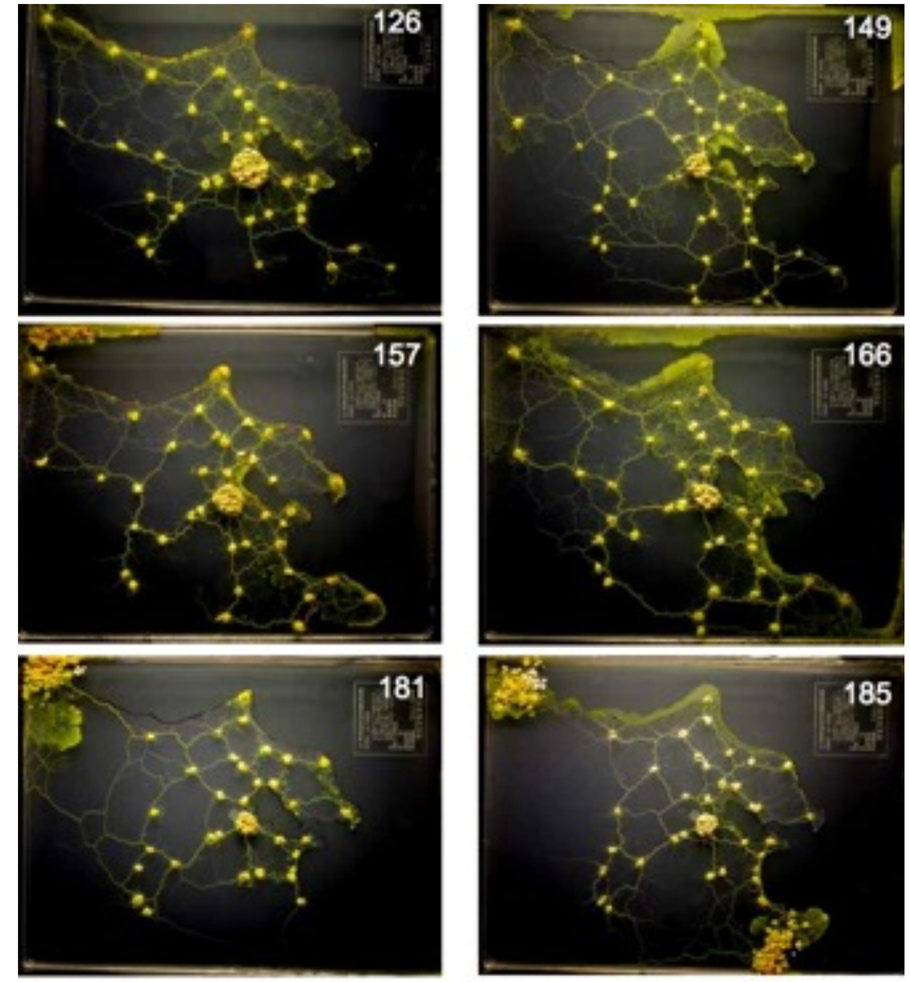
correlation btw P-SCORE and betweenness
= 0.103 (Pearson), 0.329 (Spearman)

