CMSC5733 Social Computing

Irwin King

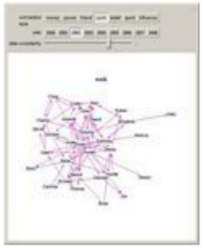
The Chinese University of Hong Kong

king@cse.cuhk.edu.hk

Outline

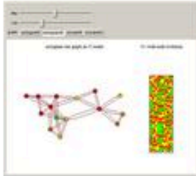
- Regular Networks
  - Diameter, Centrality, and Average Path Length
  - Type of Networks
- Random Networks
  - Generation
  - Degree Distribution, Entropy and Properties
  - Weak Ties
  - Randomization of Regular Networks
- Small-World Networks
  - Generation
  - Properties
  - Phase Transition
  - Weak Ties

http://demonstrations.wolfram.com

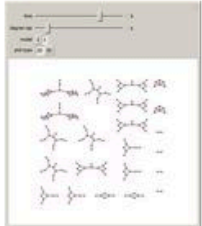


Social Networking

Social Golfer Problem

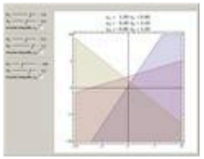


Four-Color Outer Median Cellular Automata on Graphs

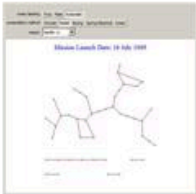


Self-Replicating Graphs

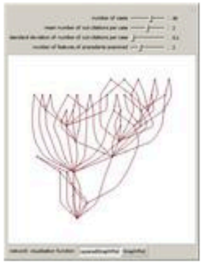
Triad Census on Random Graphs



Graph of Inequalities



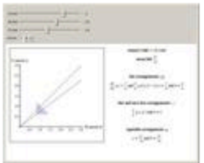
Networks of Space Flights by American Pre-Shuttle Astronauts



Synthetic Legal Precedent Structures: Feature Distance

Genealogy Graphs from XML

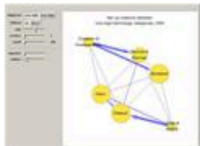
The Sensitivity of Page Rank to Connection Errors



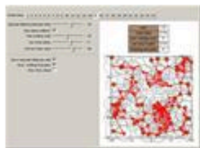
Fair, Equitable Compensation Arrangements

Communities of Nations Bridged by Language Similarity

U.S. Net and Total Patent Citation Flows, 1963-2002



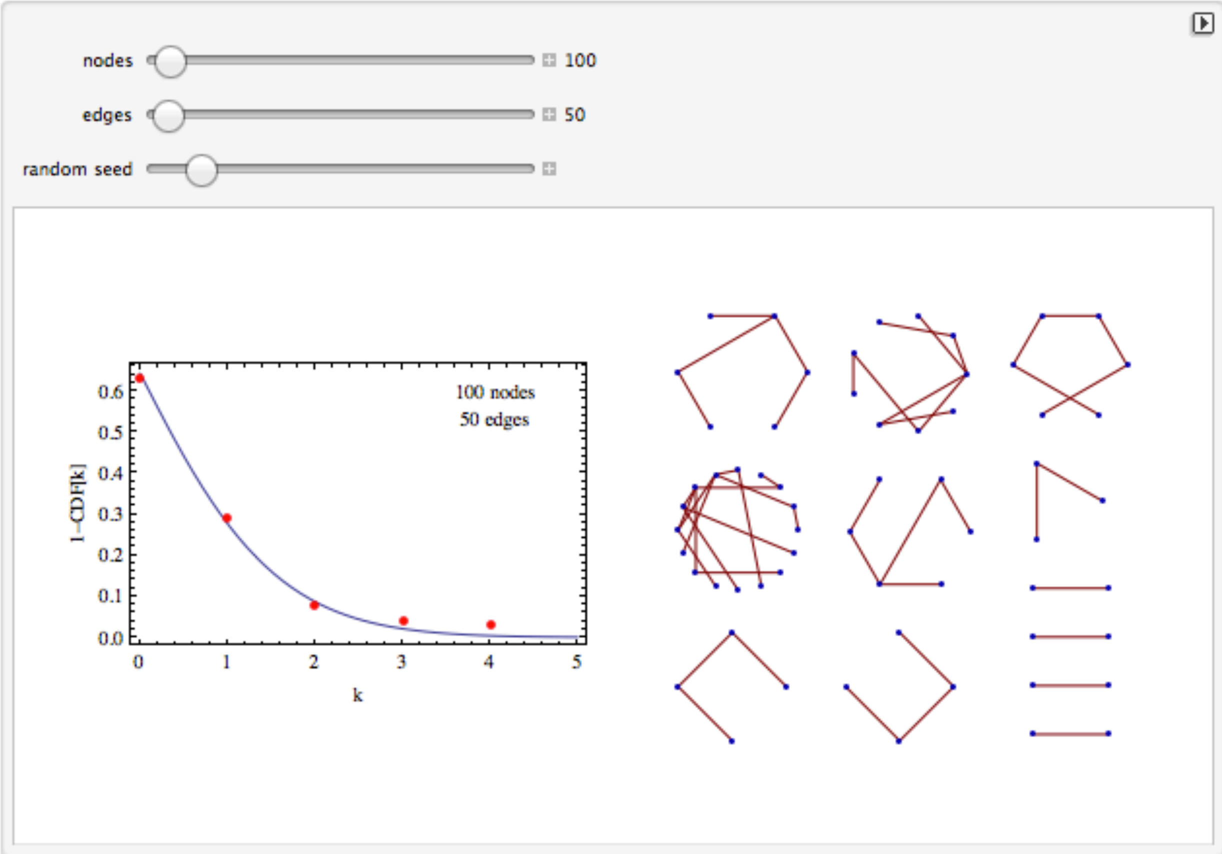
U.S. Net Patent Citation Flows, 1963-2002



