Complex networks

Diffusion and spreading on networks, resilience

János Török

Department of Theoretical Physics

May 3, 2023

Spreading on networks

- One of the most important problems on networks
- Also one of the real success
- This lecture:
  - Advanced mean-field calculations
  - Cascade models
  - Spreading in temporal networks

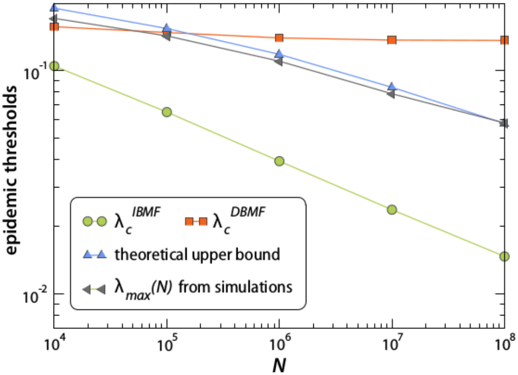
SIS: Comparison

- IBMF

$$\lambda_c^{\mathrm{IBMF}} = \begin{cases} \frac{1}{\sqrt{k_{max}}} & \text{if } \gamma \geq 5/2 \\ \frac{\langle k \rangle}{\langle k^2 \rangle} & \text{if } 2 < \gamma < 5/2 \end{cases}$$

- DBMF

$$\lambda > \lambda_c^{DBMF} = \frac{\langle k \rangle}{\langle k^2 \rangle}$$



Immunization

- Epidemic threshold (complete graph/fully mixed state):

$$R_0 = \frac{\beta}{\mu} \begin{cases} > 1 & \text{outbreak} \\ = 1 & \text{threshold} \\ < 1 & \text{localized} \end{cases}$$

- The density of the immune vertices is g, then:

$$\beta' = \beta(1-g)$$

- The threshold for networks

$$\frac{\beta(1-g)}{\mu} = \frac{\langle k \rangle}{\langle k^2 \rangle}$$

- For infinitely large scale free network with $\gamma \leq 3$ we get $g_c = 1$
- For random immunization everybody must be vaccinated

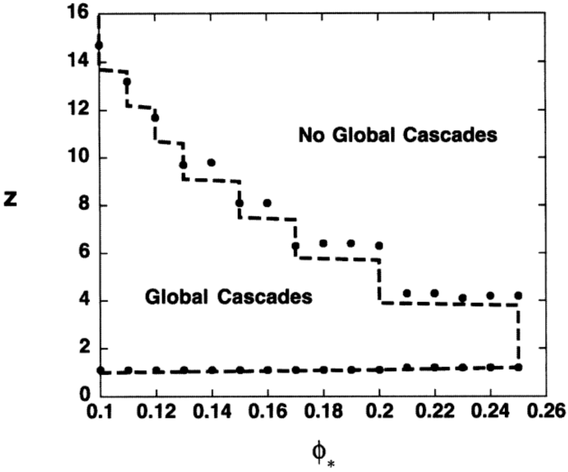
Threshold model

- Networks with average degree $\langle k \rangle = z$
- Nodes have threshold ϕ_i
- If the number of active nodes in the neighborhood reach ϕ_i then the node becomes active (too many friends have some product I will also buy it)
- Start from a small seed
- If thresholds are sufficiently low cascades may propagate through the whole system (size $\sim \mathcal{O}(N)$)

Watts, A simple model of global cascades on random networks (2002)

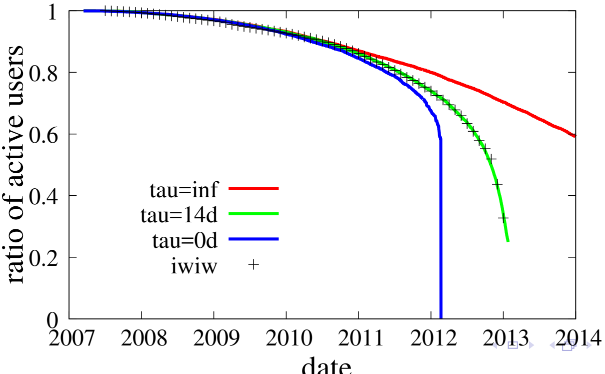
Threshold model: Phase diagram

- Top line: first order phase transition of cascades
- Bottom line: second order phase transition of network percolation limit



Fall of a social network site: Model

- Users leave due to exogenous effects (advertisements, news, etc.):
  - Here rate of leave increases with time as was the popularity of the alternative site
  - Users with low degree are more susceptible to global effects
- Users leave if their friends leave.
  - Threshold model with threshold above 45%
  - Leave is not immediate one needs time (τ) to recognize friend is inactive