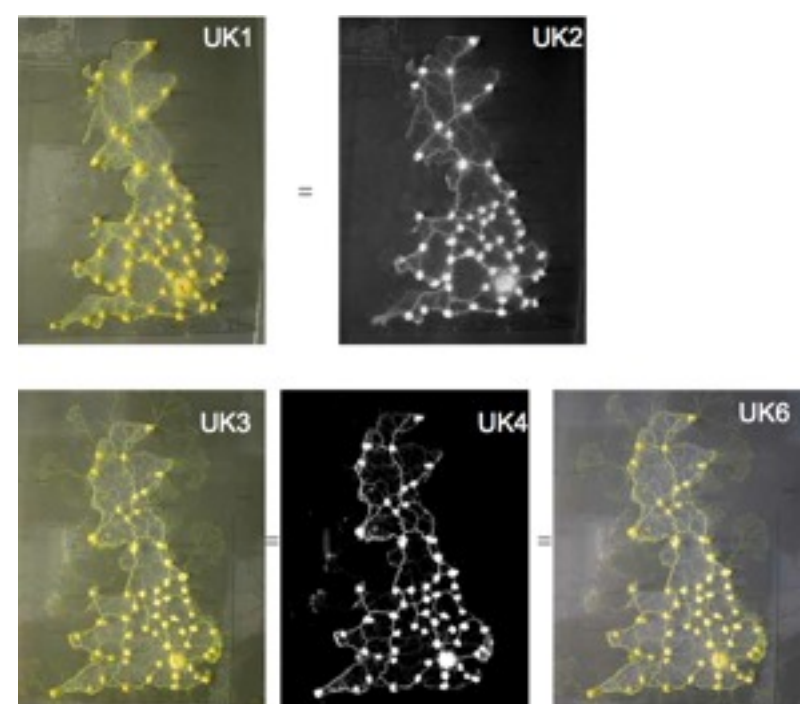


Fungal network data: another transportation networks! (provided by Mark Fricker & Dan Fenn)



