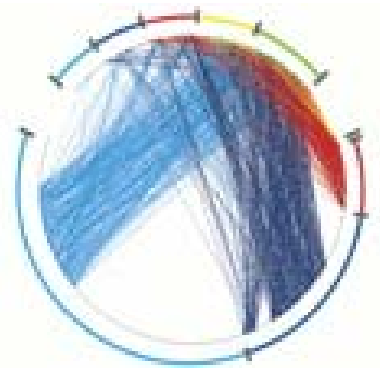
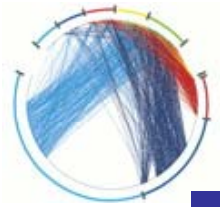
Lectures 3&4: Advanced Network Metrics

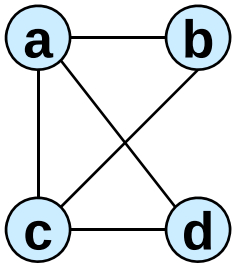


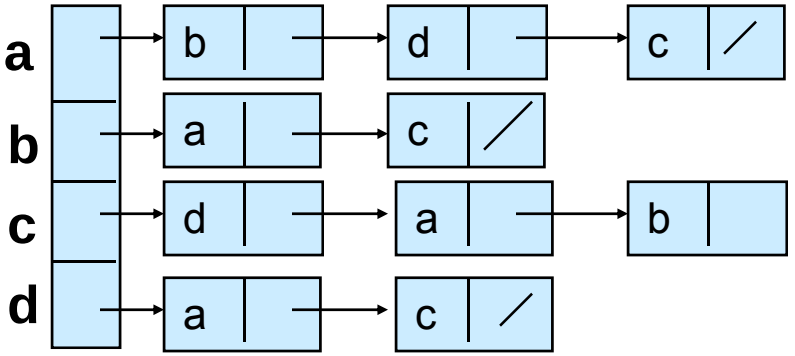


Representation of graphs

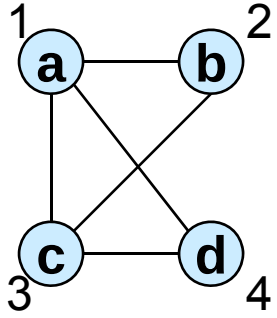
Having graph G = (V, E) with *V* = set of vertices and *E* = set of edges , there are two standard ways:

- Adjacency Lists

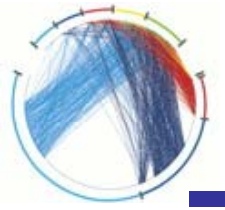




- Adjacency Matrix

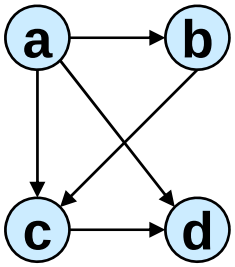


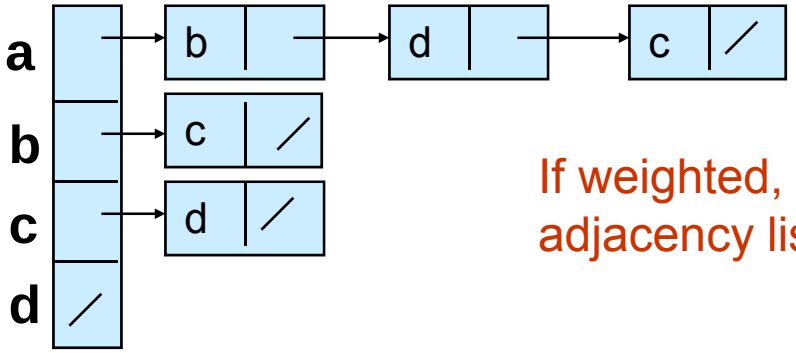
	1	2	3	4
1	0	1	1	1
2	1	0	1	0
3	1	1	0	1
4	1	0	1	0



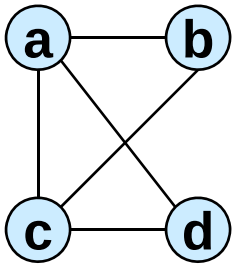
Adjacency lists

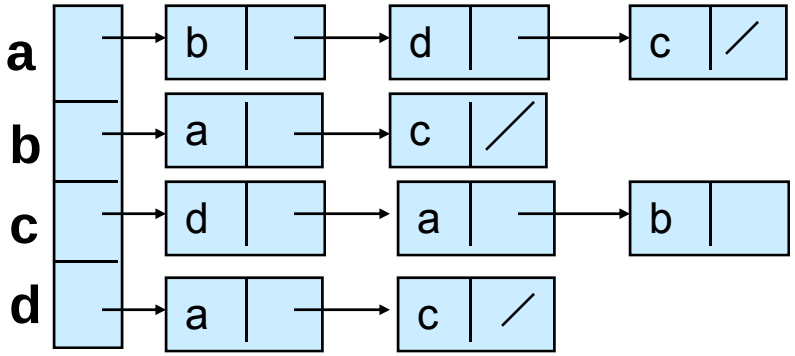
- Consists of an array Adj of |V| lists.
- One list per vertex.
- For u C V, Adj[u] consists of all vertices adjacent to *u*.

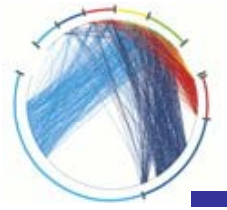




If weighted, store weights also in adjacency lists.







Storage requirements

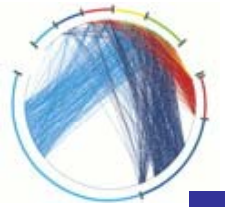
- For directed graphs:
  - Sum of lengths of all adj. lists is

    $\sum_{v \in V}$ out-degree(v) = |E|

  - Total storage: $\Theta(V+E)$
- For undirected graphs:
  - Sum of lengths of all adj. lists is

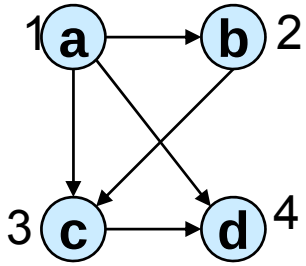
    $\sum_{v \in V}$ degree(v) = 2|E|

  - Total storage: $\Theta(V+E)$
- Advantages: i) Space-efficient, when a graph is sparse. ii) Can be modified to support many graph variants.
- Disadvantages: Determining of an edge (u,v) ε G is not efficient.
  - Have to search in u's adjacency list. Θ(degree(u)) time
  - Θ(V) in the worst case.

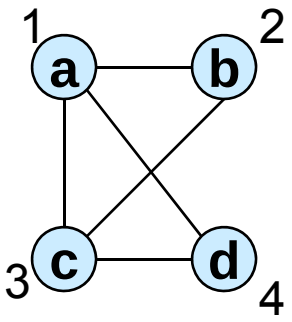


Adjacency matrix

- |V| number of nodes in matrix A.
- Number vertices from 1 to |V| in some arbitrary manner.
- A is then given by: $A[i,j] = a_{ij} = \begin{cases} 1 & \text{if } (i,j) \in E \\ 0 & \text{otherwise} \end{cases}$



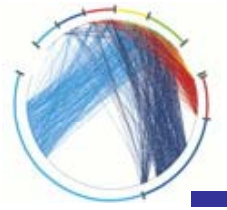
	1	2	3	4
1	0	1	1	1
2	0	0	1	0
3	0	0	0	1
4	0	0	0	0



	1	2	3	4
1	0	1	1	1
2	1	0	1	0
3	1	1	0	1
4	1	0	1	0

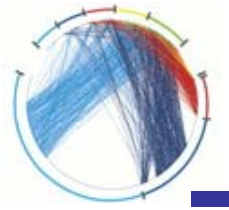
$A = A^T$ for undirected graphs.

Space: $\Theta(V^2)$. Not memory efficient for large graphs.
Time: to list all vertices adjacent to u: $\Theta(V)$.
Time: to determine if $(u, v) \in E$: $\Theta(1)$.
Can store weights instead of bits for weighted graph.



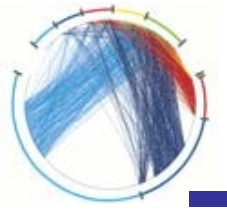
Graph-searching Algorithms

- Searching a graph:
  - Systematically follow the edges of a graph to visit the vertices of the graph.
- Used to discover the structure of a graph.
- Standard graph-searching algorithms.
  - Breadth-first Search (BFS).
  - Depth-first Search (DFS).

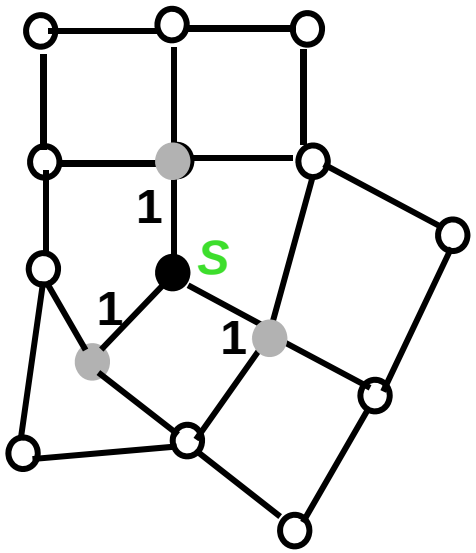


Breadth-first Search (BFS)

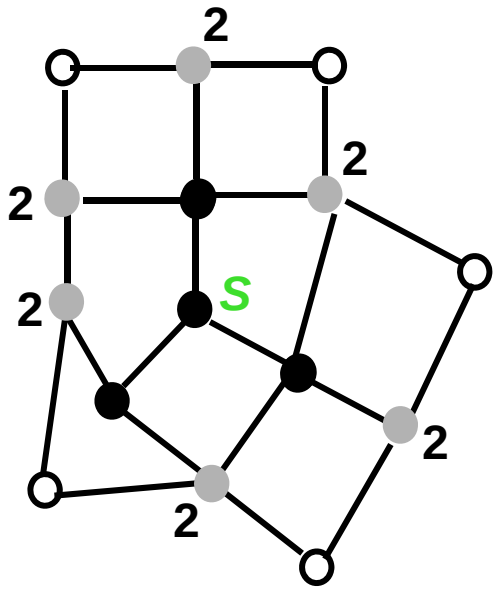
- **Input:** Graph $G = (V, E)$, either directed or undirected, and ***source vertex*** $s \in V$.
- **Output:**
  - $d[v]$ = distance (smallest # of edges, or shortest path) from s to v, for all $v \in V$. $d[v] = \infty$ if v is not reachable from s.
  - $\pi[v] = u$ such that (u, v) is last edge on shortest path $s \rightsquigarrow v$.
    - u is v's predecessor.
  - Builds breadth-first tree with root s that contains all reachable vertices.
- Expands the frontier between discovered and undiscovered vertices uniformly across the breadth of the frontier.
  - A vertex is "discovered" the first time it is encountered.
  - A vertex is "finished" if all vertices adjacent to it have been discovered.
- Colors the vertices to keep track of progress.
  - White–Undiscovered; Gray–Discovered but not finished; Black–Finished
    - Colors are required only to reason about the algorithm.
    - Can be implemented without colors.



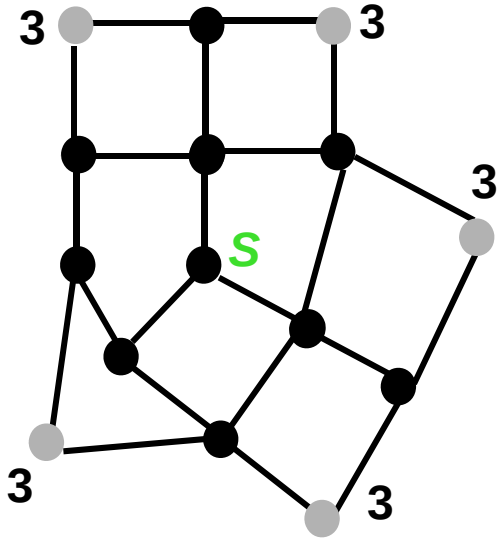
BFS for Shortest Paths



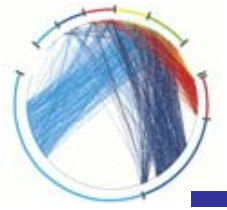
● Finished



● Discovered



○ Undiscovered



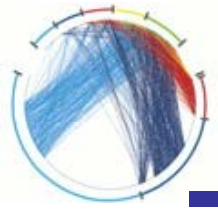
BFS for Shortest Paths

BFS(G,s)

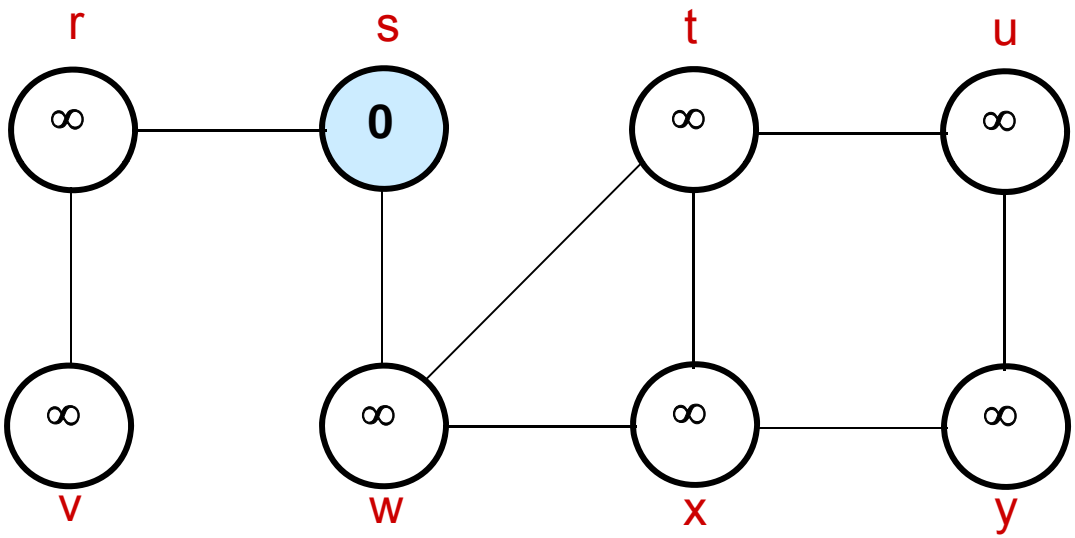
```
1.  for each vertex u in V[G] – {s}
2           do color[u] ← white
3              d[u] ← ∞
4              π[u] ← nil
5   color[s] ← gray
6   d[s] ← 0
7    π[s] ← nil
8   Q ← Φ
9   enqueue(Q,s)
10  while Q ≠ Φ
11          do u ← dequeue(Q)
12                  for each v in Adj[u]
13                          do if color[v] = white
14                          then color[v] ← gray
15                                  d[v] ← d[u] + 1
16                                   π[v] ← u
17                                   enqueue(Q,v)
18                  color[u] ← black
```

white: undiscovered
gray: discovered
black: finished

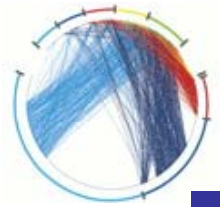
Q: a queue of discovered vertices
color[v]: color of v
d[v]: distance from s to v
π[u]: predecessor of v



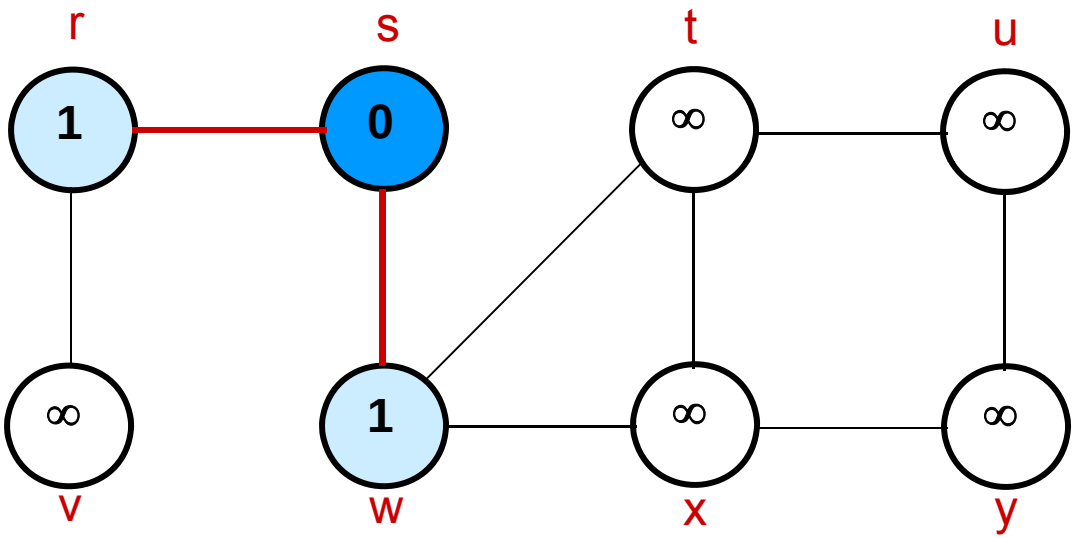
BFS (example)



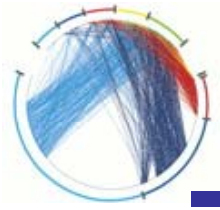
Q: s
0



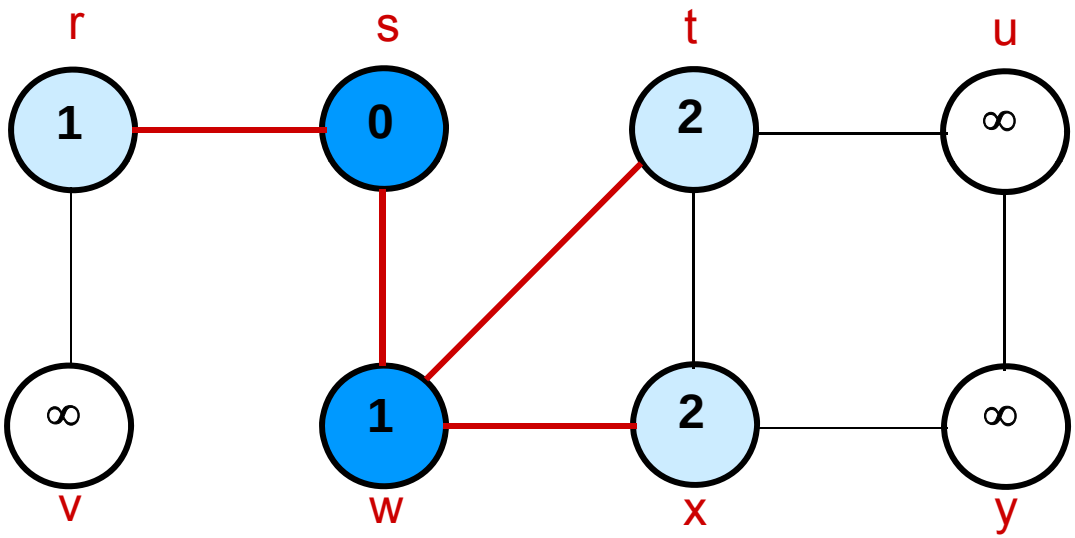
BFS (example)