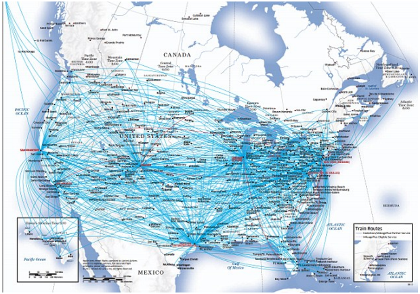


Percolation on networks (graphs)

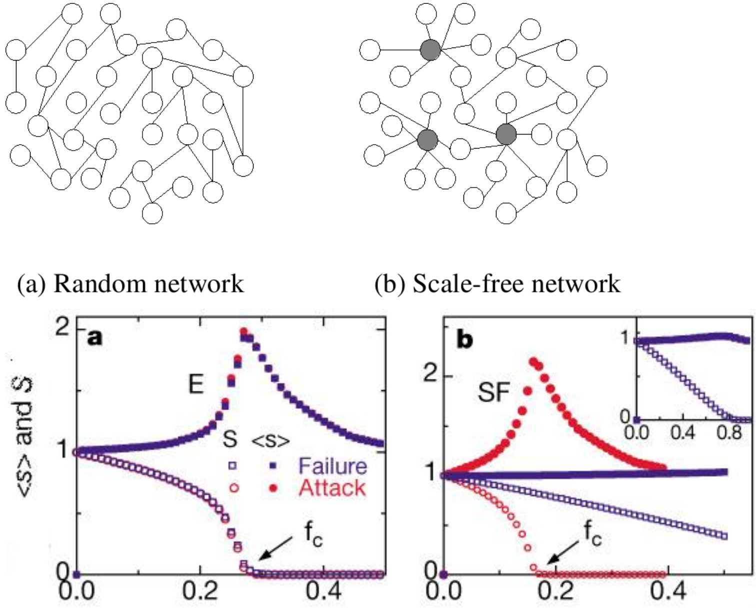
- Network is defined by nodes and links
- Percolation gives us connected components
- Link removal percolation gives information about robustness, and structure

Percolation and attack on random networks

- Failure: equivalent to percolation: remove nodes at random
- Attack: remove most connected nodes

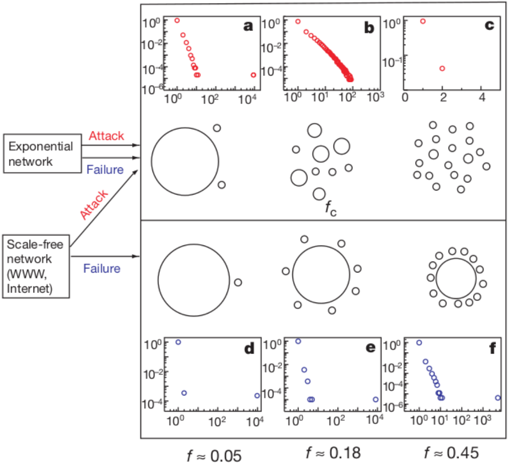


Error vs. attacks



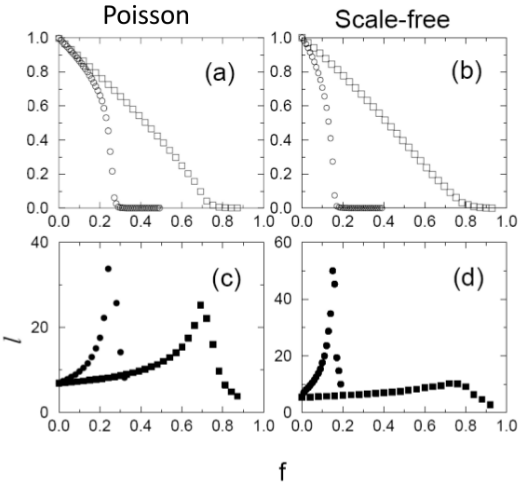
Percolation and attack on random networks

- Failure: equivalent to percolation: remove nodes at random
- Attack: remove most connected nodes



Robustness

- Link/node removal percolation
- Here: random, and largest first
- There is also weakest first



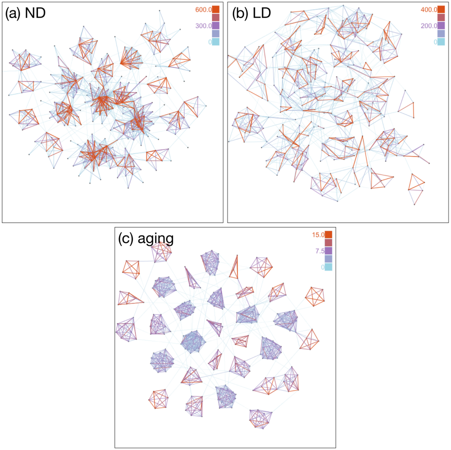
- squares: random failure
- circles: targeted attack

Failures: little effect on the integrity of the network if scale free.
Attacks: fast breakdown