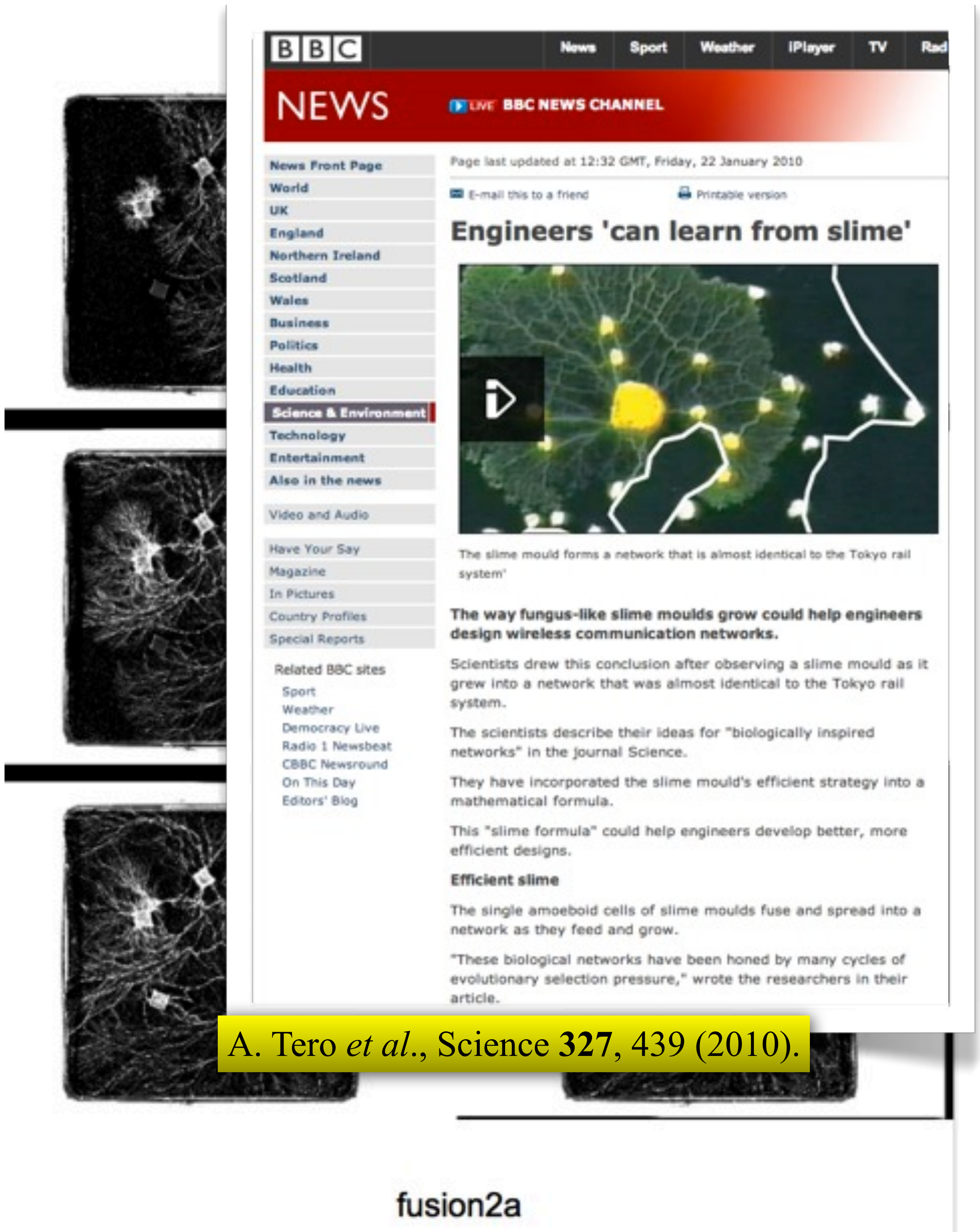
fusion2a





ref) L. Heaton *et al.*, Phys. Rev. E **86**, 021905 (2012); Proc. R. Soc. B **277**, 3265 (2012); Fungal Biology Reviews **26**, 12 (2012).

Fungal network data: another transportation networks! (provided by Mark Fricker & Dan Fenn)



BBC NEWS

News | Sport | Weather | iPlayer | TV | Rad

LIVE BBC NEWS CHANNEL

News Front Page
World
UK
England
Northern Ireland
Scotland
Wales
Business
Politics
Health
Education
Science & Environment
Technology
Entertainment
Also in the news

Video and Audio

Have Your Say
Magazine
In Pictures
Country Profiles
Special Reports

Related BBC sites
Sport
Weather
Democracy Live
Radio 1 Newsbeat
CBBC Newsround
On This Day
Editors' Blog

Page last updated at 12:32 GMT, Friday, 22 January 2010

E-mail this to a friend | Printable version

Engineers 'can learn from slime'

The slime mould forms a network that is almost identical to the Tokyo rail system'

The way fungus-like slime moulds grow could help engineers design wireless communication networks.

Scientists drew this conclusion after observing a slime mould as it grew into a network that was almost identical to the Tokyo rail system.

The scientists describe their ideas for "biologically inspired networks" in the journal Science.

They have incorporated the slime mould's efficient strategy into a mathematical formula.

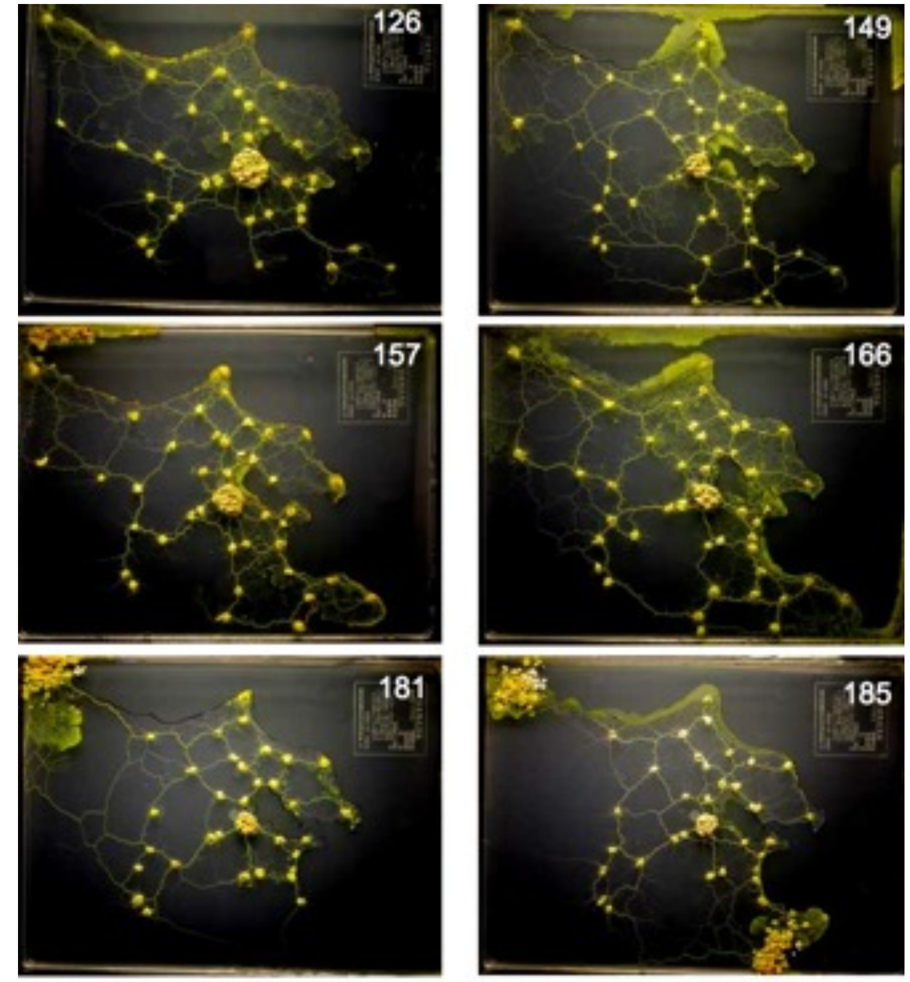
This "slime formula" could help engineers develop better, more efficient designs.