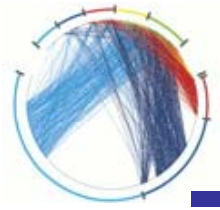
Q: w r
1 1



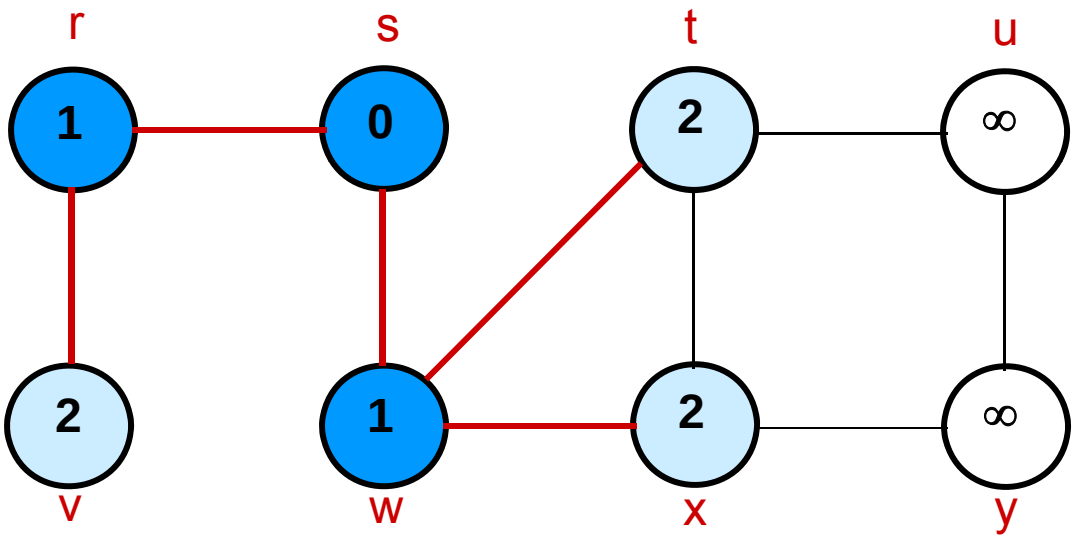
BFS (example)



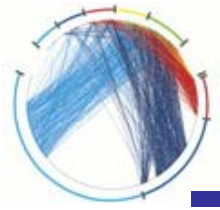
Q: r t x
1 2 2



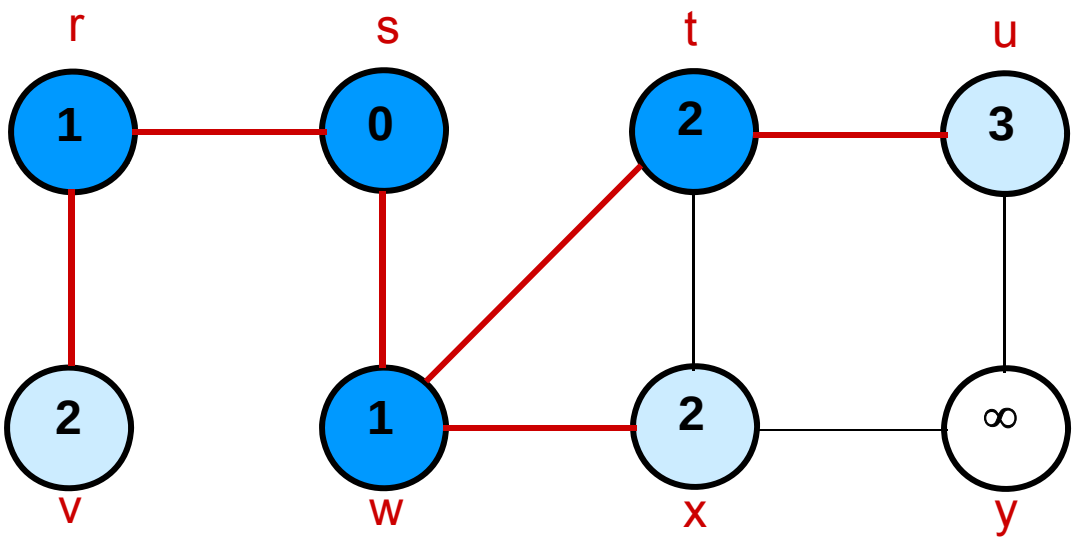
BFS (example)



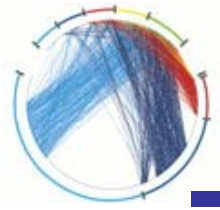
Q: t x v
2 2 2



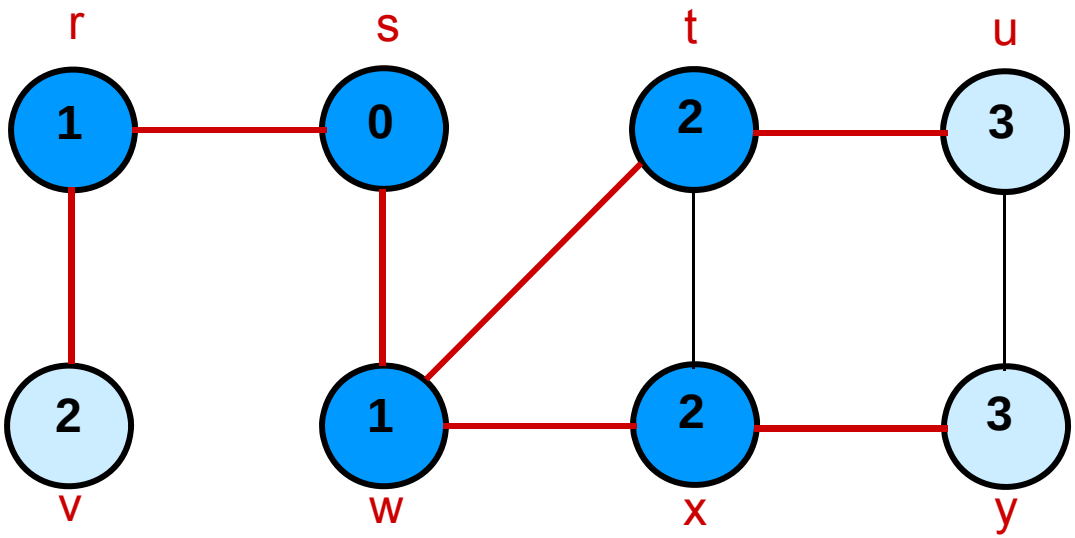
BFS (example)



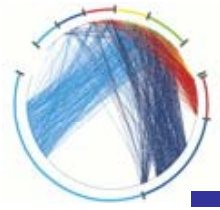
Q: x v u
2 2 3



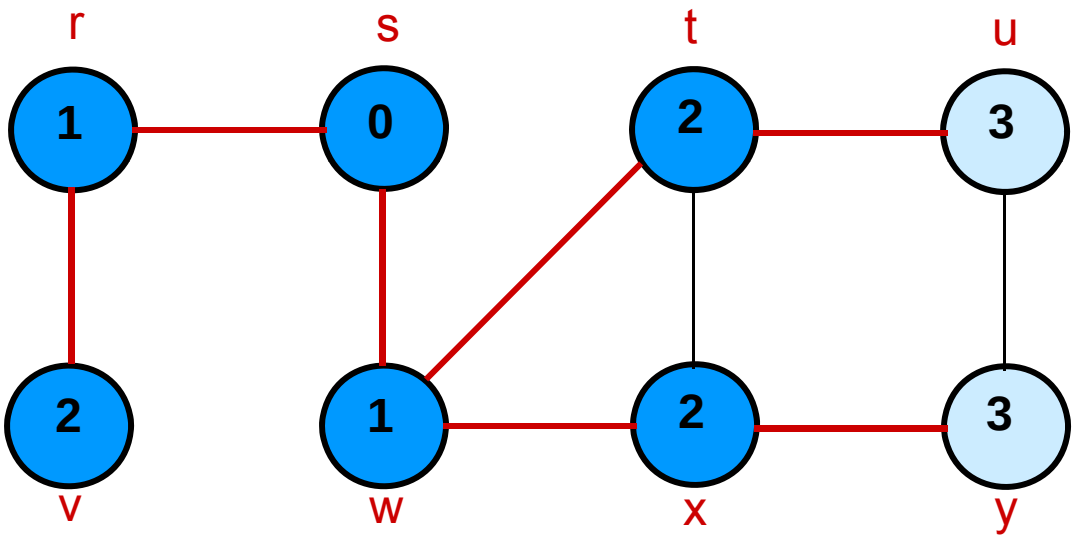
BFS (example)



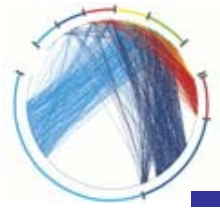
Q: v u y
2 3 3



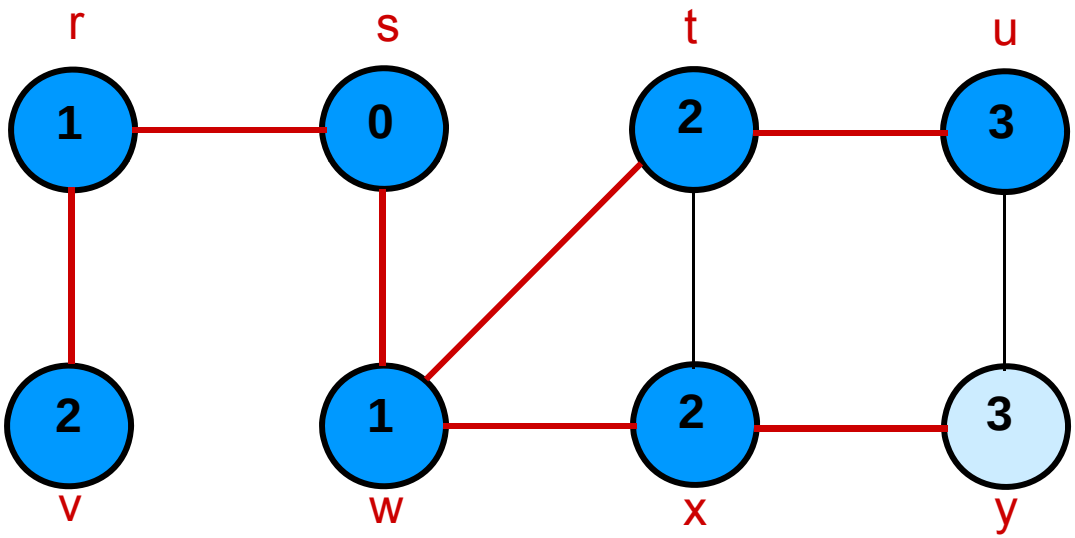
BFS (example)



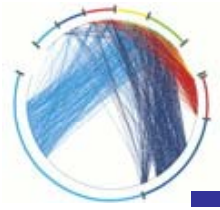
Q: u y
3 3



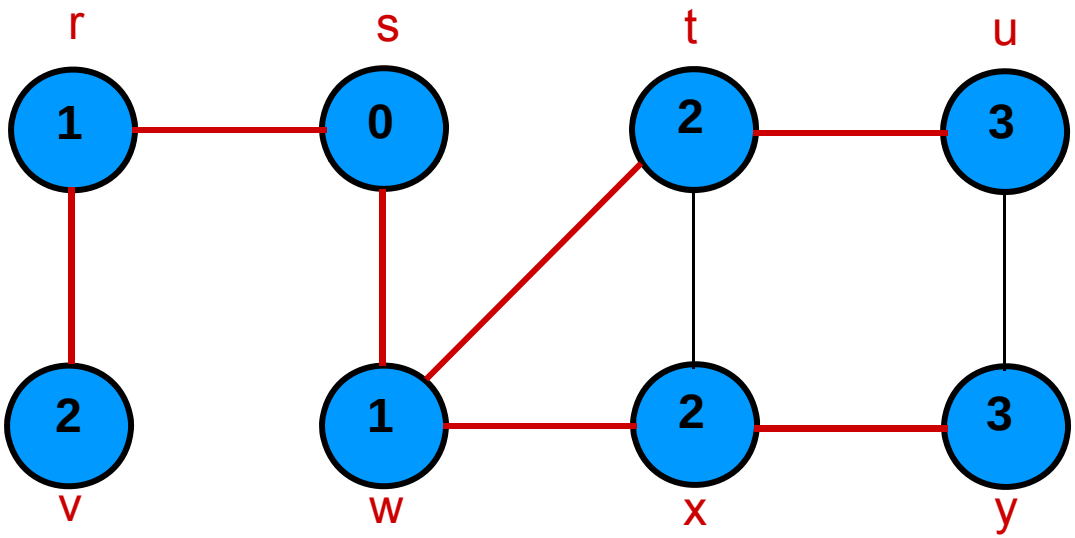
BFS (example)



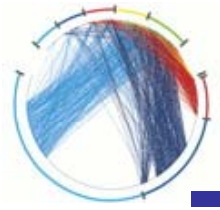
Q: y
3



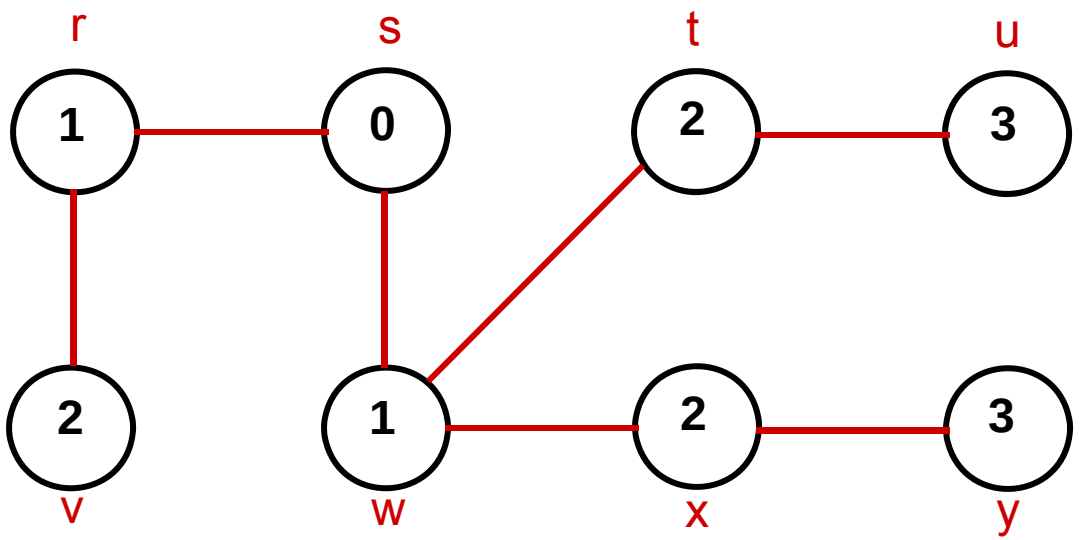
BFS (example)



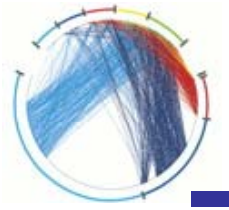
Q: ∅



BFS (example)

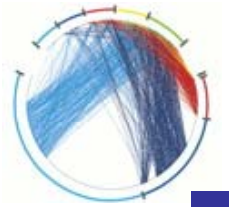


BF Tree



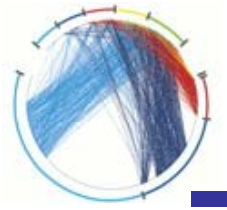
Analysis of BFS

- Initialization takes $O(V)$.
- Traversal Loop
  - After initialization, each vertex is enqueued and dequeued at most once, and each operation takes $O(1)$. So, total time for queuing is $O(V)$.
  - The adjacency list of each vertex is scanned at most once. The sum of lengths of all adjacency lists is $\Theta(E)$.
- Summing up over all vertices => total running time of BFS is $O(V+E)$, linear in the size of the adjacency list representation of graph.



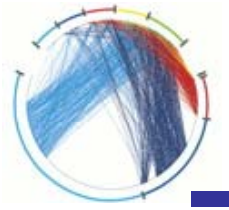
Breadth-first Tree

- For a graph $G = (V, E)$ with source s, the **predecessor subgraph** of G is $G_\pi = (V_\pi, E_\pi)$ where
  - $V_\pi = \{v \in V : \pi[v] \neq \text{NIL}\} \cup \{s\}$
  - $E_\pi = \{(\pi[v], v) \in E : v \in V_\pi - \{s\}\}$
- The predecessor subgraph G_π is a **breadth-first tree** if:
  - V_π consists of the vertices reachable from s and
  - for all $v \in V_\pi$, there is a unique simple path from s to v in G_π that is also a shortest path from s to v in G.
- The edges in E_π are called **tree edges**.
  $|E_\pi| = |V_\pi| - 1$.



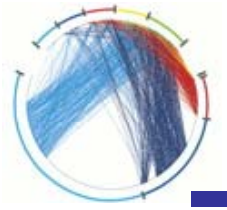
Depth-first Search (DFS)

- Explore edges out of the most recently discovered vertex v.
- When all edges of v have been explored, backtrack to explore other edges leaving the vertex from which v was discovered (its *predecessor*).
- "Search as deep as possible first."
- Continue until all vertices reachable from the original source are discovered.
- If any undiscovered vertices remain, then one of them is chosen as a new source and search is repeated from that source.