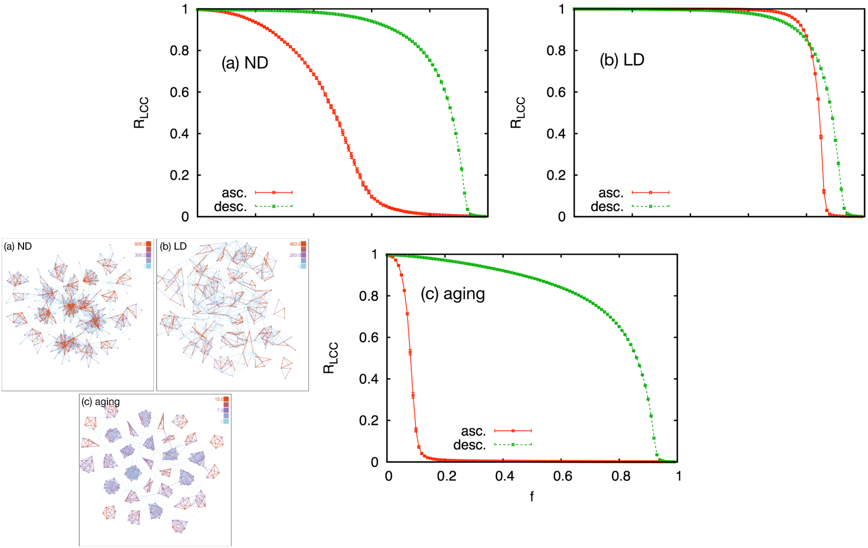
l: average component size

Link removal percolation on networks

- Granovetter hypothesis: The strength of the weak ties
- Human communities have strong connections
- These communities are connected with weak ties
- Test the structures with Link removal percolation

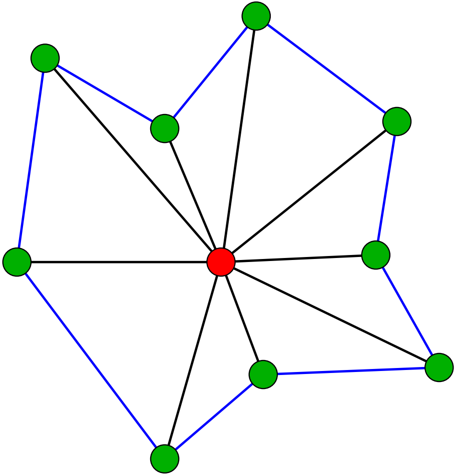


Link removal percolation on networks



Robustness

- Resistant both against random and targeted attacks.
- Must have hubs to resist random attacks
- Small degree nodes should be interconnected so they remain viable after removal of the hubs



Robustness against attacks

- Malicious attacks target central nodes, hubs
- Solution: central nodes should be connected
- Assortative mixing is preferred (high degree nodes are connected between each other)
- (Barabasi-Albert is thus a bad example)
- Robustness measure:

$$R = \frac{1}{N} \sum_{Q=1}^{N} s(Q)$$

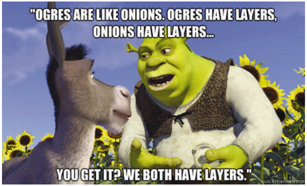
- $s(Q)$ fraction of nodes in the largest connected cluster after removing $Q = qN$ nodes
- Optimize for R

Onion structures

- Robustness measure:

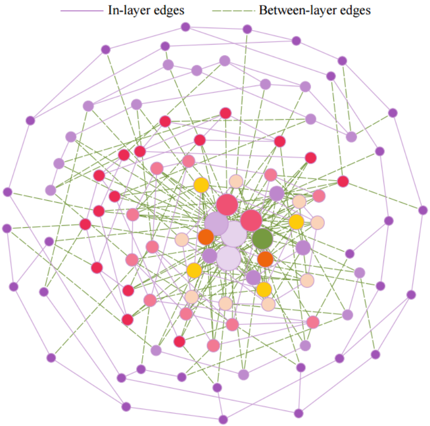
$$R = \frac{1}{N} \sum_{Q=1}^{N} s(Q)$$

- $s(Q)$ fraction of nodes in the largest connected cluster after removing $Q = qN$ nodes
- Optimize R by only rewiring and keeping degree distribution constant
- Onion structures are the most robust
  - Assortative
  - Layers with similar degree nodes
  - Inter-layer connections



Onion structures

- Assortative
- Layers with similar degree nodes
- Inter-layer connections



Château de Vincennes

Flight route optimization

- Suppose weight of a link is defined as

$$w_{ij} = d_{ij}/t_{ij}$$

  where d_{ij} is the distance, and t_{ij} is the traffic between two cities
- When more paths are possible the most economical is used:

$$C_{ij} = \min_{p \in \mathcal{P}} \sum_{l \in p} w_l$$

- Keep total traffic constant
- Function to be optimized is the average cost to pay to travel from any node to any other

$$\mathcal{L} = \frac{2}{N(N-1)} \sum_{i<j} C_{ij}$$

Flight route optimization

- Check a small circle:
- Let us assume $d_1 = d(A, B) = d(B, C) > d(A.C) = d'$
- Cost function (T is the average traffic between two cities):

$$\mathcal{L}_1 = \frac{2d + d'}{T}$$

- Cut connection (B, C). The new cost function

$$\mathcal{L}_2 = \frac{d + d'}{2T} < \mathcal{L}_1$$

- The optimal path is a tree!