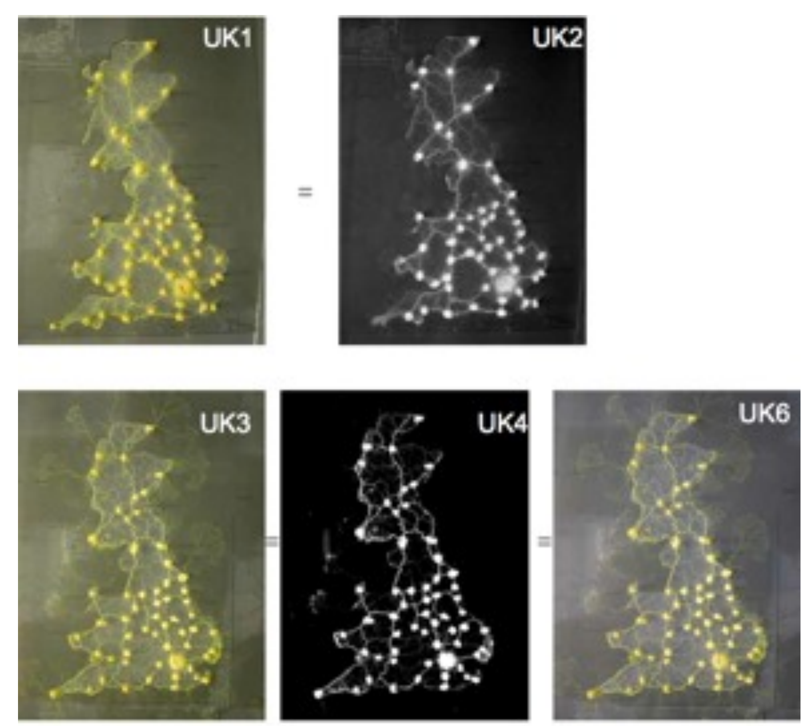
Efficient slime

The single amoeboid cells of slime moulds fuse and spread into a network as they feed and grow.

"These biological networks have been honed by many cycles of evolutionary selection pressure," wrote the researchers in their article.