DFS

- **Input:** $G = (V, E)$, directed or undirected. No source vertex given!
- **Output:**
  - 2 **timestamps** on each vertex. Integers between 1 and 2|V|.
    - $d[v]$ = ***discovery time*** (v turns from white to gray)
    - $f[v]$ = ***finishing time*** (v turns from gray to black)
  - $\pi[v]$: predecessor of $v = u$, such that v was discovered during the scan of u's adjacency list.
- Uses the same coloring scheme for vertices as BFS.



DFS

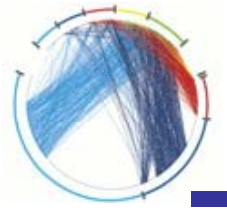
DFS(*G*)

```
1. for each vertex u ε V[G]
2.       do color[u] ← white
3.            π[u] ← NIL
4. time ← 0
5. for each vertex u ε V[G]
6.       do if color[u] = white
7.                then DFS-Visit(u)
```

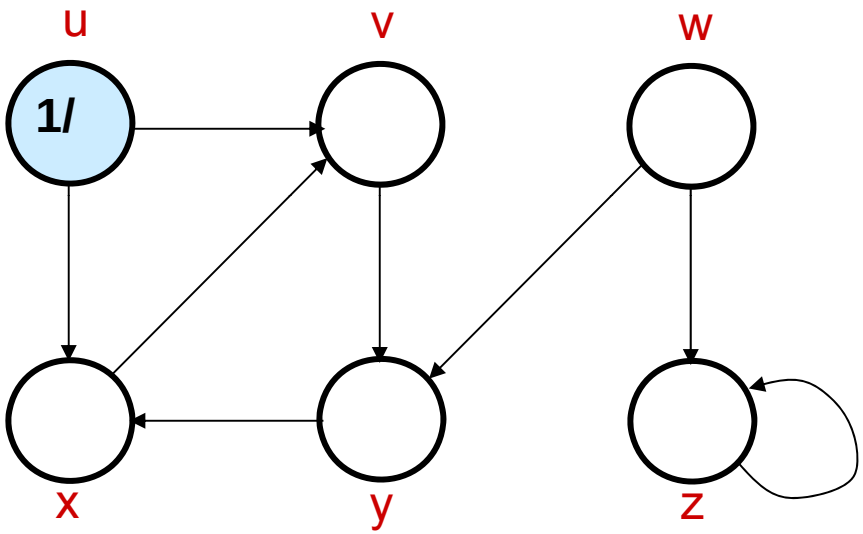
Uses a global timestamp ***time***.

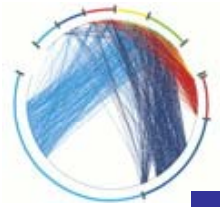
DFS-Visit(*u*)

```
1.    color[u] ← GRAY   ∇ White vertex u
      has been discovered
2.    time ← time + 1
3.    d[u] ← time
4.    for each vε Adj[u]
5.          do if color[v] = WHITE
6.                then π[v] ← u
7.                     DFS-Visit(v)
8.    color[u] ← BLACK     ∇ Blacken u;
      it is finished.
9.    f[u] ← time ← time + 1
```

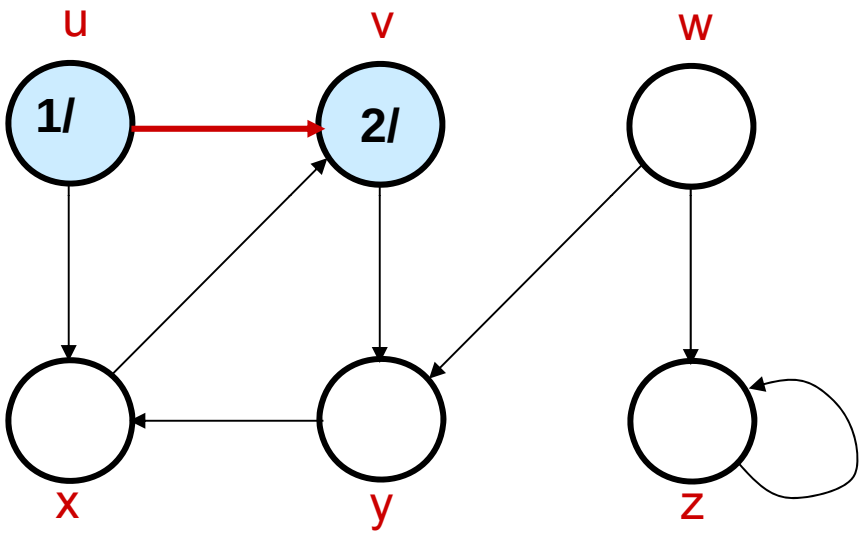


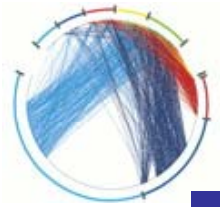
DFS (example)



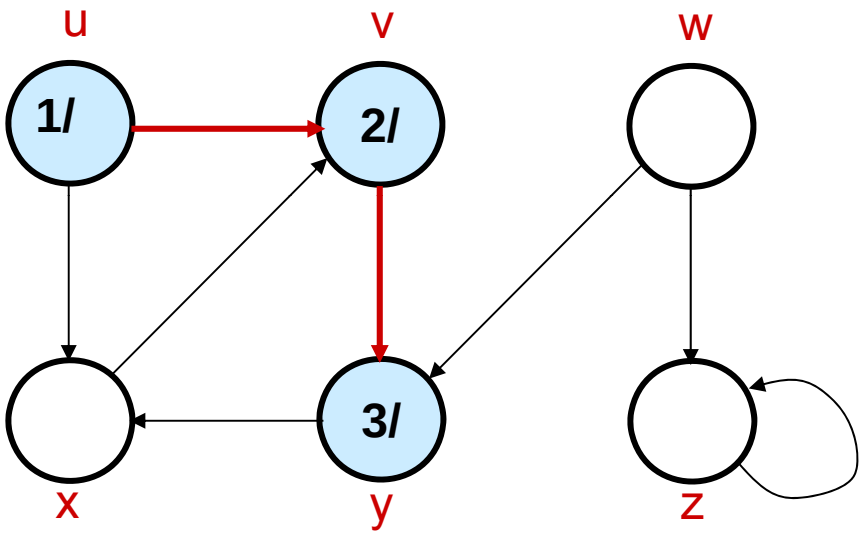


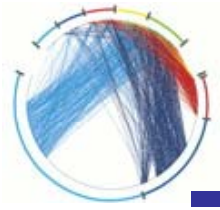
DFS (example)



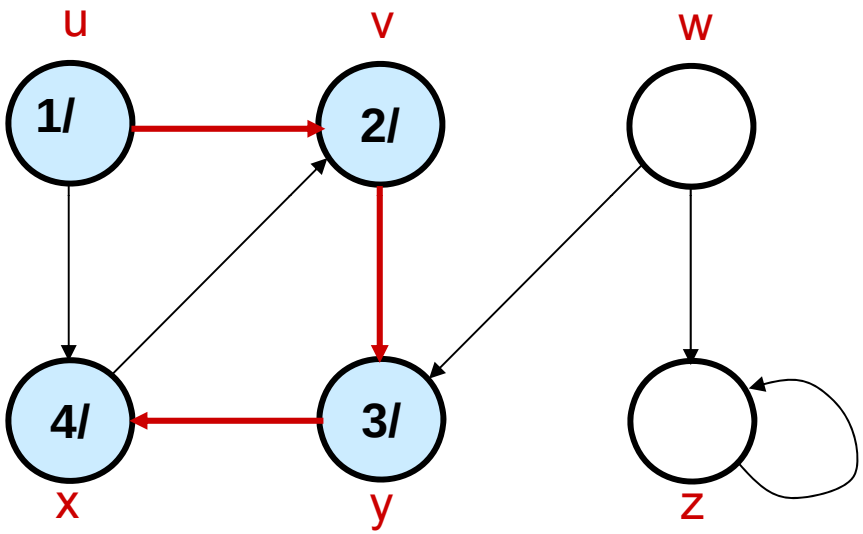


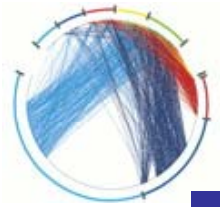
DFS (example)



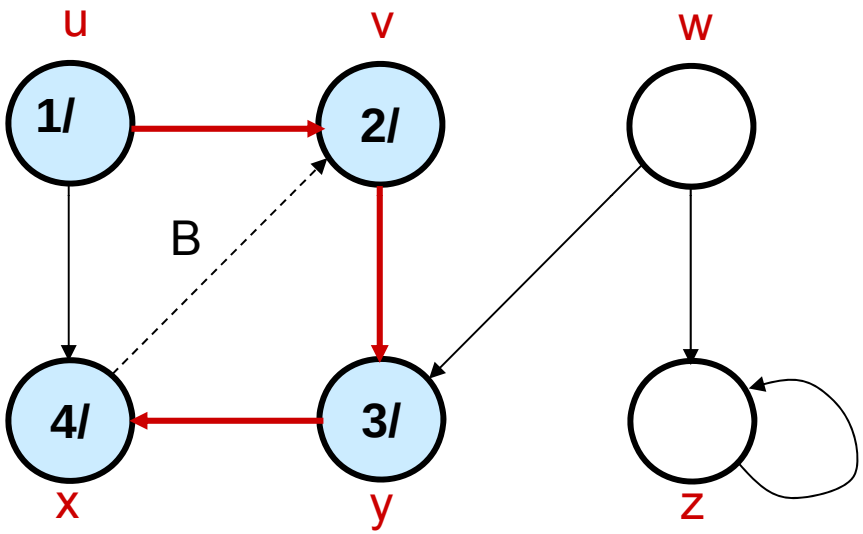


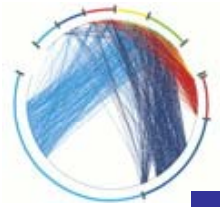
DFS (example)



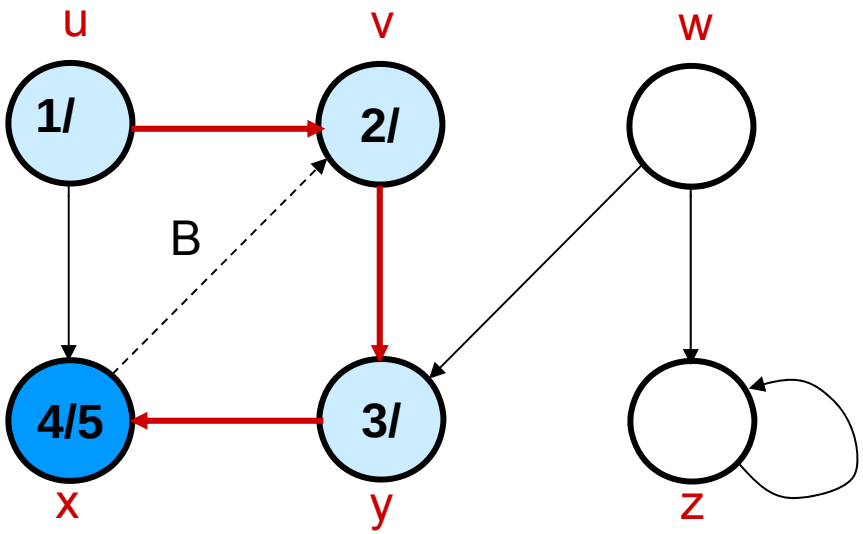


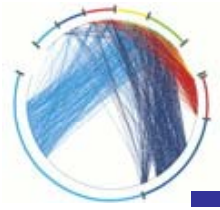
DFS (example)



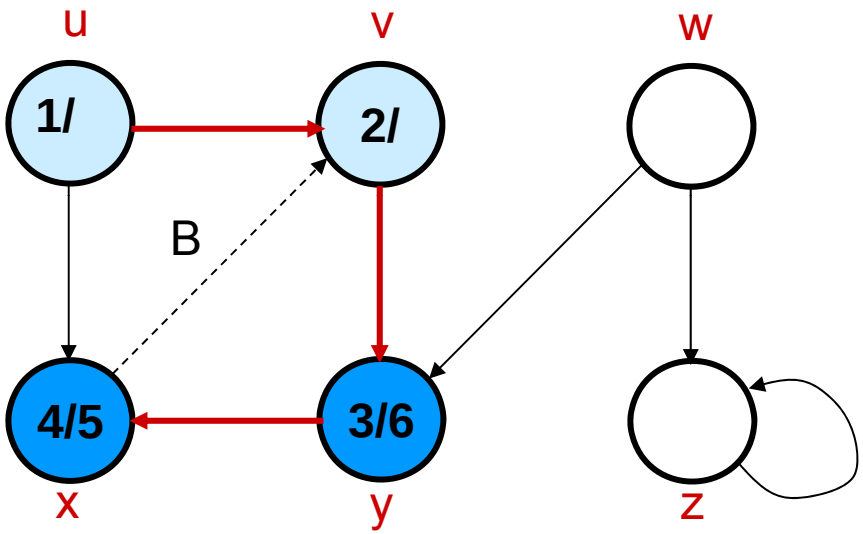


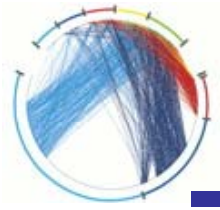
DFS (example)



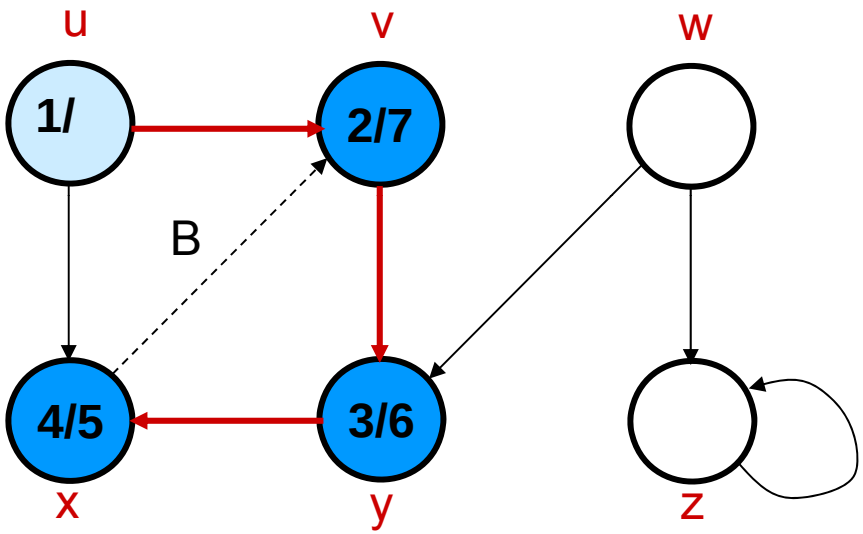


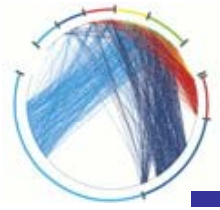
DFS (example)



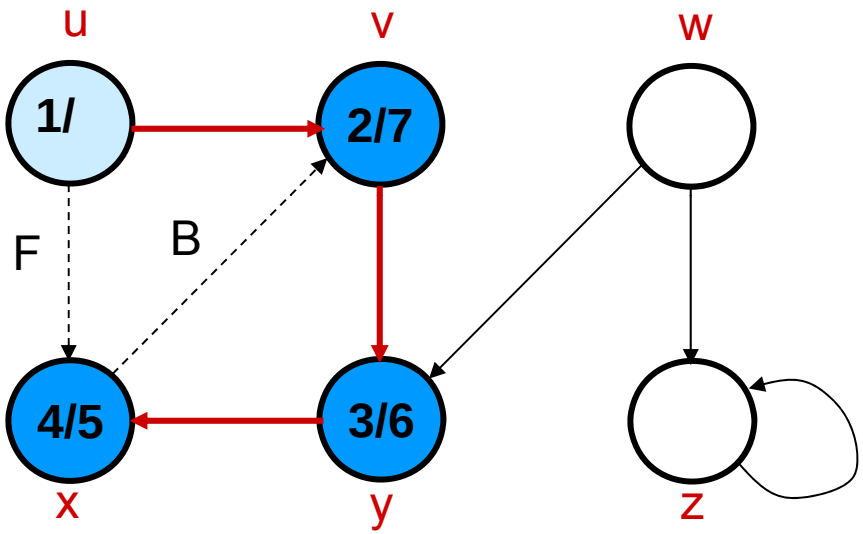


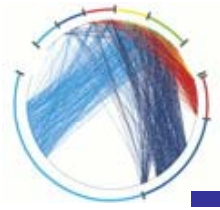
DFS (example)



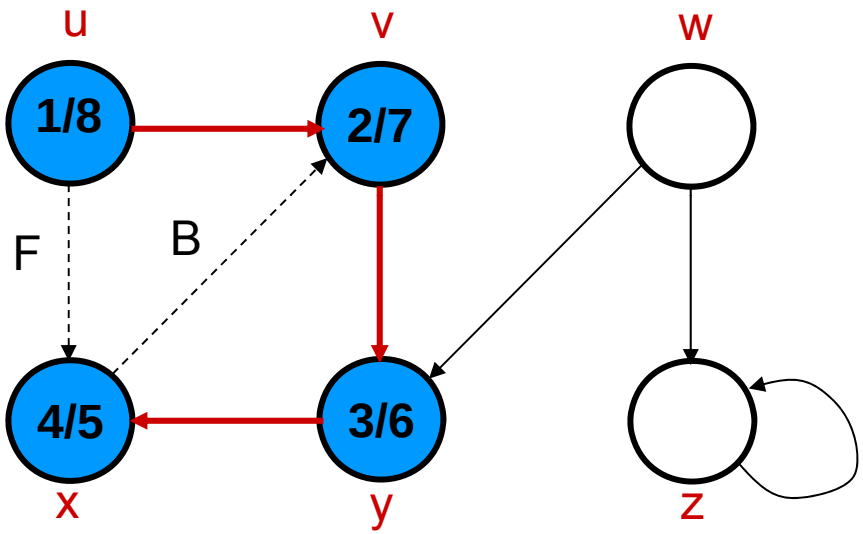


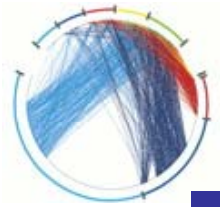
DFS (example)