Tree model

- If it is known that the network is a tree task is easier:

$$\mathcal{L}_{\sqcup} = \sum_{e \in T} b_e \frac{d_e}{t_e}$$

where b_e is the link betweenness centrality

- The optimal traffic

$$t_e = \frac{T\sqrt{b_e d_e}}{\sum_e \sqrt{b_e d_e}}$$

- The optimal traffic tree can then be obtained by minimizing

$$\mathcal{L} = \sum_{e \in T} \sqrt{b_e d_e}$$

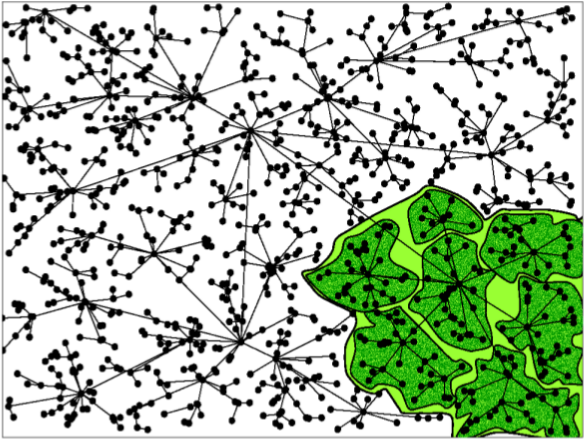
- More generally

$$\mathcal{L} = \sum_{e \in T} b_e^{\mu} d_e^{\nu}$$

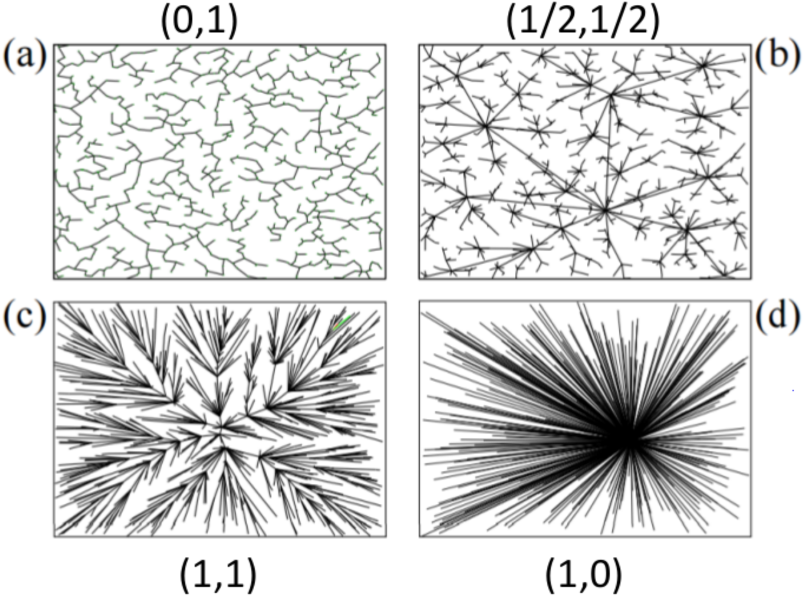
where μ and ν control the relative importance of distance against topology as measured by centrality

Optimal traffic on networks

- Exponential degree distribution
- Power law betweenness distribution
- Hierarchical organizations
- $\mu = \nu = 0.5$

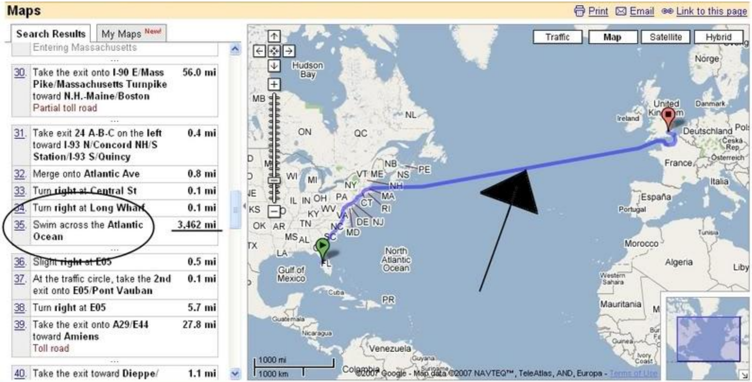


Optimal traffic on networks



Search on graphs

- Shortest path algorithm
  - Many applications: e.g. Route planning
  - Calculation of *betweenness centrality*
  - Global information needed