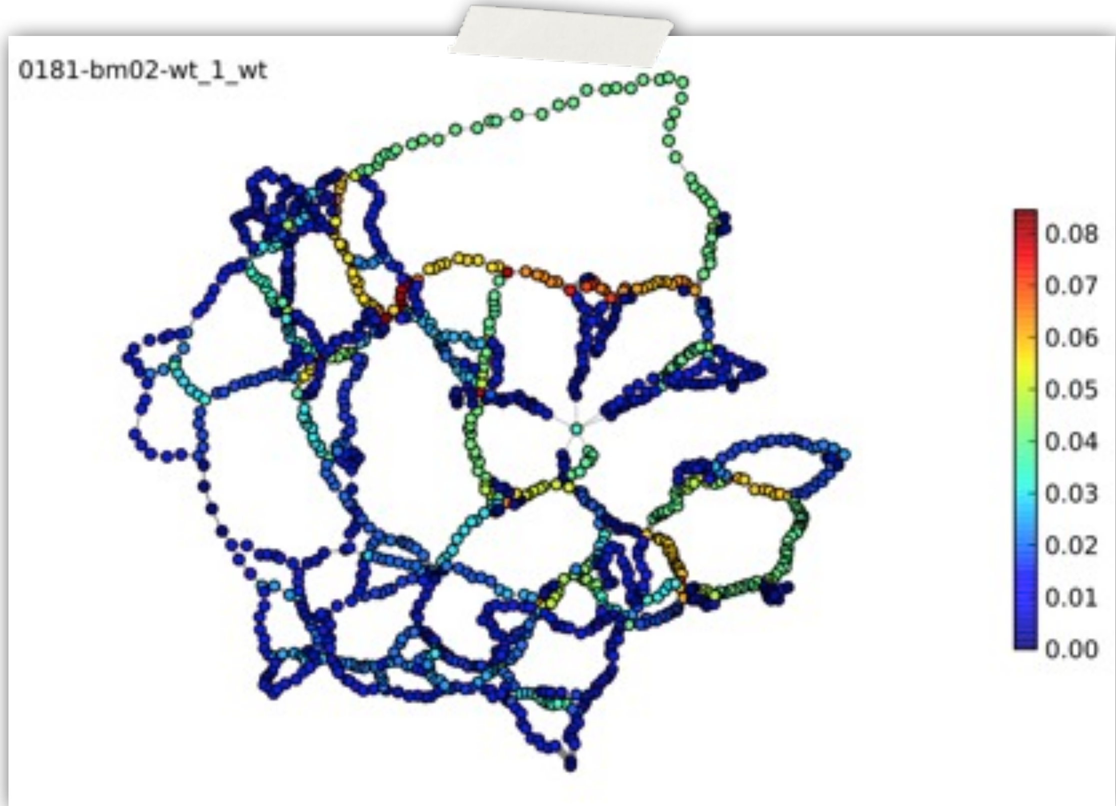
A. Tero *et al.*, Science **327**, 439 (2010).