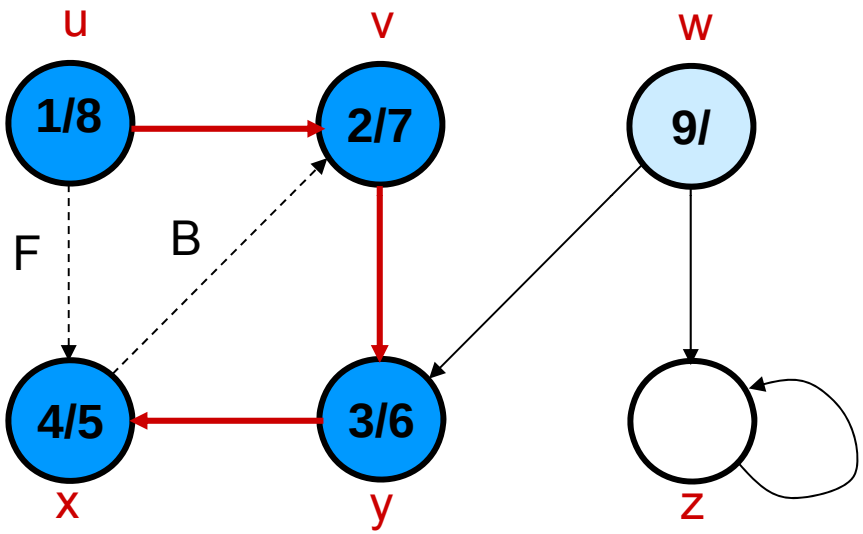


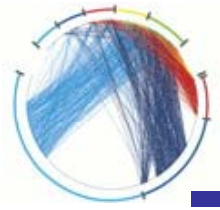
DFS (example)



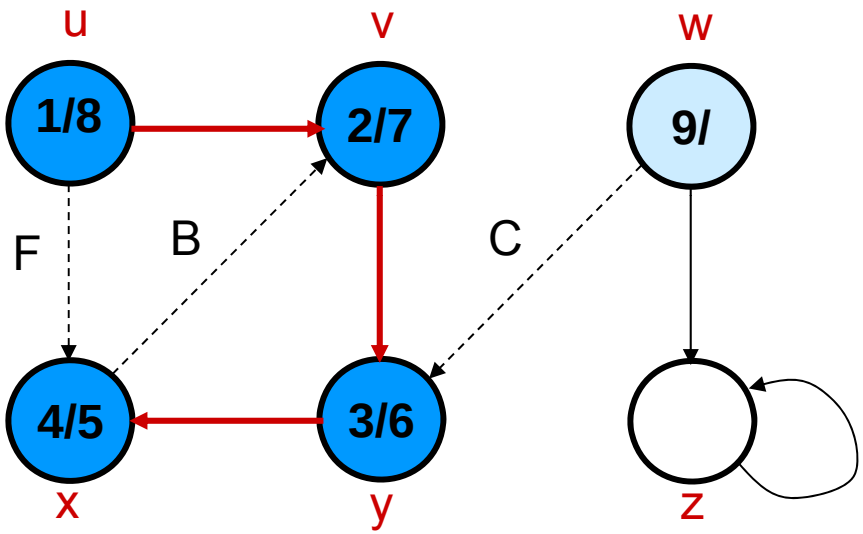


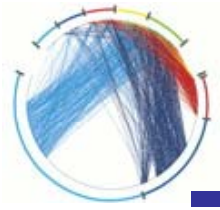
DFS (example)



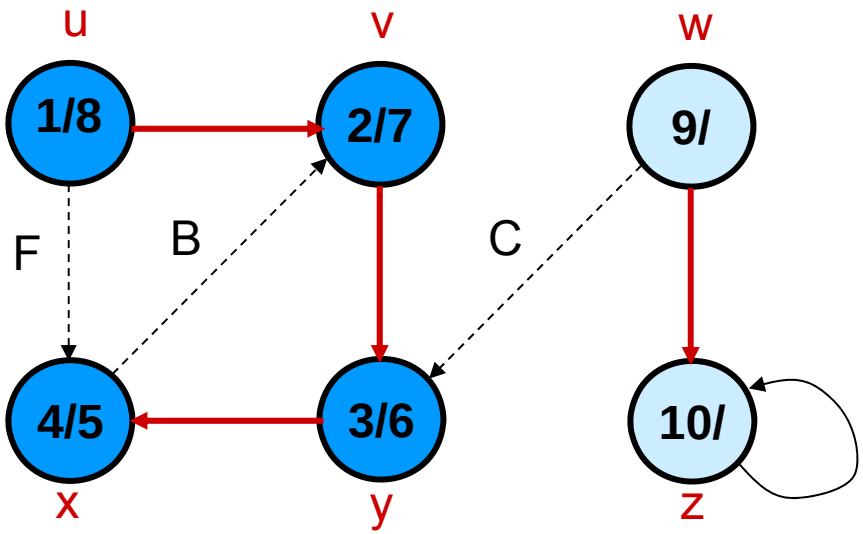


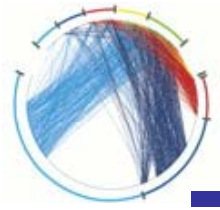
DFS (example)



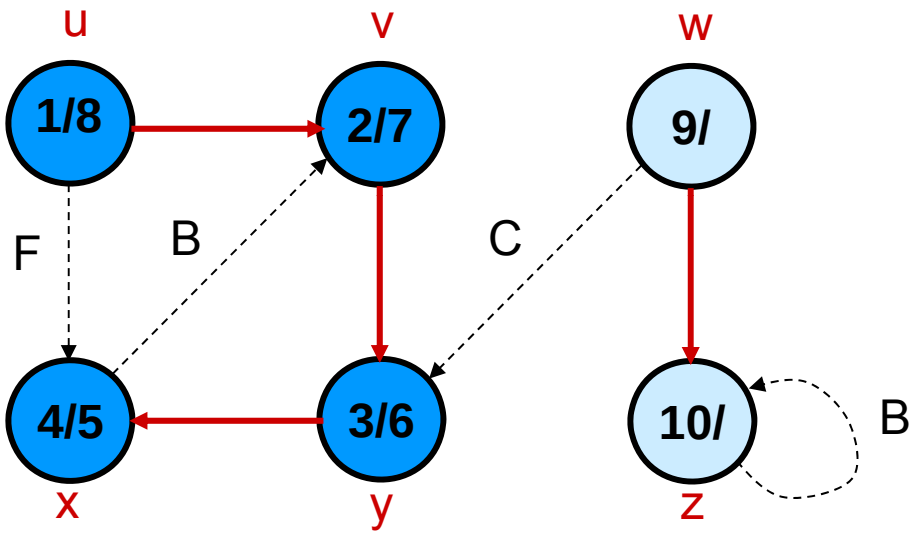


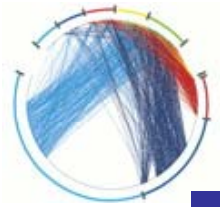
DFS (example)



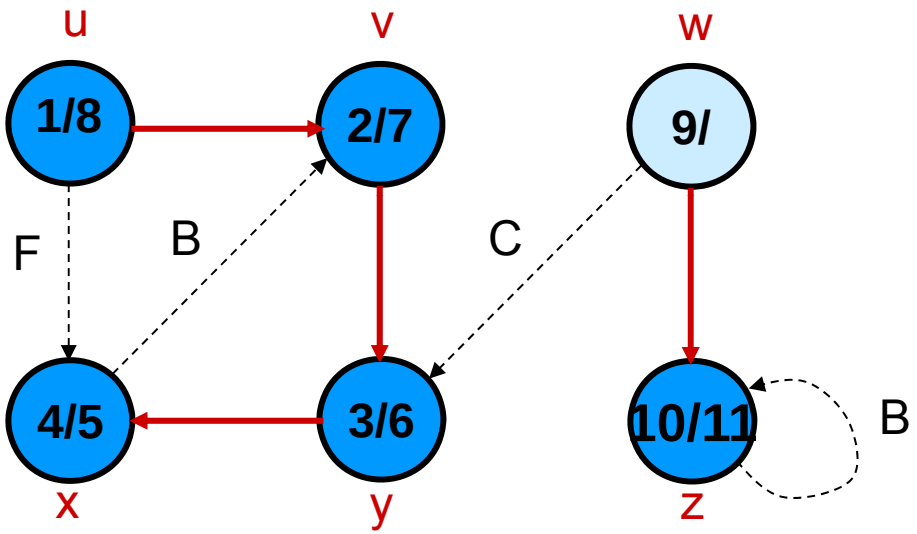


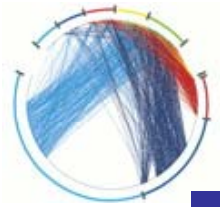
DFS (example)



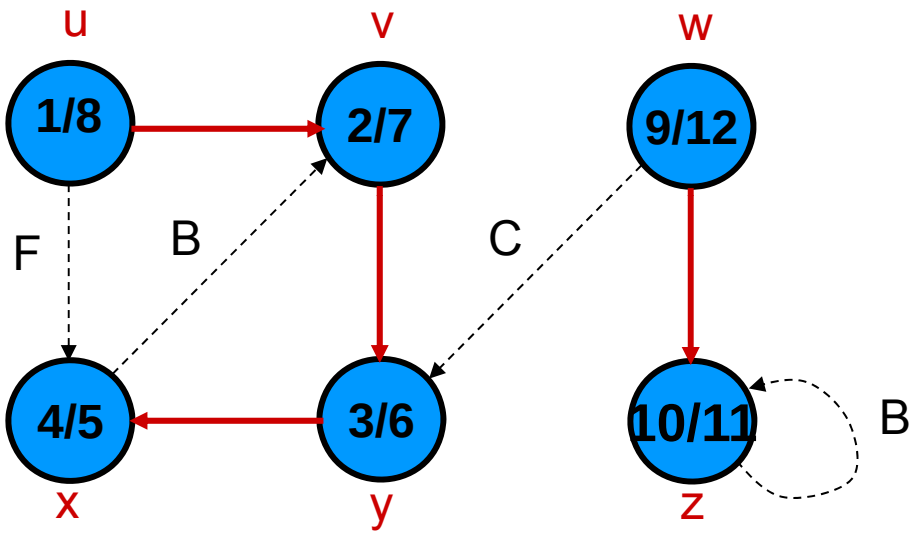


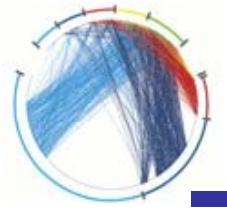
DFS (example)





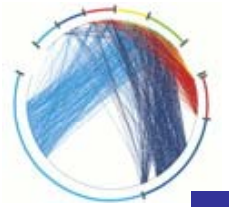
DFS (example)





Analysis of DFS

- Loops on lines 1-2 & 5-7 take $\Theta(V)$ time, excluding time to execute DFS-Visit.
- DFS-Visit is called once for each white vertex $v \in V$ when it's painted gray the first time. Lines 3-6 of DFS-Visit is executed $|\text{Adj}[v]|$ times. The total cost of executing DFS-Visit is $\sum_{v \in V} |\text{Adj}[v]| = \Theta(E)$
- Total running time of DFS is $\Theta(V+E)$.

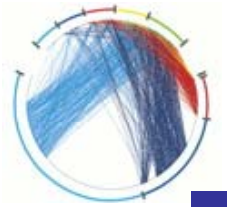


Depth-First Trees

- Predecessor subgraph defined slightly different from that of BFS.
- The predecessor subgraph of DFS is $G_\pi = (V, E_\pi)$ where $E_\pi = \{(\pi[v], v) : v \in V \text{ and } \pi[v] \neq \text{NIL}\}$.
  - How does it differ from that of BFS?
  - The predecessor subgraph G_π forms a *depth-first forest* composed of several *depth-first trees*. The edges in E_π are called *tree edges*.

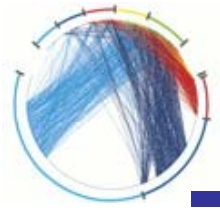
Definition:

Forest: An acyclic graph G that may be disconnected.



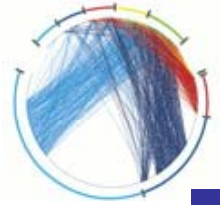
Dijkstra Path-Finding algorithm

- Assume that the average out-degree of a node is some constant k
- Initially,
  - Mark the given node as known (path length is zero)
    - Probably This takes O(1) (constant) time
  - For each out-edge, set the distance in each neighboring node equal to the cost (length) of the out-edge, and set its predecessor to the initially given node
    - If each node refers to a list of k adjacent node/edge pairs, this takes O(k) = O(1) time, that is, constant time
    - Notice that this operation takes longer if we have to extract a list of names from a hash table This takes O(1) (constant) time



Dijkstra Path-Finding algorithm

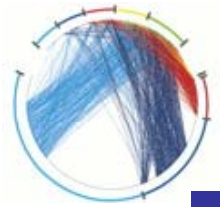
- Repeatedly (until all nodes are known),
  - Find an unknown node containing the smallest distance
    - Probably the best way to do this is to put the unknown nodes into a priority queue; this takes k * O(log n) time *each* time a new node is marked "known" (and this happens n times)
  - Mark the new node as known -- O(1) time
  - For each node adjacent to the new node, examine its neighbors to see whether their estimated distance can be reduced (distance to known node plus cost of out-edge)
    - If so, also reset the predecessor of the new node
    - There are k adjacent nodes (on average), operation requires constant time at each, therefore O(k) (constant) time
  - Combining all the parts, we get:
    O(1) + n*(k*O(log n)+O(k)), that is, O(nk log n) time



Characteristic path length

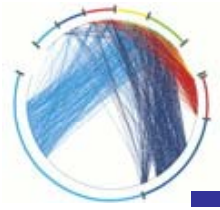
- A network with N nodes
- Compute the shortest path (distance) between any two nodes d_{ij}
- The length of the path is the number of edges (unweighted networks) or the weighted sum of the edges (weighted networks)
- If the nodes are not connected, the path length between them is set to infinity
- It is also called average geodesic distance
- If d_{ij} is infinity, it diverges

$$l = \frac{1}{N(N-1)} \sum_{i,j,i \neq j} d_{ij}$$



Betweenness Centrality

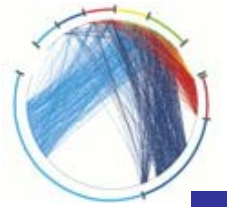
- A network with N nodes
- Compute the shortest path (distance) between any two nodes d_{ij}
- Compute the number of the shortest paths passing through an edge or node
- The length of the path is the number of edges (unweighted networks) or the weighted sum of the edges (weighted networks)
- Determines the load of the edges or nodes
- Possibility of weighted betweenness centrality



Edge Betweenness Centrality

$$\rho_{ij} = \sum_{p \neq q} \left(\Gamma_{pq}\left(e_{ij}\right) / \Gamma_{pq} \right)$$

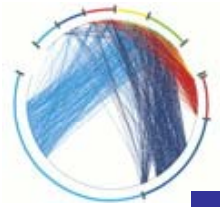
- Γ_{pq} is the number of shortest paths from the *p*-th to the *q*-th node
- $\Gamma_{pq}(e_{ij})$ is the number of these paths making use of e_{ij}.



Node Betweenness Centrality

$$C_i = \sum_{p \neq i \neq q} \left(\Gamma_{pq}(i) / \Gamma_{pq} \right)$$

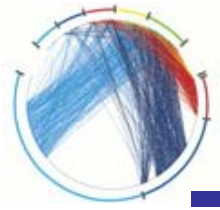
- Γ_{pq} is the number of shortest paths from the *p*-th to the *q*-th node
- $\Gamma_{pq}(i)$ is the number of these shortest paths making use of the *i*-th node (except those that are start or end nodes is *i*).



Efficiency

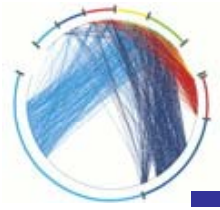
- In this way the divergence is avoided
- The inverse of efficiency E is called harmonic mean
- Efficiency is an indicator of the traffic capacity of the network
- The couple of disconnected nodes have a contribution of zero in computing E
- The more the values of E are the more the communication-efficient the network is
- It is also called global efficiency of the network

$$E = \frac{1}{N(N-1)} \sum_{i,j,i \neq j} \frac{1}{d_{ij}}$$



Vulnerability

- It is important to know which component (nodes or edges) are crucial to the best performance
- The more the drop in the efficiency by removing a component the more crucial that component
- Degree (hub node) might be a criterion
- Only degree is not enough, e.g. all vertices of a binary tree network have equal degree, i.e. no hub, but disconnection of vertices closer to the root and the root itself have a greater impact than of those near the leaves.
- The amount of change in the efficiency (or other network properties) as a component is removed can be an indicator of the vulnerability



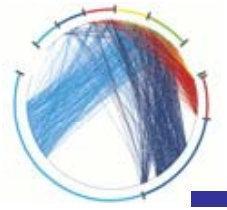
Vulnerability

$$V_i = \frac{E - E_i}{E}$$

where V_i is the vulnerability of component i and E_i is the efficiency of the networks by removed that component

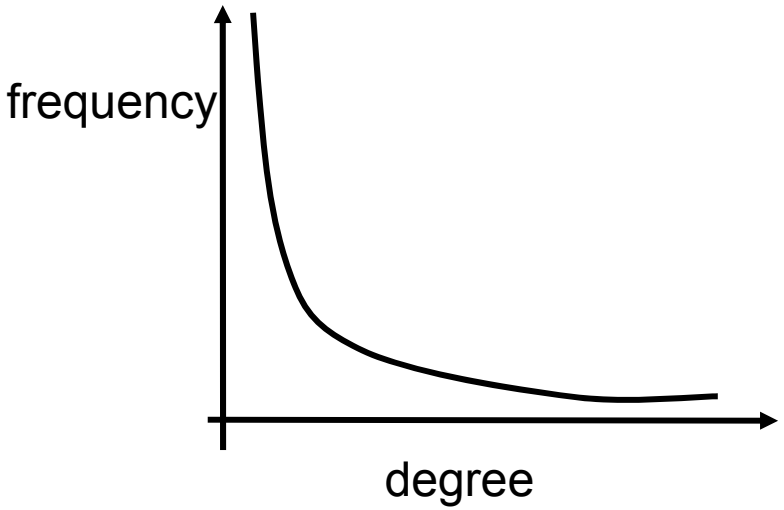
$$V = \max_i V_i$$

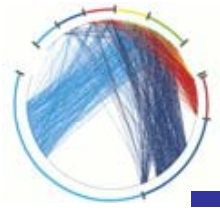
- V can be regarded as the vulnerability of the network
- the ordered distribution of nodes with respect to their vulnerability V_i is related to the network hierarchy
- The most vulnerable (critical) node occupies the highest position in the network hierarchy
- The same is also true for the edges



Degree and degree distribution

- Degree k_i of node i is a measure of its centrality
- Nodes with high degrees are called hubs
- Maximum degree $k_{max} = max_i(k_i)$ is also an important measure
- The variance of node-degrees can be an indicator of network heterogeneity, i.e. the more the variance the more the heterogeneity
- Degree distribution





Degree-degree correlation

- It is important to know if the nodes with degree k are connected to nodes with degree k'
- One way is to use the method proposed by Pastor Satoras et al. and plot the mean degree of the neighbors as a function of the degree

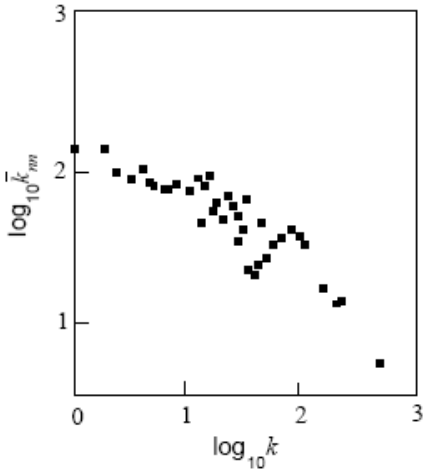
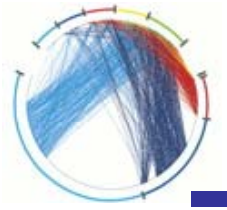


FIG. 3.13. Correlations of the degrees of nearest-neighbour vertices (autonomous systems) in the Internet at the interdomain level (after Pastor-Satorras, Vázquez, and Vespignani 2001). The empirical dependence of the average degree of the nearest neighbours of a vertex on the degree of this vertex is shown in a log–log scale. This empirical dependence was fitted by a power law with exponent approximately 0.5.