Dijkstra's algorithm

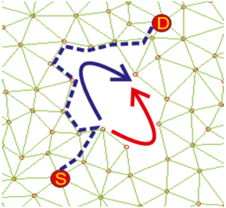
- Find the shortest path from a source
- Known: links, link weights (node distances)
- Store: distance to that point, link to previous element in shortest path
- List of unvisited path sorted by distance to origin (set to infinity if unknown)
- Algorithm:
  1. Choose the unvisited node with the smallest distance to the origin
  2. Visit all its unvisited neighbors: if distance is smaller than the current distance to that point, store it and set link to previous element to the current active node
  3. Mark node as finished
  4. If list of unvisited nodes is not empty, go to 1.

Related problems

- Finding out of a labyrinth
- Search path with local knowledge
  - Very important!
  - Global optimization can be too expensive
  - Global structure may net be known, or varies fast
- Recommender systems
- File sharing

Greedy routing

- Agents have only local information
- They know how far their neighbors are from the target
- They forward the packet to the neighbor with the smallest distance to the target
- May lead to dead end
- Navigability:
  - Fully: The network is navigable if there exists a greedy path between all pairs of nodes
  - Fractional p_s: the fraction of node pairs with greedy route between them



Navigability of scale free networks

- Scale free network (configuration model) $P(k) \sim k^{\gamma}$
- Metric space is needed, here nodes are randomly placed on a ring
- Probability of connection:

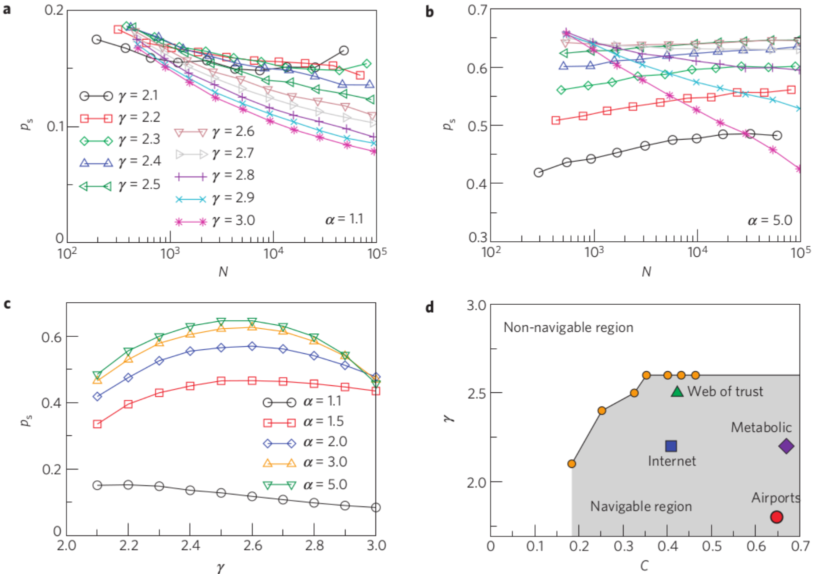
$$r(g; k, k') = \left(1 + \frac{d}{kk'}\right)^{-\alpha}$$

- the probability of link connection between two nodes is decreases with the distance as $\sim d^{-\alpha}$
- Increases with their degrees as $\sim (kk')^{\alpha}$
- Measure: Greedy navigation success rate p_s

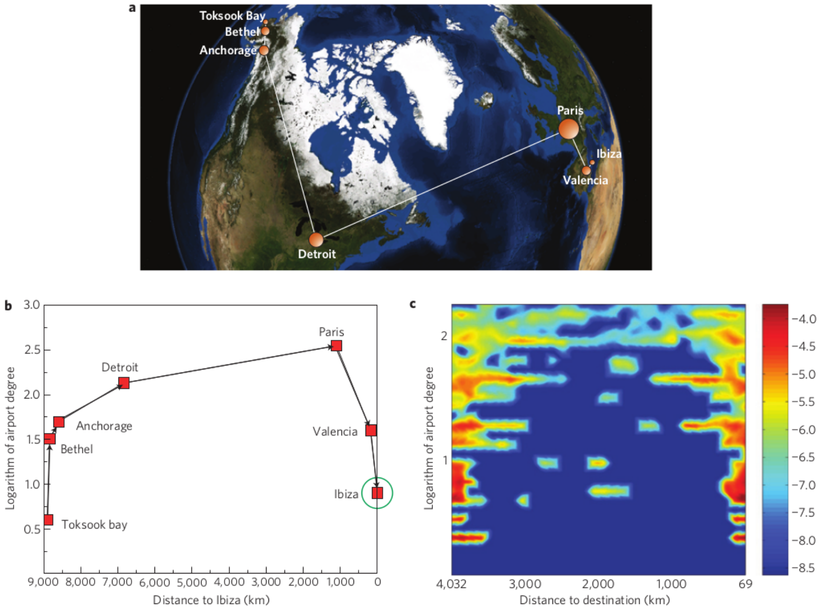
Boguñá et al. Navigability of complex networks, (2009)