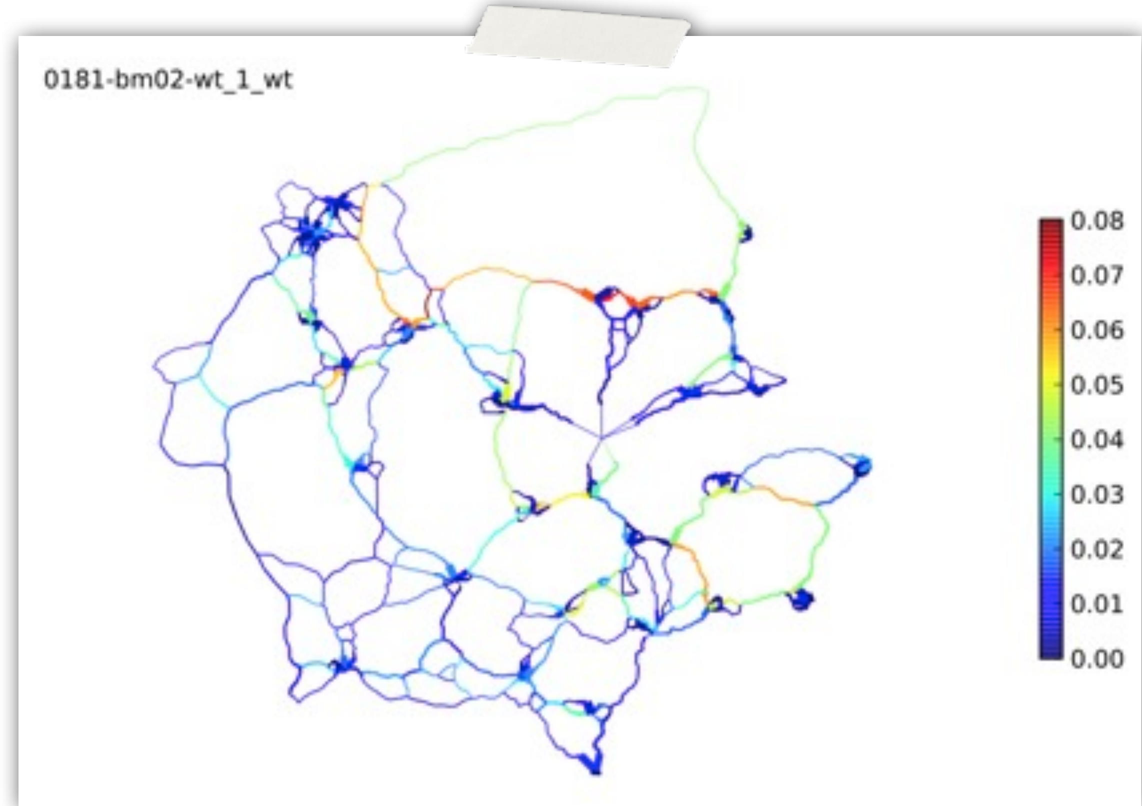


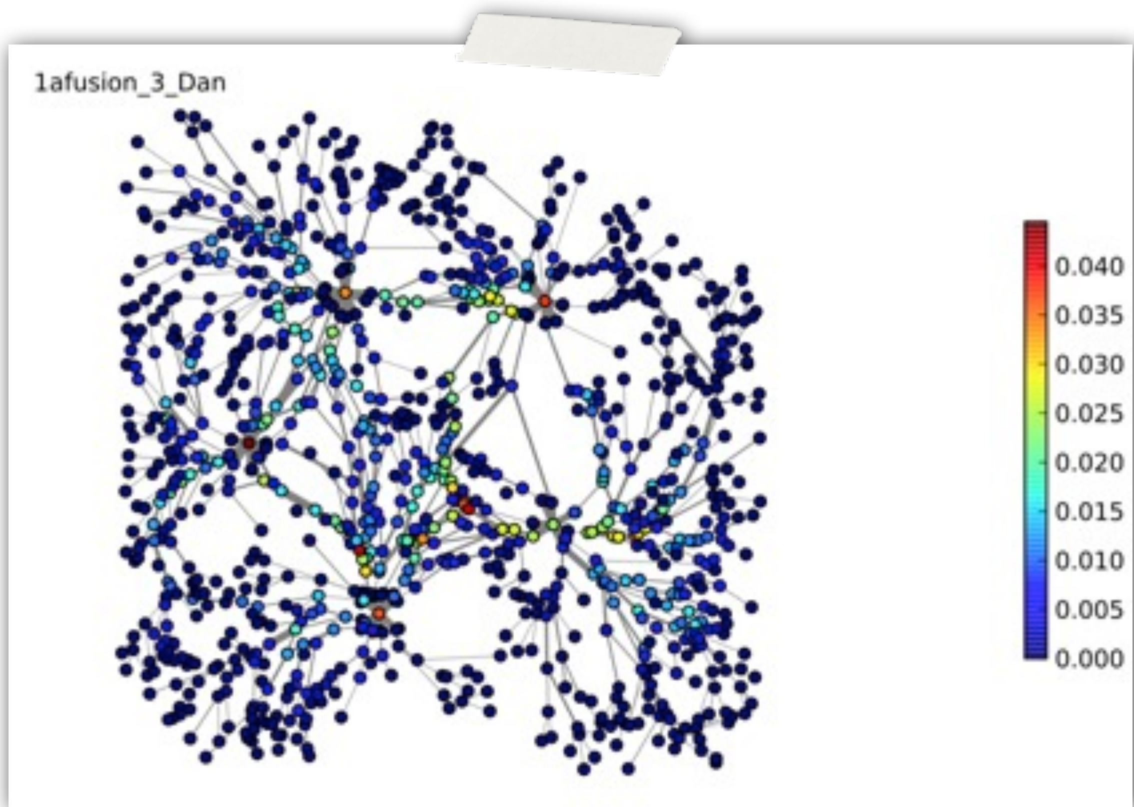


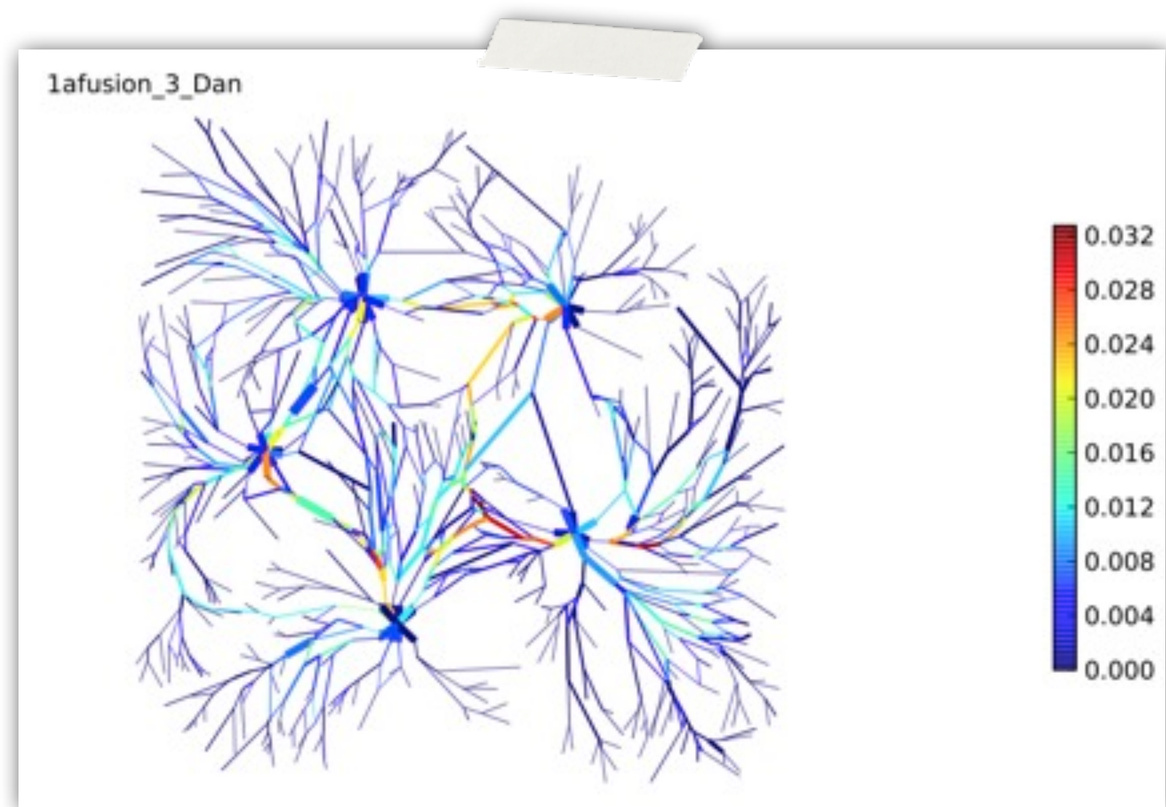
ref) L. Heaton *et al.*, Phys. Rev. E **86**, 021905 (2012); Proc. R. Soc. B **277**, 3265 (2012); Fungal Biology Reviews **26**, 12 (2012).

P-SCORE for nodes and edges, considering the optimal path maximizing the sum of “conductance”