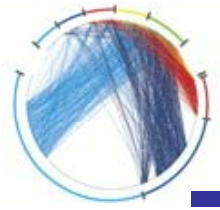


Degree-degree correlation

- Another way is to use the method proposed by Newman and compute the correlation coefficient
- Degree-degree correlation is computed as

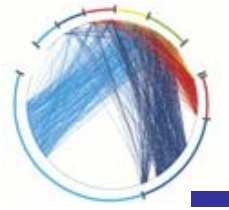
$$r = \frac{\frac{1}{E}\sum_{j>i} k_i k_j a_{ij} - \left[\frac{1}{E}\sum_{j>i}\frac{1}{2}\left(k_i + k_j\right)a_{ij}\right]^2}{\frac{1}{E}\sum_{j>i}\frac{1}{2}\left(k_i^2 + k_j^2\right)a_{ij} - \left[\frac{1}{E}\sum_{j>i}\frac{1}{2}\left(k_i + k_j\right)a_{ij}\right]^2}$$

- E is the total number of edges
- a_{ij} is the entry (i,j) of the adjacency matrix
- k_i is the degree of node i



Degree-degree correlation

- r > 0: the network is called assortative
  - Node with large degree intent to connect to those with large degrees and nodes with low degrees intend to connect to those with low degrees (rich with rich and poor with poor)
- r < 0: the network is called disassortative
  - Node with large degree intent to connect to those with low degrees and nodes with low degrees intend to connect to those with high degrees (rich with poor)
- r = 0: no correlations
  - There is no specific intention in the connection between the nodes in the sense of their degrees

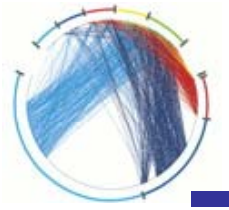


Bipartivity degree

- A special case of assortativity is for bipartite networks
- It is a known fact that a network is bipartite if and only if it has no loops of odd length
- Some networks are bipartite by construction, others, are only approximately bipartite
- A way to quantify how much a network is bipartite is needed
- A possible measurement is based on the number of edges between nodes of the same subset in the best possible division

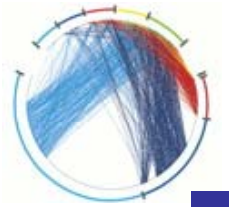
$$b = 1 - \frac{\sum_{i,j} a_{ij} \delta_{\vartheta(i),\vartheta(j)}}{\sum_{i,j} a_{ij}}$$

- ϑ(i) maps the nodes i to its type and δ is the Kronecker delta



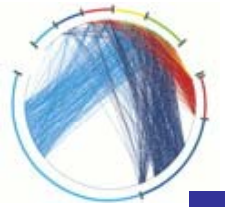
Bipartivity degree

- The smallest value of b for all possible divisions is the bipartivity of the network
- However, its computation is NP-complete, due to the necessity of evaluating b for the best possible division
- A measurement that approximates b, but is computationally easier has been proposed that is based on a process of marking the minimum possible number of edges as responsible for the creation of loops of odd length.
- Another approach is based on the subgraph centrality



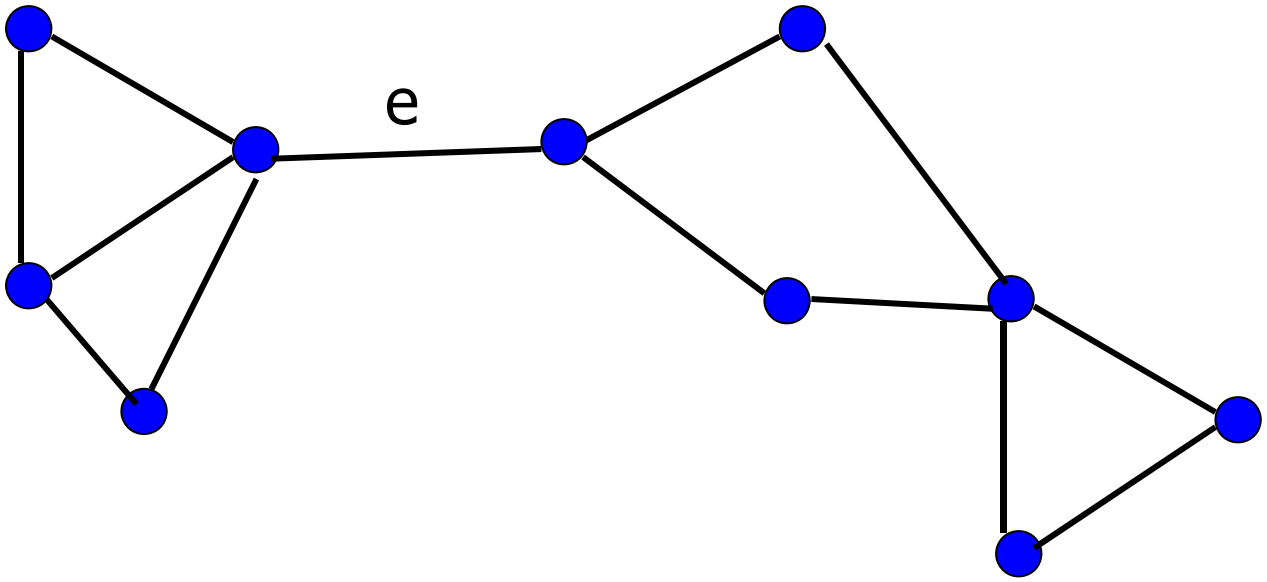
Disconnecting and cut sets

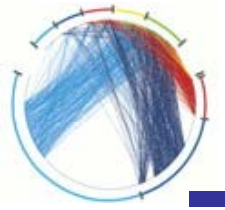
- How many edges or nodes must be removed in order to disconnect an originally connected graph?
- If a node is removed then all edges joining it will also be removed
- But, the converse may not be true, i.e. an edge may removed without necessarily removing the nodes tipping to that
- Disconnecting set: a set of edges $E_o(G)$, after it is being removed, the graph G will be become disconnected
- Cut set: the smallest disconnecting set, i.e. No proper subset of which is a disconnecting set



Bridge

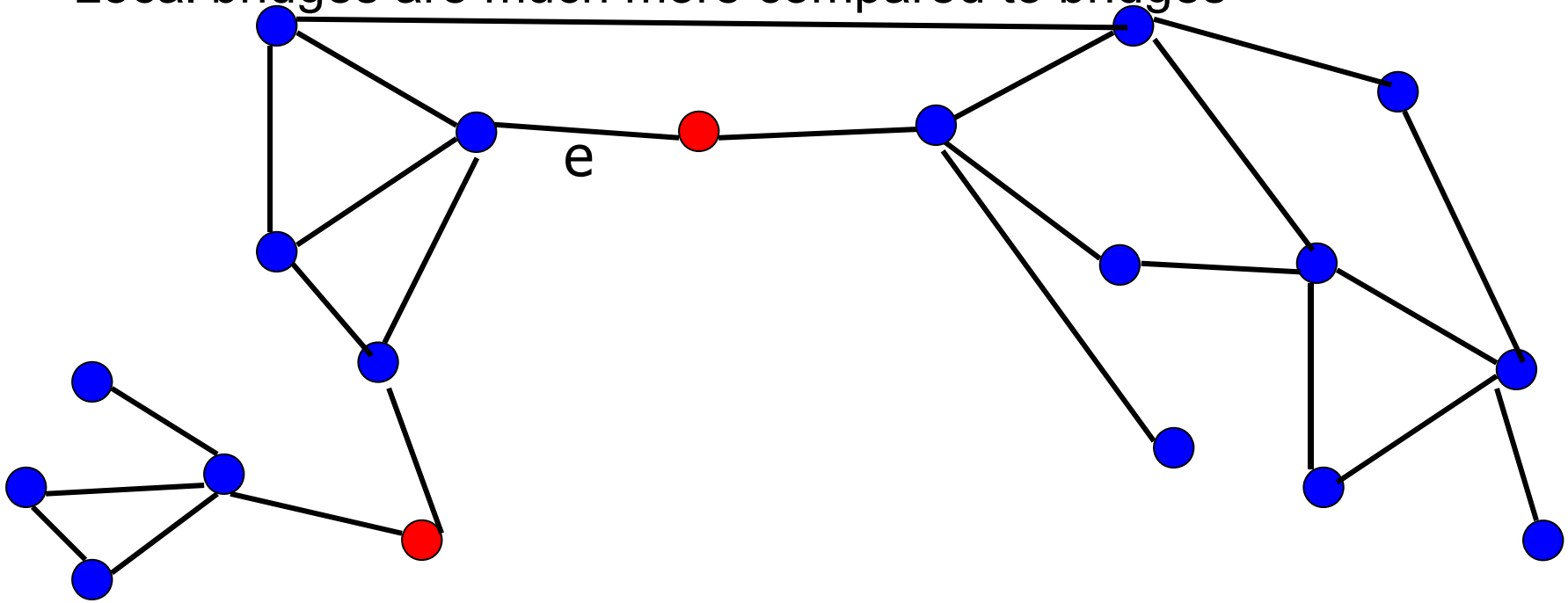
- A cut set with only one edge is called a bridge
- Deleting a bridge will cause the network separated in two different components
- Example: the cut set {e} below

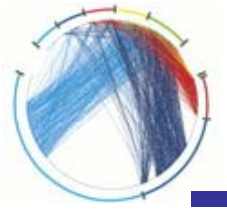




Bridge

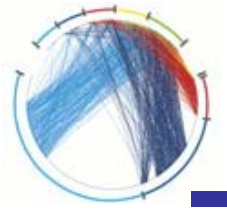
- Why bridges are important
- In a network, a node with low degree (red nodes below) might be more important than a node with higher degree → bridge
- e is a local bridge if by its removal the new distance between its tipping nodes (span) increases strictly more than two
- Local bridges are much more compared to bridges





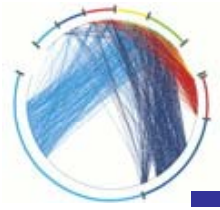
Strong and weak ties

- Bridging the local and global is one of the important roles of the networks
- Strength of weak ties hypothesis in sociology:
  - In the late 1960s, Mark Granovetter interviewed people who had recently changed employers to learn how they discovered their new jobs
  - He found that many people learned information leading to their current jobs through personal contacts
  - These personal contacts were often described by interview subjects as acquaintances rather than close friends
  - This means that distant acquaintances are more constructive than close friends in helping you to get a new job



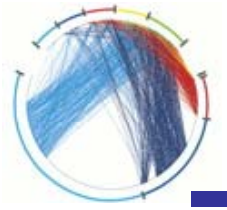
Strong and weak ties

- The answer that Granovetter proposed to this question is striking in the way it links two different perspectives on distant friendships
  - Structural: focusing on the way these friendships span different portions of the full network
  - Interpersonal: considering the purely local consequences that follow from a friendship between two people being either strong or weak
- In this way, the answer transcends the specific setting of job- seeking, and offers a way of thinking about the architecture of social networks more generally.



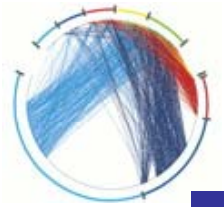
Triadic closure

- triadic closure principle:
  - If two people in a social network have a friend in common, then there is an increased likelihood that they will become friends themselves at some point in the future
- If three nodes are all-to-all connected, they form a triangle.
- If we observe snapshots of a social network at two distinct points in time, then in the later snapshot, we generally find a significant number of new edges that have formed through this triangle-closing operation, between two people who had a common neighbor in the earlier snapshot.



Clustering coefficient

- It is to measure the local connectivity in the network
- Shows somehow the local information exchange
- Originally comes from social sciences
- Shows to how much extent the friends (neighbors) of two connected nodes are connected themselves
- Measures the density of triangles (local clusters) in the networks
- Two different ways to measure it

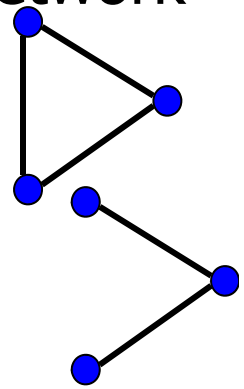


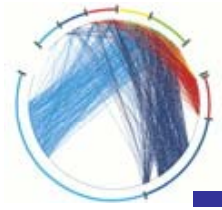
Clustering coefficient 1 (transitivity)

$$\begin{cases} C = \dfrac{3N_{\Delta}}{N_3} \\ N_{\Delta} = \sum_{k>j>i} a_{ij}a_{ik}a_{jk} \\ N_3 = \sum_{k>j>i} \left(a_{ij}a_{ik} + a_{ji}a_{jk} + a_{ki}a_{kj} \right) \end{cases}$$

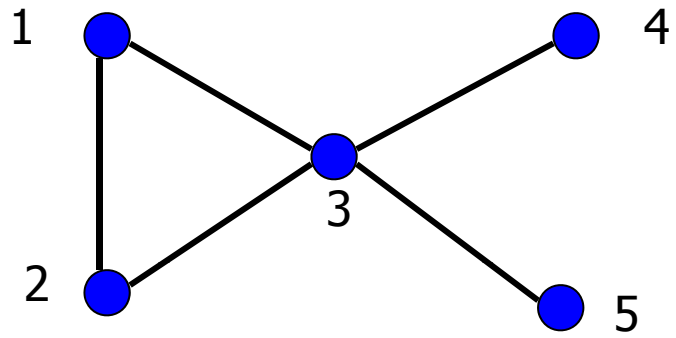
$$C = \frac{\sum_{i} \text{triangles centered at node i}}{\sum_{i} \text{triples centered at node i}}$$

- C is clustering coefficient and A= (a_{ij}) is the adjacency matrix
- N_{Δ} is the number of triangles (local clusters) in the network
- N_3 is the number of connected triples is the network

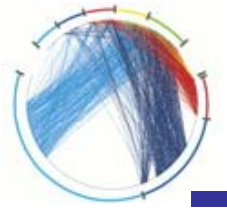




Clustering coefficient 1 (transitivity)



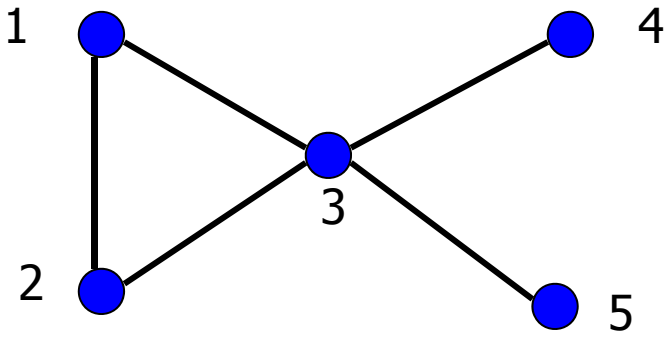
$$\begin{cases} N_{\Delta} = 1 \\ N_{3} = 8 \\ C = \dfrac{3}{8} \end{cases}$$



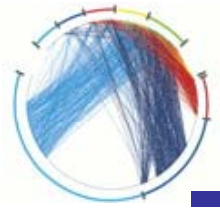
Clustering coefficient 2

$$\begin{cases} C_i = \dfrac{N_\Delta(i)}{N_3(i)}; C = \dfrac{1}{N}\sum_i C_i \\ N_\Delta = \sum_{k>j} a_{ij} a_{ik} a_{jk} \\ N_3 = \sum_{k>j} a_{ij} a_{ik} \end{cases}$$

$$C_i = \frac{\text{triangles centered at node i}}{\text{triples centered at node i}}$$



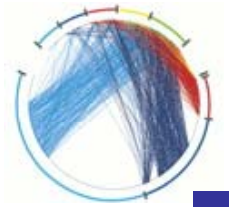
$$\begin{cases} C_1 = 1 \\ C_2 = 1 \\ C_3 = \dfrac{1}{6} \Rightarrow C = \dfrac{1}{5}\left(1+1+\dfrac{1}{6}\right) = \dfrac{13}{30} \\ C_4 = 0 \\ C_5 = 0 \end{cases}$$



Clustering coefficient of weighted networks

$$\begin{cases} s_i = \sum_i w_{ij} \\ k_i = \sum_i a_{ij} \\ C_i^w = \dfrac{1}{s_i\left(k_i - 1\right)} \sum_{j,k} \dfrac{w_{ij} + w_{ik}}{2} a_{ij} a_{ik} a_{jk} \\ C^w = \dfrac{1}{N} \sum_i C_i^w \end{cases}$$

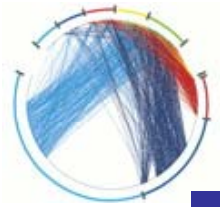
- A= (a_{ij}) is the adjacency matrix and W=(w_{ij}) is the weight matrix
- k_i is the degree and s_i is the strength of node i



Clustering coefficient of real networks

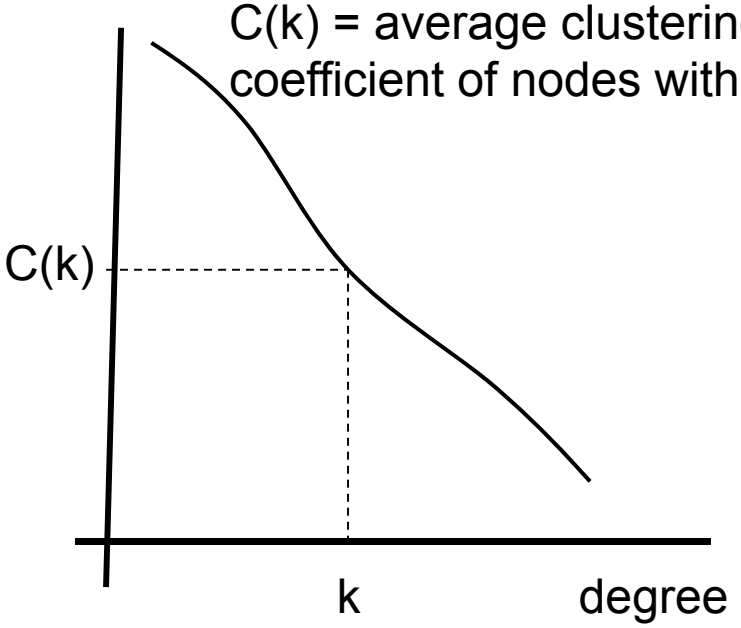
Table 1: Clustering coefficients, C, for a number of different networks; n is the number of node, z is the mean degree. Taken from [146].