Navigability of scale free networks

- Navigable if $p_s(N)$ increases with N
- C is clustering for given α

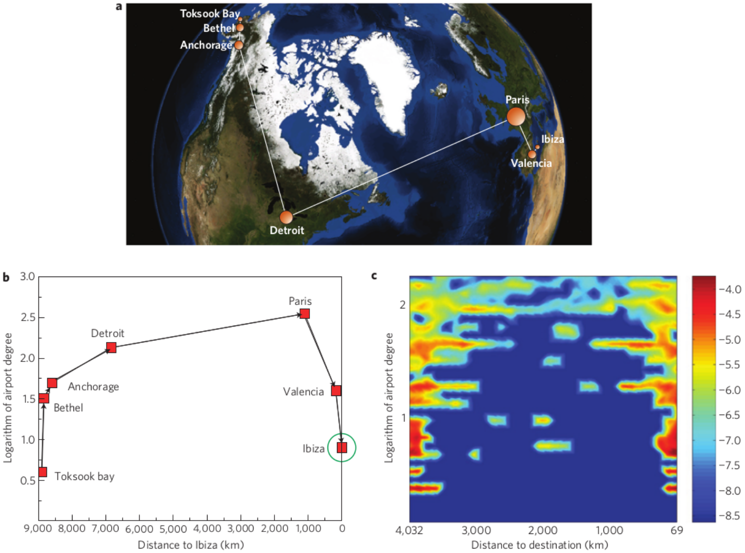


Navigability of scale free networks: Airport example



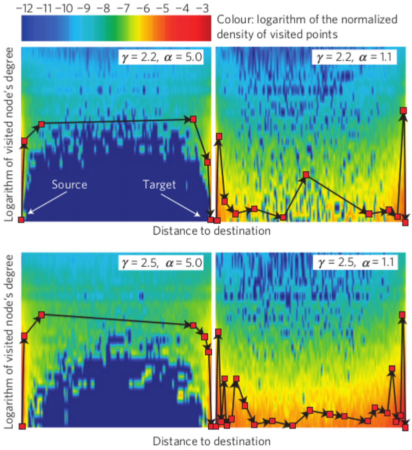
Navigability of scale free networks: Airport example

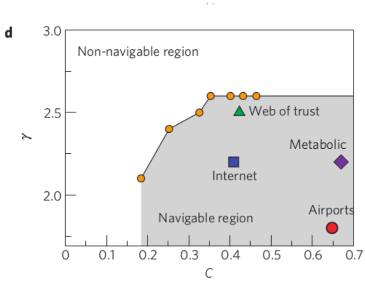
- General greedy routes: generally go though large degree nodes



Navigability of scale free networks: Airport example

- Results of the model
- C increases with α



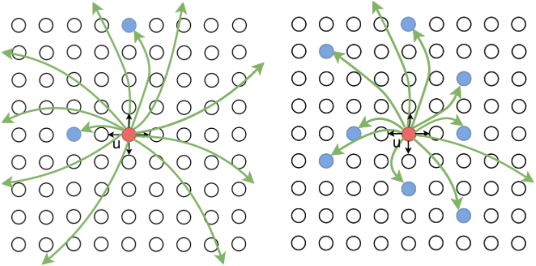


Kleinberg model

- Square lattice, with next nearest neighbor links
- Distance is defined in lattice (Manhattan) distance
- One long range link to a randomly selected node with probability proportional to $r^{-\alpha}$ (here also r is measured in Manhattan distance)
- Expected behaviour

$$T \sim L^x$$

- In small word we expect $x \to 0$ and thus $T \sim \log^y L$



(a) $r << 2$ (b) $r \sim 2$ (c) $r >> 2$

Kleinberg Navigation in a small world, (2000)

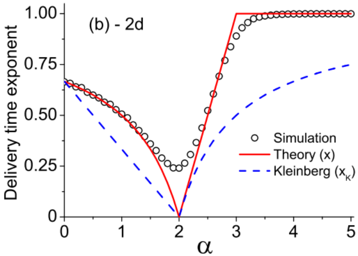
Kleinberg model

- Expected behaviour $T \sim L^x$
- Kleinberg lower bounds (2d):

$$x = \begin{cases} (2-\alpha)/2 & 0 \leq \alpha < 2 \\ (\alpha-2)/(\alpha-1) & \alpha > 2 \end{cases}$$