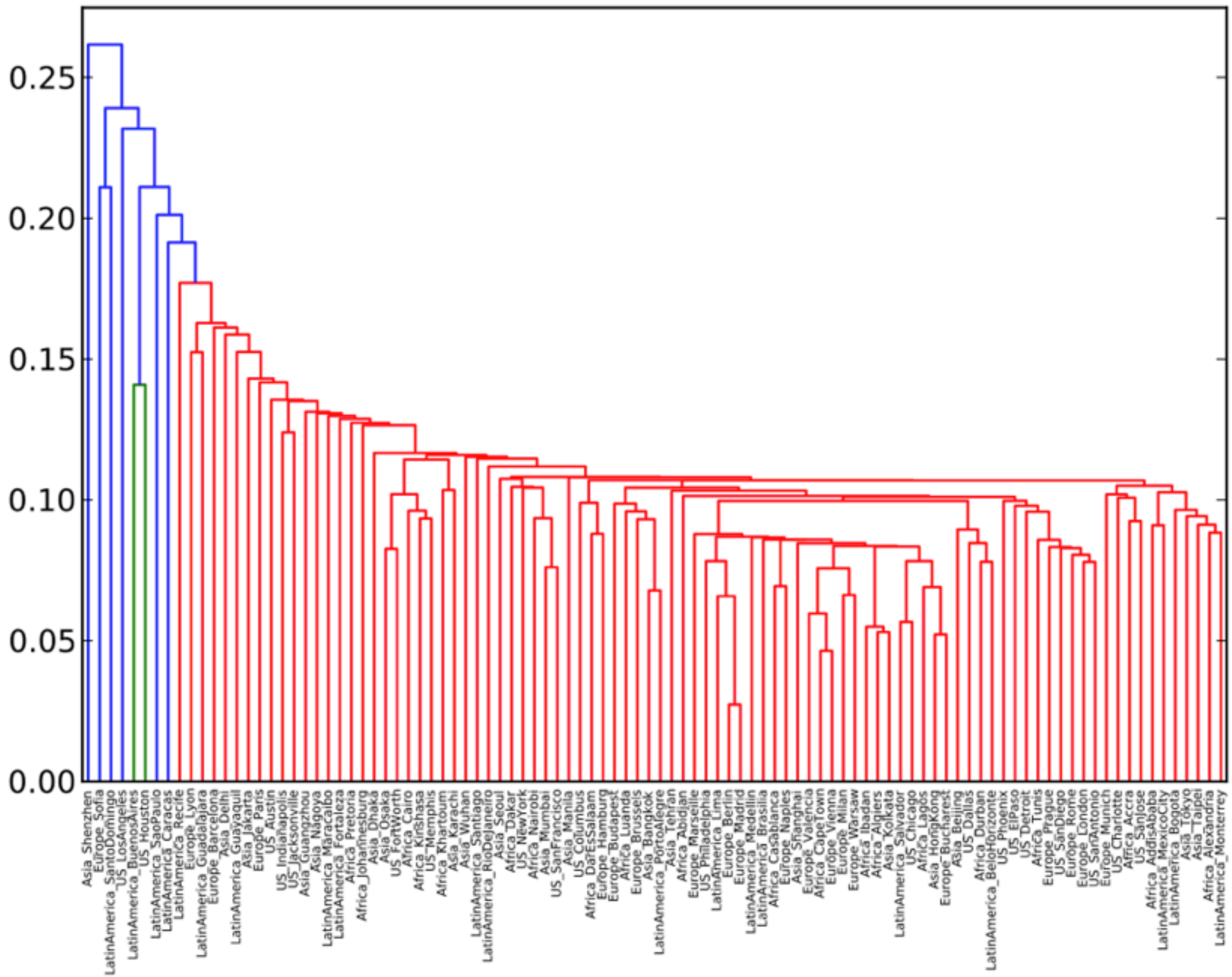






and more . . . (518 networks in total)

Dendrogram of road networks, based on the mesoscopic response function (MRF) analysis



ref) J.-P. Onnela, D. J. Fenn, S. Reid, M. A. Porter, P. J. Mucha, M. D. Fricker, and N. S. Jones, Phys. Rev. E **86**, 036104 (2012).

Summary and Outlook

- **greedy navigation**: a more realistic approach, exploiting local geometric information
    - modified centrality measures
    - “Braess edge” phenomenon due to greediness

- properties of greedy-navigation-friendly network **topology** (shortcut construction) or **geometry** (layout optimization)

- data: **100 road networks, 518 fungal networks**, etc. (any suggestion or donation? ;)
    - **core-periphery** structure
    - other mesoscopic properties (e.g., “taxonomy” analysis)
    - . . .

Summary and Outlook

- **greedy navigation**: a more realistic approach, exploiting local geometric information
  - modified centrality measures
  - "Braess edge" phenomenon due to greediness

- properties of greedy-navigation-friendly network **topology** (shortcut construction) or **geometry** (layout optimization)

- data: **100 road networks, 518 fungal networks**, etc. (any suggestion or donation? ;)
  - **core-periphery** structure
  - other mesoscopic properties (e.g., "taxonomy" analysis)
  - . . .

Thank you for your attention! =)