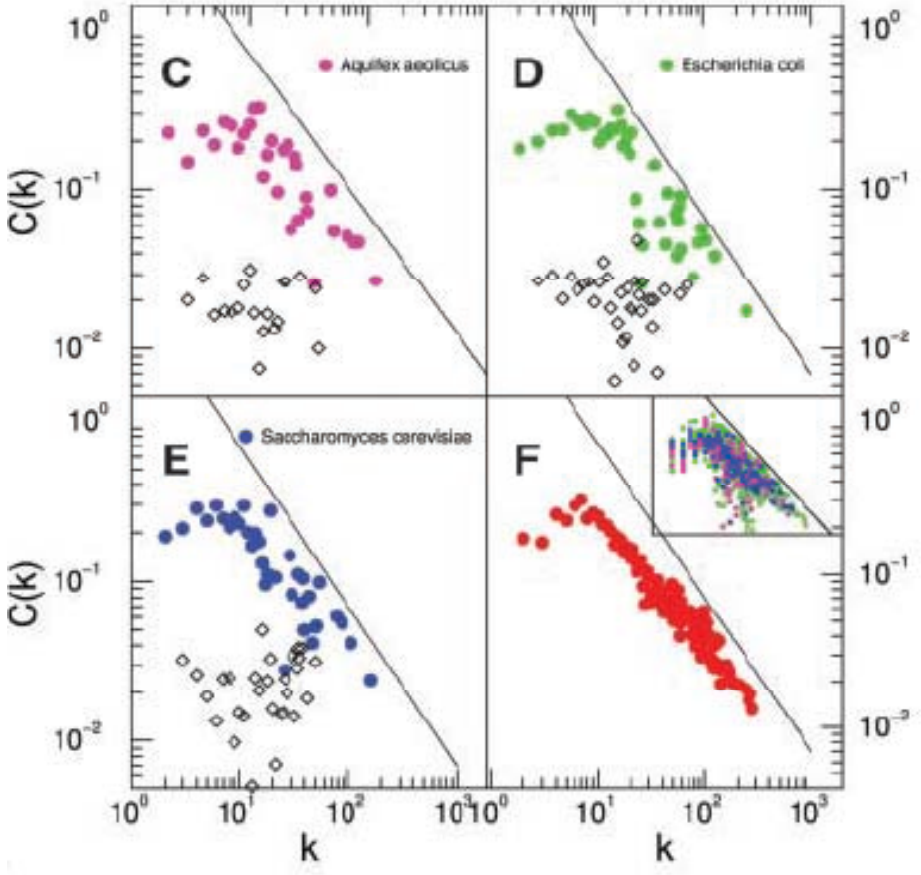
Network	n	z	C measured	C for random graph
Internet [153]	6,374	3.8	0.24	0.00060
World Wide Web (sites) [2]	153,127	35.2	0.11	0.00023
power grid [192]	4,941	2.7	0.080	0.00054
biology collaborations [140]	1,520,251	15.5	0.081	0.000010
mathematics collaborations [141]	253,339	3.9	0.15	0.000015
film actor collaborations [149]	449,913	113.4	0.20	0.00025
company directors [149]	7,673	14.4	0.59	0.0019
word co-occurrence [90]	460,902	70.1	0.44	0.00015
neural network [192]	282	14.0	0.28	0.049
metabolic network [69]	315	28.3	0.59	0.090
food web [138]	134	8.7	0.22	0.065

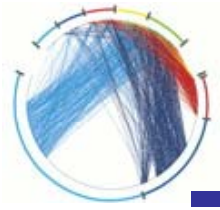


Distribution of clustering coefficient with respect to degree [C(k)]

- The C(k) distribution is supposed to capture the hierarchical nature of the network
  - when constant: no hierarchy
  - when power-law: hierarchy







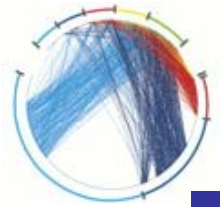
Local efficiency

- Local efficiency is a close concept to clustering coefficient
- Based on the neighborhood subgraphs
- Having a graph G, let denote the subgraph of neighbours of node i by G_i
- $E(G_i)$ is the efficiency of G_i (N_i is the number of nodes in G_i)

$$E(G_i) = \frac{1}{N_i(N_i - 1)} \sum_{k,j \in G_i} \frac{1}{d_{kj}}$$

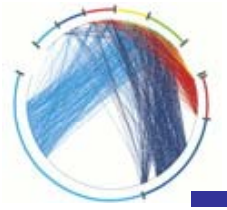
- The local efficiency is defined as

$$E_{\text{loc}} = \frac{1}{N} \sum_i E(G_i)$$



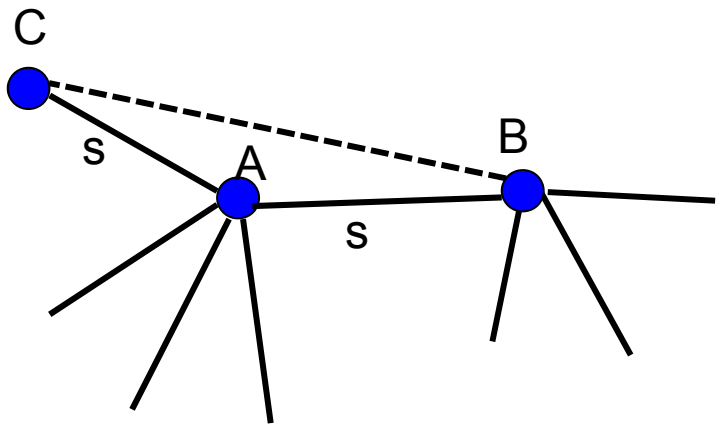
Strong and weak ties

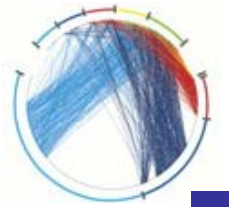
- In social networks the links are considered in two categories:
  - Strong ties: stronger links corresponding to friends
  - Weak ties: weaker links corresponding to acquaintances
- Now, think of triadic closure in terms of strong and weak ties:
  - If a node A has edges to nodes B and C, then the B-C edge is especially likely to form if A's edges to B and C are both strong ties.
  - We say that a node A violates the Strong Triadic Closure Property if it has strong ties to two other nodes B and C, and there is no edge at all (either a strong or weak tie) between B and C. We say that a node A satisfies the Strong Triadic Closure Property if it does not violate it.



Local bridges and weak ties

- If a node A in a network satisfies the Strong Triadic Closure Property and is involved in at least two strong ties, then any local bridge it is involved in must be a weak tie:
  - Consider A that satisfies the Strong Triadic Closure Property and is involved in at least two strong ties.
  - Now suppose A is involved in a local bridge to B that is a strong tie
  - This is impossible. First, since A is involved in at least two strong ties, and the edge to B is only one of them, it must have a strong tie to some other node C.
  - Now let's ask: is there an edge connecting B and C? Since the edge from A to B is a local bridge, A and B must have no friends in common, and so the B-C edge must not exist. But this contradicts Strong Triadic Closure, which says that since the A-B and A-C edges are both strong ties, the B-C edge must exist. This contradiction shows that our initial premise, the existence of a local bridge that is a strong tie, cannot hold.



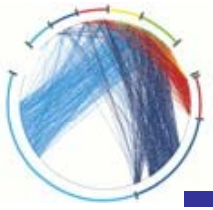


Neighborhood overlap

- The neighborhood overlap of an edge connecting A and B is:

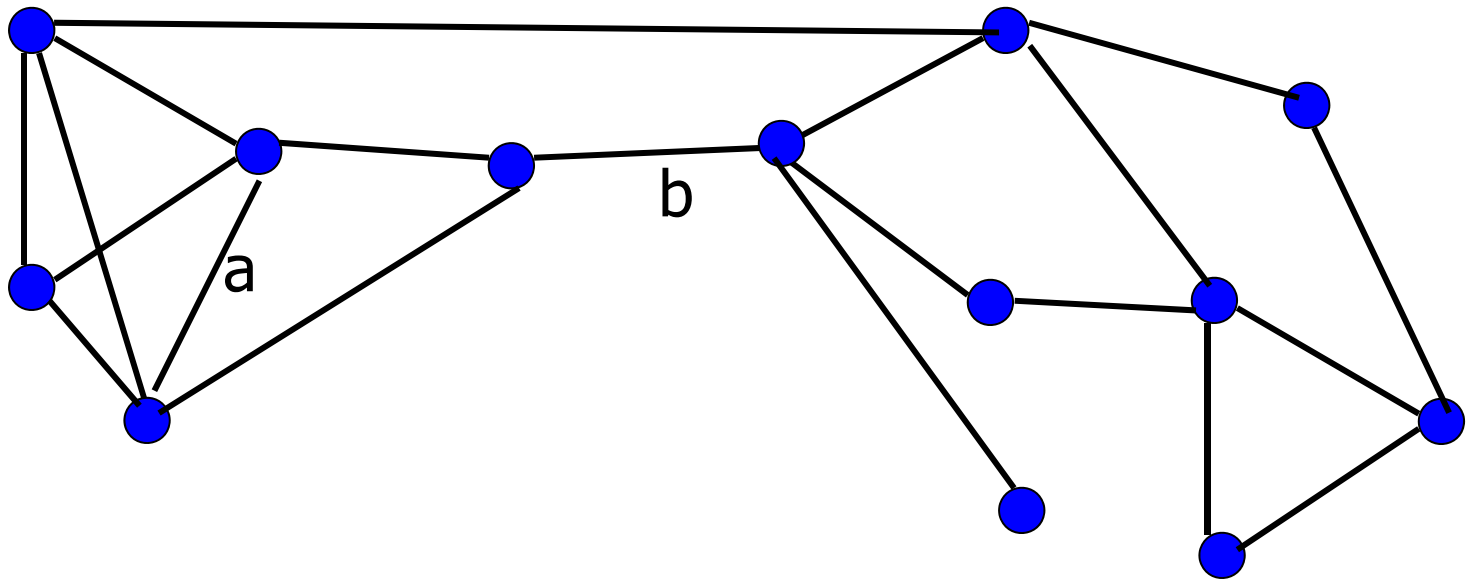
$$\frac{\text{number of nodes who are neighbors of both A and B}}{\text{number of nodes who are neighbors of at least one of A or B}}$$

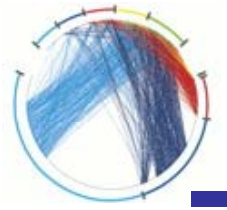
- This ratio is 0 precisely when the numerator is 0, and hence when the edge is a local bridge
- So, local bridges are the edges of neighborhood overlap 0
- We can think of edges with very small neighborhood overlap as being “almost” local bridges



Embeddedness

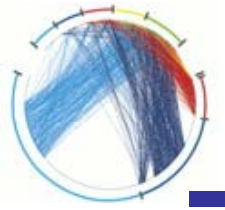
- The embeddedness of an edge is defined by the number of common neighbors of its two endpoints
  - The embeddedness of edge a is 3 and that of b is 0
- The embeddedness of an edge is equal to the numerator in the ratio in the neighborhood overlap
- Local bridges are precisely those with embeddedness of 0





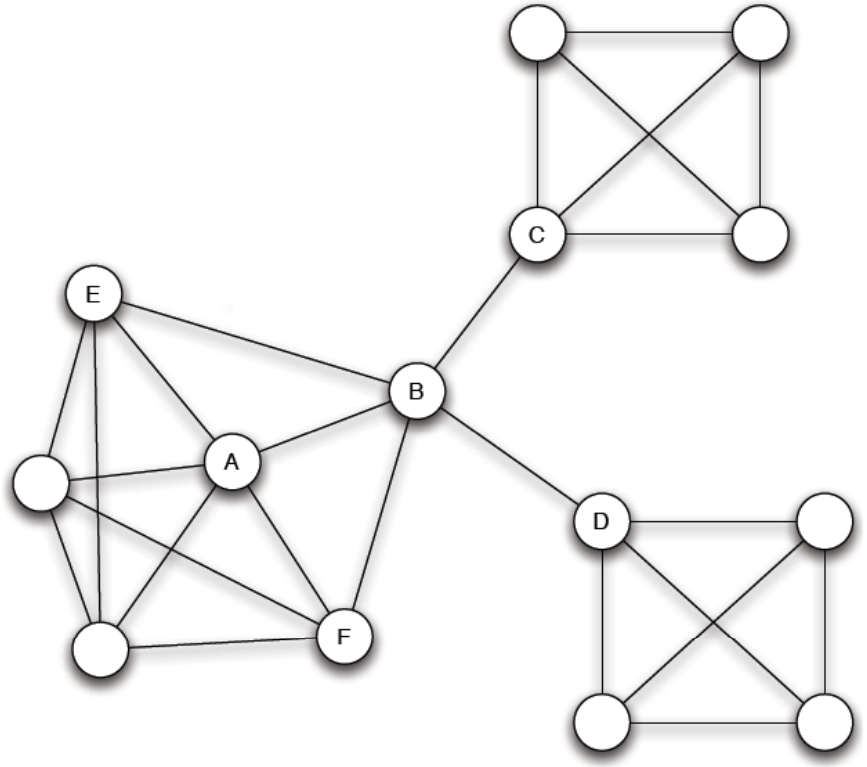
Embeddedness

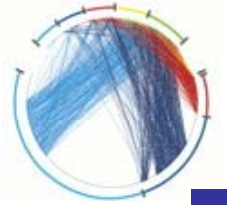
- In sociology it is argued that if two individuals are connected by an embedded edge, then this makes it easier for them to trust one another, and to have confidence in the integrity of the transactions (social, economic, or otherwise) that take place between them
- The presence of mutual friends puts the interactions between two people “on display” in a social sense, even when they are carried out in private; in the event of misbehavior by one of the two parties to the interaction, there is the potential for social sanctions and reputational consequences from their mutual friends.
- No similar kind of deterring threat exists for edges with zero embeddedness, since there is no one who knows both people involved in the interaction.



Structural hole

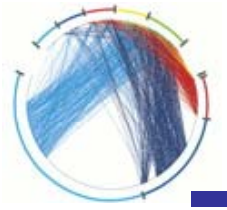
- Node B, with its multiple local bridges, spans a “structural hole” in the organization
- Structural hole is the “empty space” in the network between two sets of nodes that do not otherwise interact closely
- The argument is that B's position offers advantages in several dimensions relative to A's:
  - B has early access to information originating in multiple, non-interacting parts of the network. Any one person has a limited amount of energy they can invest in maintaining contacts across the organization, and B is investing its energy efficiently by reaching out to different groups rather than basing all her contacts in the same





Closure and bridge as social capital

- Alejandro Portes's writes “Consensus is growing in the literature that social capital stands for the ability of actors to secure benefits by virtue of membership in social networks or other social structures”
- James Coleman speaks of social capital alongside physical capital and human capital
- Pierre Bourdieu considers social capital in relation to economic capital and cultural capital
- Borgatti discusses social capital as a property of a group
- Burt discusses social capital as a tension between closure and brokerage - interactions at the interface between different groups, across structural holes
- Robert Putnam's dichotomy between bonding capital and bridging capital correspond roughly to the kinds of social capital arising respectively from connections within a tightly-knit group and from connections between such groups.