- Master equation

$$x = \begin{cases} \frac{d-\alpha}{d+1-\alpha} & 0 \leq \alpha < d \\ \alpha - d & d \leq \alpha < d+1 \\ 1 & \alpha > d+1 \end{cases}$$



Search strategies: Network

- What if there is no underlying metric?
- The position of the target is unknown
- Networks
  - Power law degree distribution with exponent between 2 and 3
  - Random weights on the links: smaller weights correspond to shorter paths
  - No global information: each node has information about its neighbors (or second neighbors)
  - structure may change in time

Thadakamalla et al. Search in weighted complex networks (2006)

Search strategies

- Strategies:
  - Random walk
  - (Semi) Self avoiding random walk (do not send the package back to the one from which it was received)
  - Self avoiding random walk, do not send back to nodes where packed already has been. (can lead to dead ends!)
  - Pass through the link with the smallest weight (at least it is not expensive)
  - Choose the best connected neighbor (we saw in the metric version that it is not a bad idea)
  - Choose the neighbor with the smallest average link weight (it is close to many)
  - Choose neighbor with the highest link betweenness centrality (use all available information)

Search strategies: Results

- Random graph:

Search strategy	Beta $\sigma^2 = 2.3$	Uniform $\sigma^2 = 8.3$	Exp. $\sigma^2 = 25$	Power-law $\sigma^2 = 4653.8$
Random walk	1271.91	1284.9	1253.68	1479.32
Minimum edge weight	1017.74	767.405	577.83	562.39
Highest degree	994.64	1014.05	961.5	1182.18
Minimum average node weight	1124.48	954.295	826.325	732.93
Highest LBC	980.65	968.775	900.365	908.48

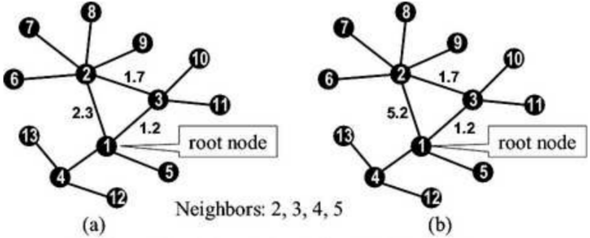
- Scale free network:

Search strategy	Beta $\sigma^2 = 2.3$	Uniform $\sigma^2 = 8.3$	Exp. $\sigma^2 = 25$	Power-law $\sigma^2 = 4653.8$
Random walk	1107.71 (202%)	1097.72 (241%)	1108.70 (272%)	1011.21 (344%)
Minimum edge weight	704.47 (92%)	414.71 (29%)	318.95 (7%)	358.54 (44%)
Highest degree	379.98 (4%)	368.43 (14%)	375.83 (26%)	394.99 (59%)
Minimum average node weight	1228.68 (235%)	788.15 (145%)	605.41 (103%)	466.18 (88%)
Highest LBC	**366.26**	**322.30**	**298.06**	**247.77**

- Columns are for different edge weight distributions

Search strategies

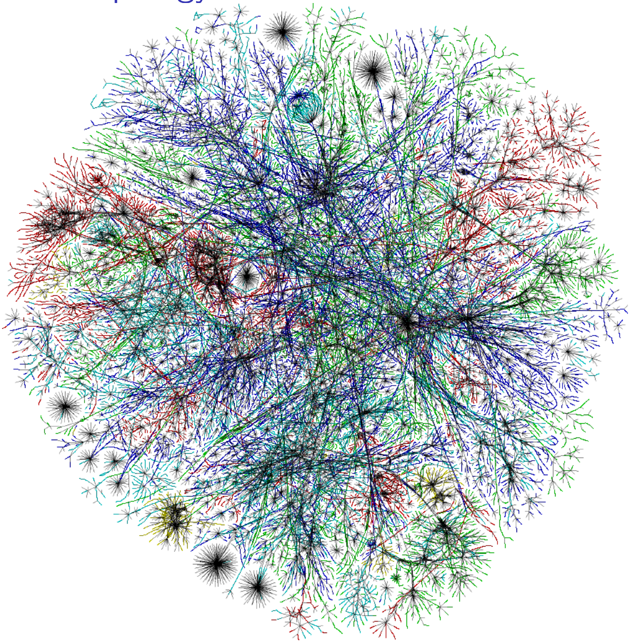
- Strategies:
  - If heterogeneity is small the best performing method is the minimum weight search, which outperforms methods using more information
  - If link weights get homogeneous ($\sigma \sim 1$) then minimum edge weight becomes random walk, highest LBS becomes highest degree and the latter performs better
  - In scale free networks: highest LBS performs best as it incorporates both degree and weight information
- Edge weights not shown is 1