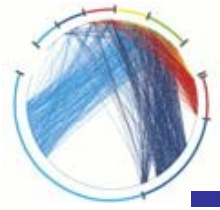
Cyclic coefficient

- It is defined in order to measure how cyclic a network is
- The local cyclic coefficient of a node i *is* defined as the average of the inverse of the sizes of the smallest cycles formed by node i and its neighbours

$$\Theta_i = \frac{2}{k_i(k_i - 1)} \sum_{k,j} \frac{1}{S_{ijk}} a_{ij} a_{ik}$$

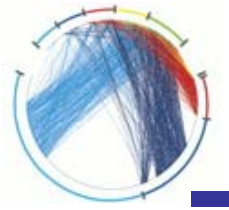
- S_{ijk} is the size of the smallest cycle that passes through nodes i, j and k
  - If j and k are connected, the smallest cycle is a triangle and Sijk = 3
  - If there is no loop passing through i, j and k, then these nodes are tree-like connected and $S_{ijk} = \infty$
- The cyclic coefficient of a network is

$$\Theta = \frac{1}{N} \sum_i \Theta_i$$



Rich-club coefficient

- In science, influential researchers of some areas tend to form collaborative groups and publish papers together
- This tendency is also observed in other real networks
- Indicates the tendency of hubs to be well connected with each other
- The above phenomenon is known as rich-club
- It can be measured by the rich-club coefficient, introduced by Zhou and Mondragon
- It is very similar to the definition of clustering coefficient



Rich-club coefficient

- Let us assume the set of nodes with degree greater than k

$$\Re(k) = \{i | k_i > k\}$$

- The rich-club coefficient of degree k is

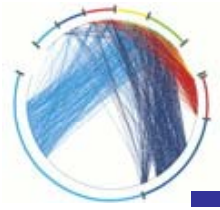
$$\phi(k) = \frac{1}{|\Re(k)||\Re(k)-1|} \sum_{i,j \in \Re(k)} a_{ij}$$

- For weighted networks:
- If the set of nodes with strength greater than s is

$$\Re^w(s) = \{i | s_i > s\}$$

- The weighted rich-club coefficient is defined as

$$\phi^w(s) = \frac{\sum_{i,j \in \Re^w(k)} w_{ij}}{\sum_{i \in \Re^w(k)} s_i}$$

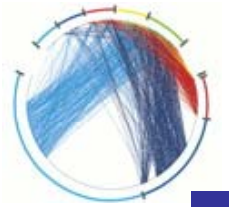


Network entropy (of degree distribution)

- Entropy is an indicator of disorder
- The more the disorder, the more the entropy
- The entropy of the degree distribution provides an average measurement of the heterogeneity of the network

$$H = -\sum_{k} P(k) \log P(k)$$

- P(k) is the probability of having degree k in the network
- The maximum value of entropy is obtained for a uniform degree distribution
- The minimum value $H_{min} = 0$ is achieved whenever all nodes have the same degree

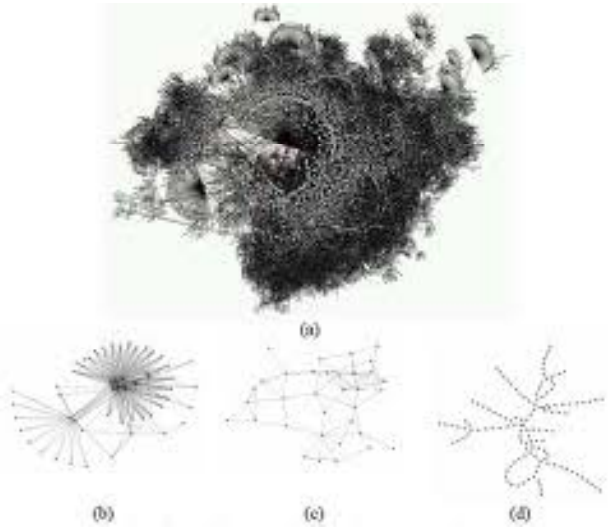


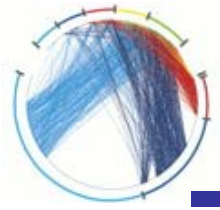
Network complexity

- Lattices and other regular structures, as well as purely random graphs, should have small values of complexity
- One possibility is the use of the computational complexity of a parallel algorithm for the generation of a network as a complexity measurement of the network model
- If there is a known parallel algorithm for the generation of the network of order O(f(N)), with f(x) a given function, then the complexity of the network model is defined as O(f(N))
- Another possibility is to associate the complexity of the network with the number of topologically non-equivalent graphs generated by splitting nodes and partitioning the edges of the original nodes among the new nodes
- One other option is the entropy of a specially defined vertex-vertex edge correlation matrix

Fractal dimensionality

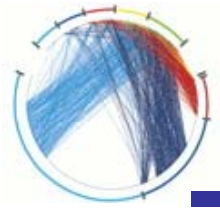
- Fractals are objects or quantities that display self-similarity (or self-affinity) in all scales
- It has been shown that real complex networks may consist of self-repeating patterns on all length scales.
- In order to measure the fractal dimension of complex networks, a box counting (BC) method and a cluster growing (CG) method has been proposed





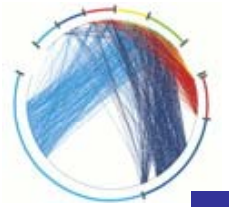
Fractal dimensionality (BC)

- In the BC method, the network is covered with N_B boxes
- All nodes in each of the boxes are connected by a minimum distance smaller than l_B
- N_B and l_B are found to be related by
- $N_B \sim l_B^{-d}$
- d is the fractal box dimension of the network



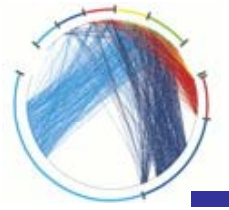
Fractal dimensionality (CG)

- In the CG method, a seed node is chosen at random and a cluster is formed by nodes distant at most l from the seed
- The above process is repeated many times
- The average mass of resulting clusters is calculated as a function of l as $< M_C > \sim l^d$
- d is the fractal cluster dimension of the network
- The average mass of resulting clusters $< M_C >$ is defined as the number of nodes in the cluster



Edge reciprocity

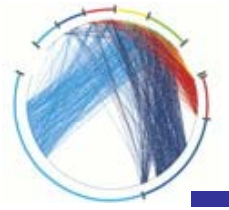
- For directed networks, it is often interesting to know if their edges are reciprocal, i.e. if node i is linked to node j, is node j also linked to node i?
- Such information may help to obtain a better characterization of the network
- It can be used to test network models against real networks
- It gives indication of how much information is lost when the direction of the edges is discarded (e.g. for the computation of some measurements that only apply to undirected networks).
- A standard way to obtain information about reciprocity is to compute the fraction of bilateral edges



Edge reciprocity

$$\rho = \frac{\sum_{i,j} a_{ij} a_{ji}}{M}$$

- M is the total number of edges
- ρ is an indicator of edge reciprocity in the network
- The above measurement has a problem that is its value is only relevant with respect to a random version of the network
- Networks with higher connectivity tend to have a higher number of reciprocal edges due exclusively to random factors.

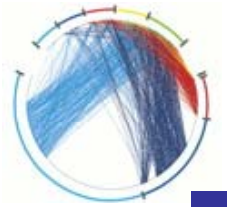


Edge reciprocity

- To overcome this problem it has been proposed to use the correlation coefficient of the adjacency matrix:

$$\mathrm{P} = \frac{\sum_{i,j}\left(a_{ij}-a\right)\left(a_{ji}-a\right)}{\sum_{i,j}\left(a_{ji}-a\right)^2}$$

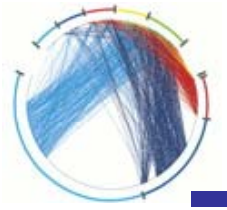
- a is the mean value of the elements of the adjacency matrix
- The above relation is simplified to $\mathrm{P} = \frac{\rho - a}{1-a}$
- This value is an absolute quantity:
  - Values of ρ greater than zero imply larger reciprocity than the random version (reciprocal networks)
  - Values below zero imply smaller reciprocity than a random network (antireciprocal networks)
- This concept can be easily extended to weighted networks by substituting a_{ij} for w_{ij} in the above expressions



Matching index

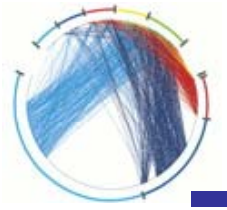
- It might be interesting to quantify the similarity between the connectivity of the two nodes adjacent to an edge
- To this end, a matching index can be assigned to each edge
  - A low value of the matching index identifies an edge that connects two dissimilar regions of the network, thus possibly playing an important role as a shortcut between distant network regions
- The matching index of edge between nodes i and j is computed as the number of matching connections of nodes i and j (i.e. connections to the same other node k), divided by the total number of connections of both nodes (excluding connections between i and j),— neighbourhood overlap

$$\mu_{ij} = \frac{\sum_{k \neq i,j} a_{ik} a_{jk}}{\sum_{k \neq j} a_{ik} + \sum_{k \neq i} a_{jk}}$$



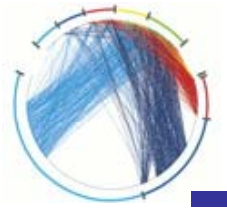
Matching index

- For directed networks, matching connections are only those in the same direction
- In such cases, incoming and outgoing connections of nodes i and j should be considered separately
- The concept of matching index has also been adapted to apply to consider all the immediate neighbours of a node, instead of a single edge



Homophily in social networks

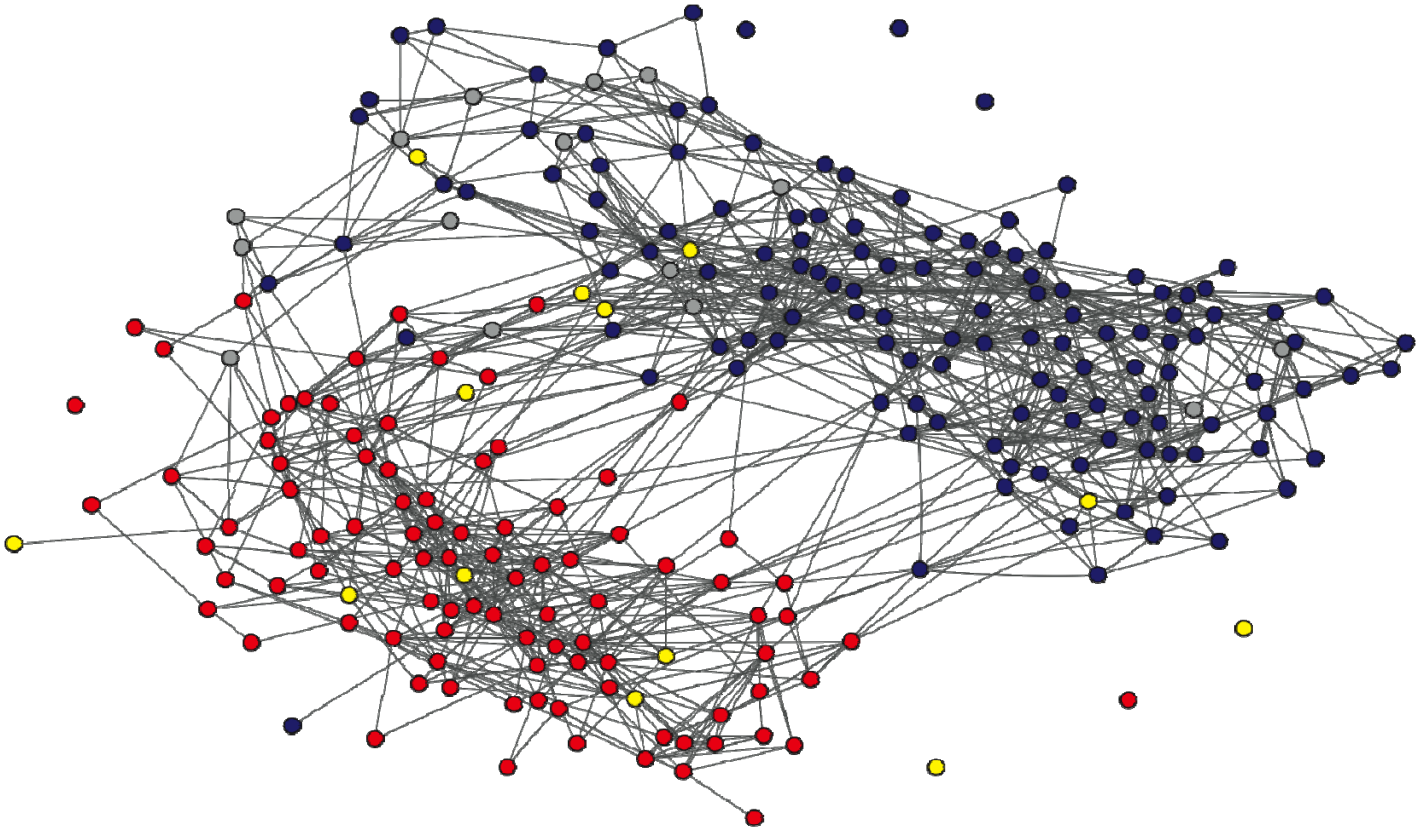
- Homophily: tendency of people to connect to other people similar to themeselves
- Certainly not a new observation: Aristoteles: “people love those who are like themselves”, Plato: “similarity begins friendship”
- Early studies: school friendships (1929). Homophily in play is observed in race, gender, age, intelligence, attitudes.
- Mid-century: strong interest in homophily driven by school segregation and peer effects on behaviour.
- From ‘70s: application of statiscal inference allows to study large networks.



Homophily in social networks

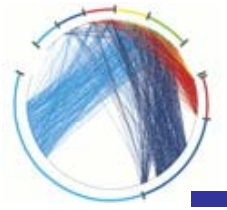
- Hypothesizing intrinsic mechanisms:
  - Individuals B and C have a common friend A
  - So, there are increased opportunities and sources of trust on which to base their interactions,
  - As a results, A will also have incentives to facilitate their friendship.
- Since we know that A-B and A-C friendships already exist, the principle of homophily suggests that B and C are each likely to be similar to A in a number of dimensions
- As a result, based purely on this similarity, there is an elevated chance that a B-C friendship will form; and this is true even if neither of them is aware that the other one knows A.

Example



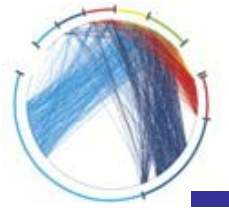
U.S. Midwest Urban school. Red = Black, Blue = White, Yellow = Hispanic, Grey Asian. A link means a nominated friendship.

Source: Add Health Dataset and Currarini-Jackson-Pin (2009).



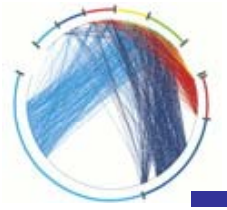
Measuring Homophily

- Given a particular characteristic of interest (like race, or age), is there a simple test we can apply to a network in order to estimate whether it exhibits homophily according to this characteristic?
- This measure is like other graph metrics measuring a similarity in the connections



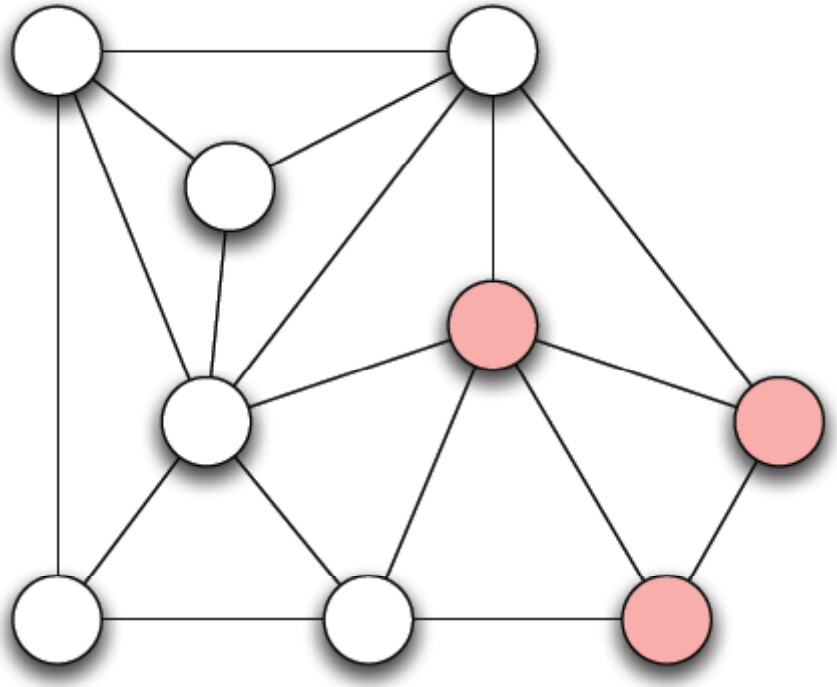
Measuring Homophily

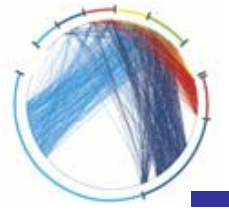
- Consider a network with
  - p fraction of all individuals male
  - q fraction of all individuals female
- If we independently assign each node the gender male with probability p and the gender female with probability q, then both ends of the edge will be male with probability p^2, and both ends will be female with probability q^2.
- On the other hand, if one of the ends is male and another is female, then we have a cross-gender edge, so this happens with probability 2pq
- **Homophily Test: If the fraction of cross-gender edges is significantly less than 2pq, then there is evidence for homophily.**



Measuring Homophily

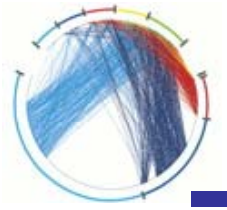
- Colors for different genders
- p = 2/3, q = 1/3
- 18 edges of which 5 are cross-gender = 5/18
- 2pq = 4/9 = 8/18
- With no homophily, one should expect to see 8 cross-gender edges rather than 5
- This example shows evidence of homophily
- Note however that these values should be **significantly different**





Measuring Homophily

- When the characteristics under study is more than two (e.g. red, white, black and yellow), we say that an edge is heterogeneous if it connects two nodes that are different according to the characteristic in question
- We then ask how the number of heterogeneous edges compares to what we'd see if we were to randomly assign values for the characteristic to all nodes in the network
- In this way, even a network in which the nodes are classified into many groups can be tested for homophily using the same underlying comparison to a baseline of random mixing.



Measuring Homophily (Formal)

Let us Denote:
by w_i the relative size of group i.
by s_i the per capita number of intra-group ties for group "i"
by t_i the total per capita number of ties for group "i"

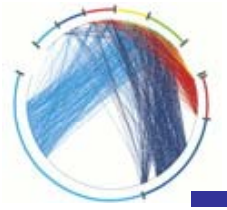
Homohily Index: $H_i = s_i / t_i$

If things happened randomly, we would have $H_i = w_i$ for each group.

If $H_i > w_i$, group i's behaviour exhibits mode homophily than under random assortment. This may be due to mainly two factors:

- group members correct the random assortment through behaviour and choice;
- the share w_i does not reflect opportunties of encounter, that follow some process that differs from a uniform random assortment.

This excess homophily is referred to as "inbreeding homophily".



Measuring Homophily (Formal)

The simple difference ($H_i - w_i$) is not a good index if we wish to compare the amount of inbreeding homophily of different groups.