(a) L(2) = **76.0**, L(3) = 42.0, L(4) = 42.0, L(5) = 0.0

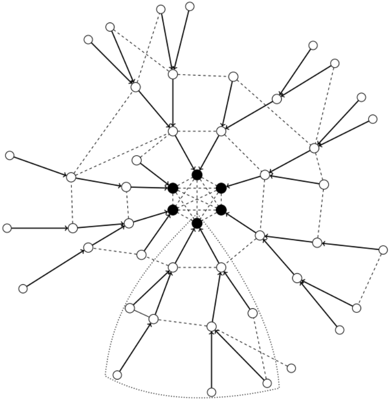
(b) L(2) = 76.0, L(3) = **92.0**, L(4) = 42.0, L(5) = 0.0

Internet topology



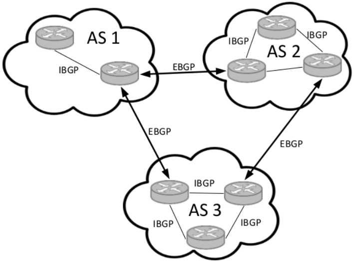
Optimization: costs

- Internet Autonomous system topology
- Providers can connect to the top tier or be a customer
- They are responsible for directing the Internet traffic
- Simple protocols define the routing (mainly greedy)
- Many optimizes the structure



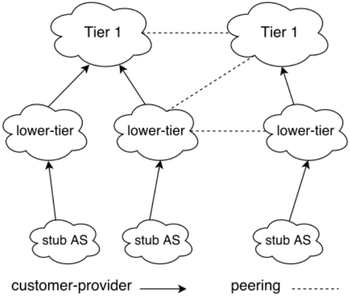
Internet topology

- Autonomous systems (AS) of the Internet
- Routing between AS
- Must be *fully* navigable
- Impossible to know the full structure $\rightarrow$ local routing



Internet topology

- Traffic
  - local
  - transit
- Relationship
  - customer-provider
  - peering

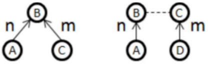


AS routing policy

1. Valley-free route
2. Highest local preference
3. Shortest AS path
4. etc.

AS routing policy

1. Valley-free route
   - Flow of traffic must coincide with the flow of cash
   - Data forward from AS A to B only if
     - incoming traffic is from a costumer of A
     - or B is a costumer of A
   - A valid path contains n customer-provider, at most 1 peer and m provider-customer link strictly in this order



Valley-free

not Valley-free

2. Highest local preference
3. Shortest AS path
4. etc.

AS routing model

- Number of players $\mathcal{P}$
- Edges: (p) directed *provider*, (r) undirected *peer*
- Valley-free routing: u can forward traffic coming from w to v only if
  1. w is a costumer of u (the relationship between u and v can be anything)
  2. v is costumer of u (the relationship between u and w can be anything)
- Payoff (of u):

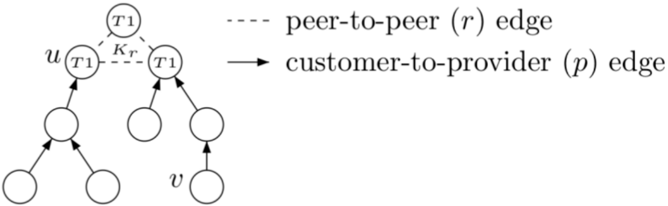
$$C_u = \sum_{v \neq u} d_{VF}(u, v) + \phi_p u_p + \phi_r u_r$$

where ϕ_x is the cost of an edge of type $x \in \{r, p\}$, u_x is he number edges of type x

$$d_{VF}(u, v) = \begin{cases} 0 & \text{There is a VF path between } u, v \\ \infty & \text{otherwise} \end{cases}$$

AS routing model

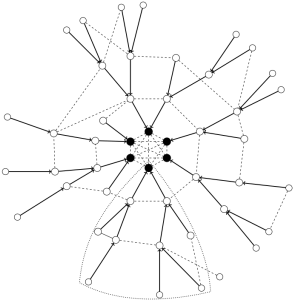
- Number of players $\mathcal{P}$
- Edges: (p) directed *provider*, (r) undirected *peer*
- Valley-free routing between all pairs
- Independent of the cost functions (provided they are positive)
- Resulting network
  - Has a clique core with only peer (r) links
  - Trees rooted at the core consisting exclusively from provider links (p), the provider is always closer to the clique than the consumer



AS routing model

- Include *Highest Local Preference* rule
- Player always picks from the available VF paths according to its local interest
- Players do not like customer-provider links
- Cost function

$$d_{VF}(u,v) = \begin{cases} 0 & \text{VF + first is peer or p} \rightarrow \text{c} \\ 1 & \text{VF + first is c} \rightarrow \text{p} \\ \infty & \text{otherwise} \end{cases}$$



AS routing model

- *Valley-free* rule
- *Highest Local Preference* rule