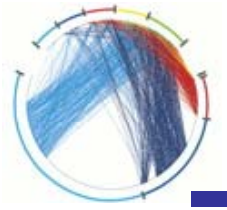
This because very large groups have very small "potential" inbreeding homophily (measured by the term ($1 - w_i$)).

The sociologist Coleman (1958) has proposed the following normalized measure:

Inbreeding Homohily Index: $IH_i = (H_i - w_i) / (1 - w_i)$

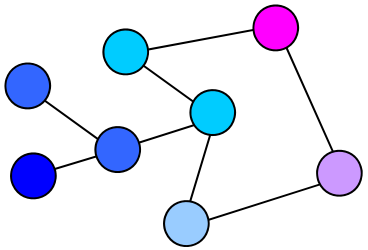
The IH index is positive when a group has inbreeding homophily, and zero in case of pure baseline homophily.

Understanding the sources of inbreeding homophily and the generative process of observed social networks are the main agenda in the sociological and economics literature on this subject.

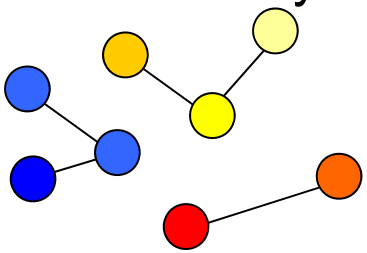


Mechanisms underlying Homophily

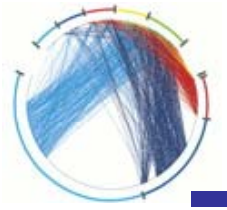
- **Selection:** people select to show interact
- **Social influence:** people may modify their behaviors to bring them more closely into alignment with the behaviors of their friends
- Often, the interplay between selection and social influence governs the formation of social networks within communities
- Both processes contribute to homophily, but
  - Social influence leads to community-wide homogeneity
  - Selection leads to fragmentation of the community



homogeneity



fragmentation



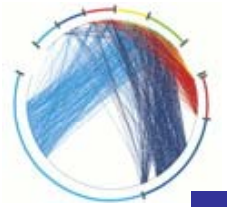
Segregation

- The notion of segregation is very much related to that of homophily.
- While homophily measures refer usually to individuals or groups, measures of segregation refer to the system where groups interact.
- One such measure compares the odds of at tie being intra-groups with the odds of being inter-group.

	Intra-Group Pair	Inter-Group Pair
Tie	A	C
No Tie	B	D

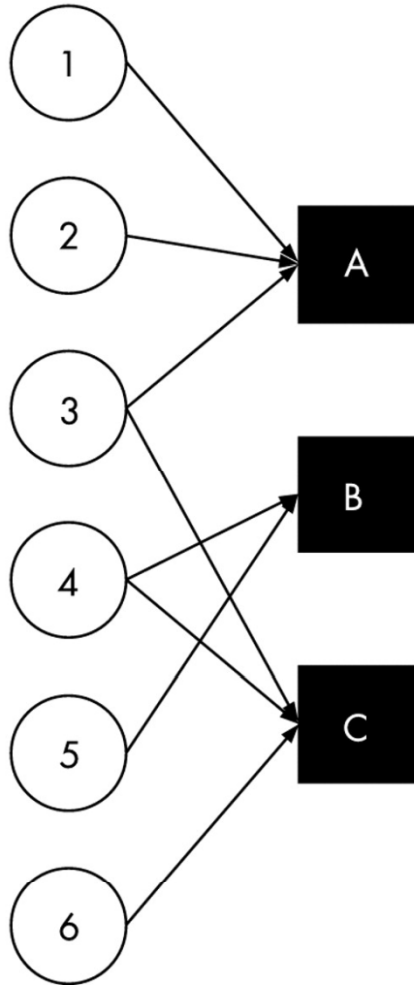
$$\alpha=(A/B)/(C/D)$$

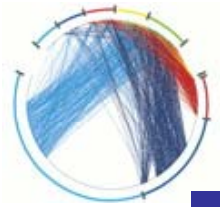
When α = 1 there is no segregation
The larger α the more segregation



Affiliation

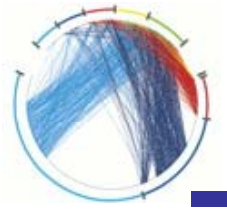
- **Foci (focus):** activities a person involved
- **Affiliation networks:** we can represent the participation of a set of people in a set of foci using a graph as follow:
- We will have a node for each person, and a node for each focus, and we will connect person A to focus X by an edge if A participates in X.
- As we can see, these graphs are bipartite
- A nice widely studied affiliation network is the composition of boards of directors of major corporations





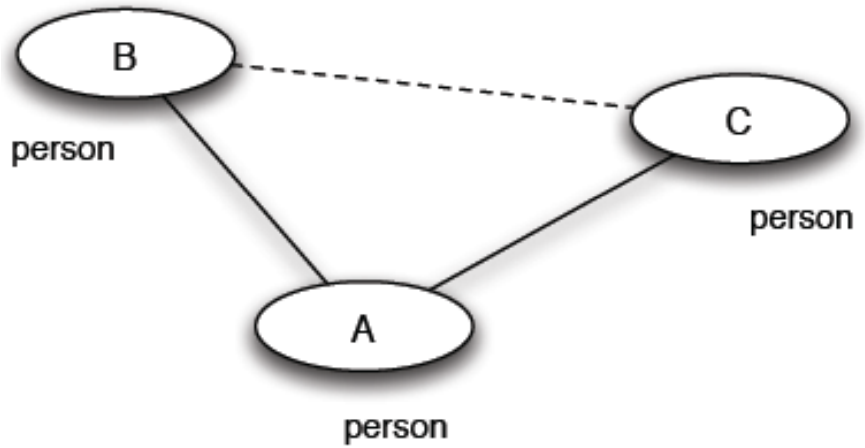
Co-evolution of social and affiliation networks

- Both social and affiliation networks change over time
  - New friendships are formed
  - New members are engaged in the foci
  - …
- There is sort of co-evolution
  - if two people participate in a shared focus, this provides them with an opportunity to become friends
  - if two people are friends, they can influence each other's choice of foci
- A network with people and foci as nodes and two kind of edges as
  1. Connecting two people and indicating friendship
  2. connecting a person to a focus and indicating the participation of the person in the focus
- We call these network social-affiliation network

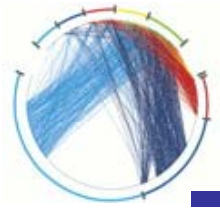


Social-affiliation networks

- At least three different mechanisms for closure
  1. Triadic closure (we have seen this item): If the links A-B and A-C exist and A, B, and C each represent a person, then the formation of the link between B and C is triadic closure

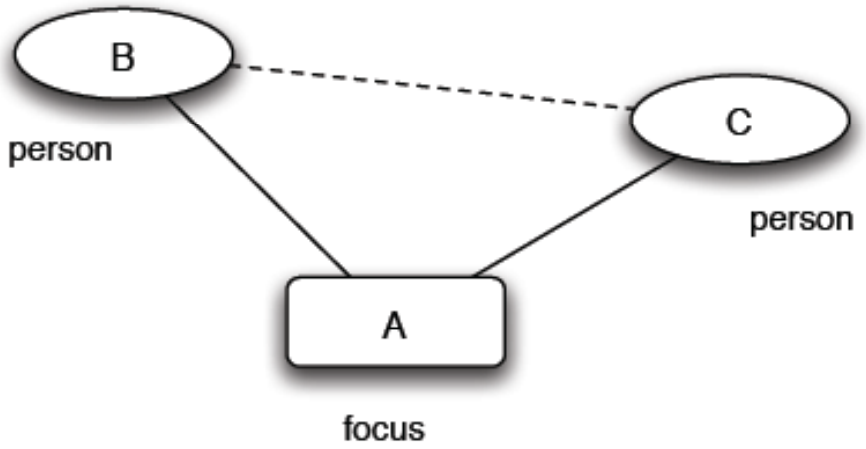


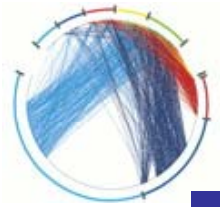
  2. Focal closure
  3. Membership closure



Focal closure

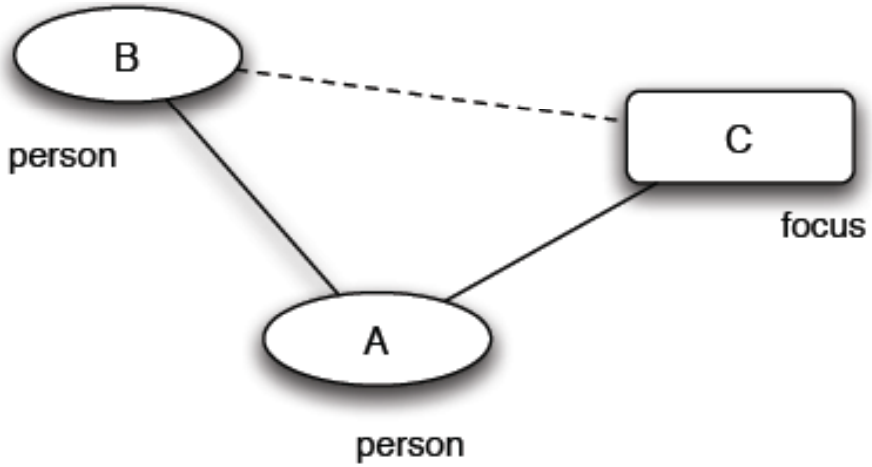
- B and C represent people, but A represents a focus
- It is the tendency of two people to form a link when they have a focus in common.
- This is an aspect of the more general principle of selection, forming links to others who share characteristics with you.
- This process has been called focal-closure

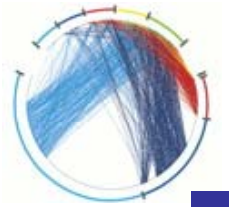




Membership closure

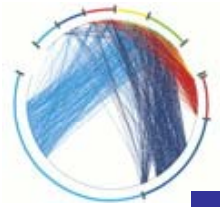
- If A and B are people, and C is a focus
- B takes part in a focus that her friend A is already involved in
- B's behavior comes into closer alignment with that of her friend A
- we will refer to this kind of link formation as membership closure



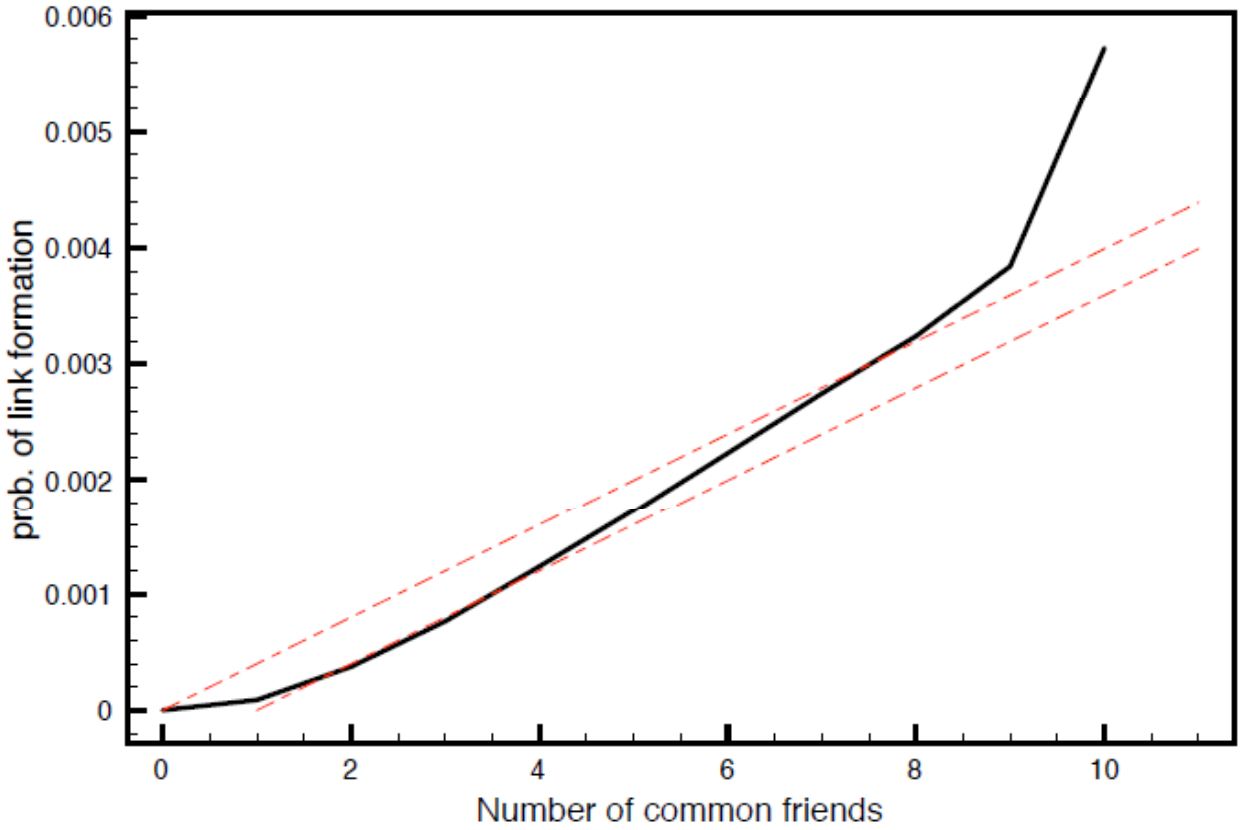


How to track link formation in data

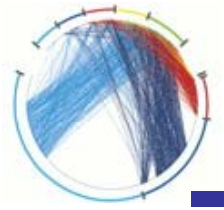
1. If A and B are people, and C is a focus, we take two snapshots of the network at different times.
2. For each k, we identify all pairs of nodes who have exactly k friends in common in the first snapshot.
3. We define T(k) to be fraction of these pairs that have formed an edge by the time of the second snapshot. This is our empirical estimate for the probability that a link will form between two people with k friends in common.
4. We plot T(k) as a function of k to illustrate the effect of common friends on the formation of links.



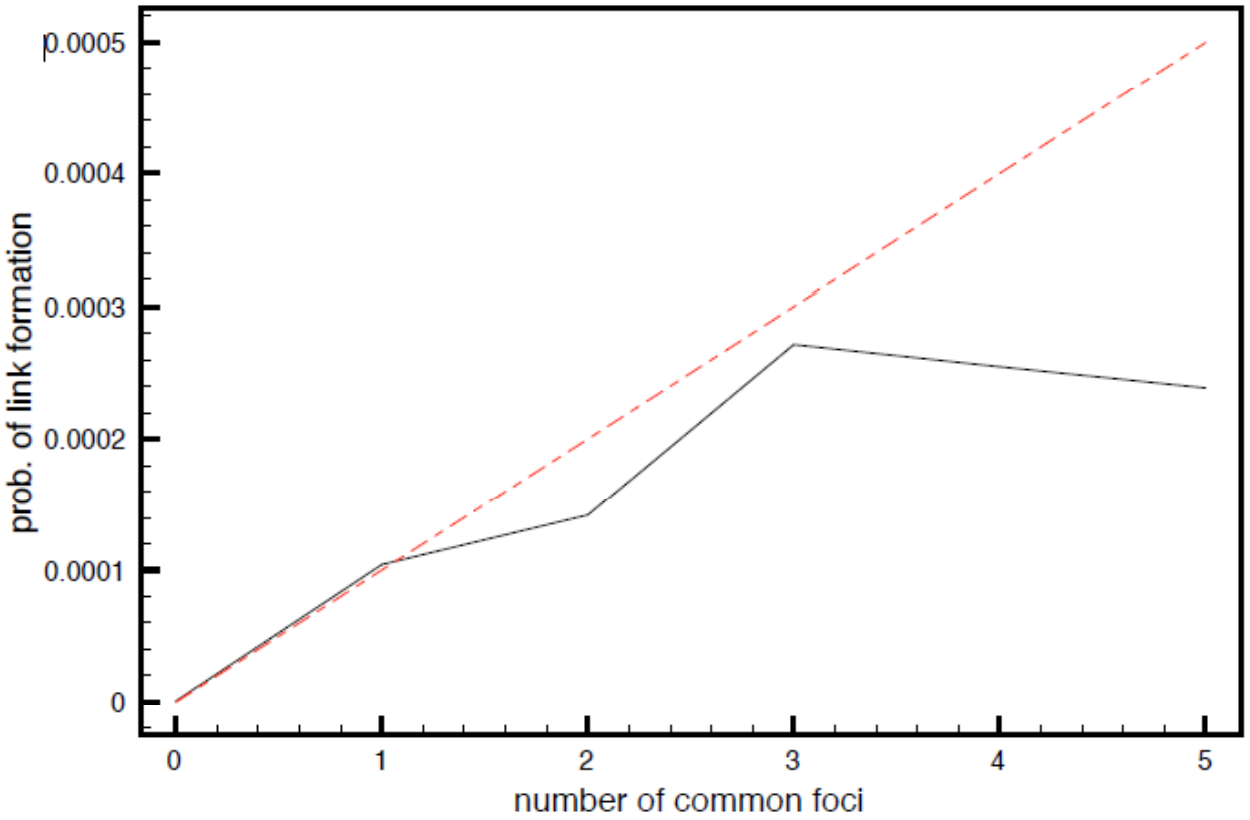
Example for triadic closure



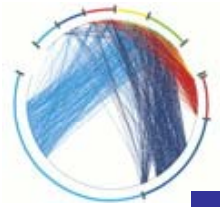
Quantifying the effects of triadic closure in an e-mail dataset (Kossinets and Watts. Empirical analysis of an evolving social network, Science, 311:88-90, 2006.) [255]. The curve determined from the data is shown in the solid black line; the dotted curves show a comparison to probabilities computed according to two simple baseline models in which common friends provide independent probabilities of link formation.



Example for focal closure



Quantifying the effects of focal closure in an e-mail dataset (Kossinets and Watts. Empirical analysis of an evolving social network, Science, 311:88-90, 2006.). Again, the curve determined from the data is shown in the solid black line, while the dotted curve provides a comparison to a simple baseline.



Readings

- L. da F. Costa, F. A. Rodrigues, G. Travieso, and P. R. Villas Boas. Characterization of complex networks: A survey of measurements. Advances in Physics, 56(1):167 – 242, 2007.
- "Networks, Crowds, and Markets" by Easley and Kleinberg (Chapters 3 and 4)