Fuel Break

http://demonstrations.wolfram.com/DegreeDistributionOnARandomNetwork/

Degree Distribution on a Random Network



An Erdös-Rényi random graph is one chosen at random from all the graphs with a given number of nodes (n) and edges (e). The degree of a vertex in such a graph follows a Poisson distribution with mean $2e/n$. The blue curve in the left plot is a continuous approximation of $1-f(k)$, where $f(k)$ is the cumulative distribution function of a Poisson distribution with parameter $\lambda = 2e/n$. The red dots show the fraction of nodes of a particular graph with degree k.

REGULAR NETWORKS

Link Efficiency

- Link Efficiency
  - The tradeoff between number of links and number of hops in the average path length of a network:

$$E(G) = \frac{m - avg\_path\_length(G)}{m}$$

where *m* is the number of links in *G*

  - Let *t* be the total number of paths and $r_{i,j}$ the length of the direct path between node v_i and v_j :

$$\text{avg\_path\_length} = \sum_i \sum_j \frac{r_{i,j}}{t}$$

  - A network is *scalable* if link efficiency approaches 100% as network size *n* approaches infinity

TABLE 1 Link Efficiency of Several Network Classes, $n \gg 1$

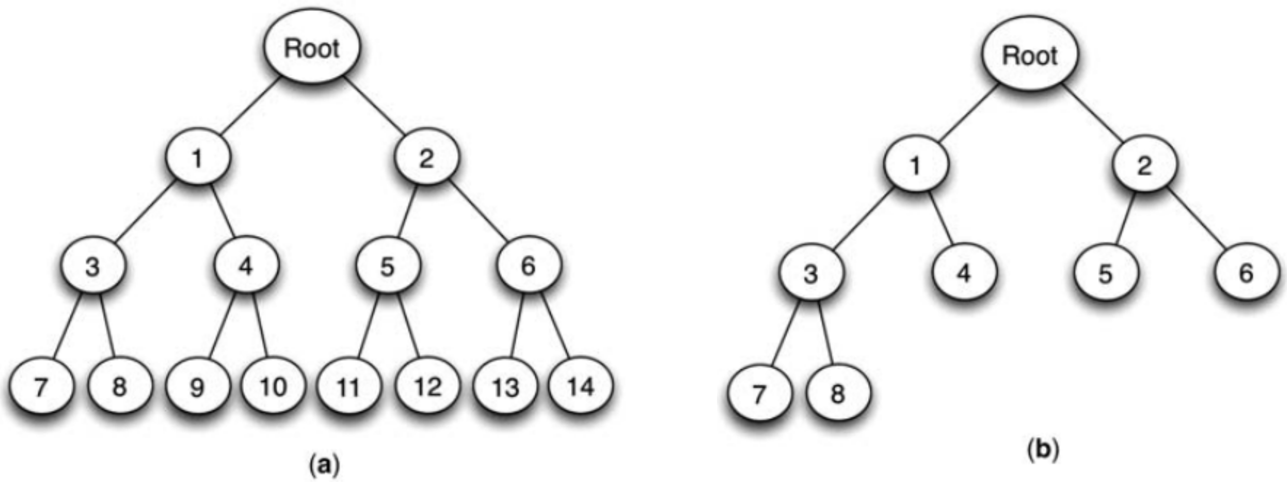
Network Class	Efficiency	Example
Line	$\frac{2n-4}{3(n-1)}$	Asymptotic to $\frac{2}{3}$
Ring	$\frac{3n-1}{4n}$	Asymptotic to $\frac{3}{4}$
Binary tree	$1-\frac{2\log_2(n+1)-6}{n-1}$	$n=127$, $m=126$, $E=93.4\%$
Toroid	$1-\frac{1}{4\sqrt{n}}$	$n=100$, $m=200$, $E=97.5\%$
Random	98.31%	$n=100$, $m=200$, avg_path_length = 3.38
Hypercube	$1-\frac{1}{n-1}$	$n=128$, $m=448$, $E=99.2\%$
Complete	~ 1.0	$m=n\frac{n-1}{2}$, avg_path_length = 1

Binary Tree Network

- A line graph is not link-efficient
  - The number of links grows as fast as the number of hops in its average path length
- The *binary tree* is more link-efficient
- A binary tree is defined recursively
  - The root node, has degree 2 and connects two subtrees, which in turn connect to two more subtrees, and so forth
  - This recursion ends with a set of nodes called the leaf nodes, which have degree 1
  - As it grows, its average path length grows much slower than its number of links

Binary Tree Network

- Balanced binary tree
  - A balanced binary tree contains *k* levels and exactly $2^k - 1$ nodes, m = (n – 1) links, for *k* =1,2, ...
- Unbalanced binary tree
  - An unbalanced binary tree contains less than 2^k - 1 nodes



Properties

- Center
  - The root node with radius r = k – 1
  - the leaf nodes lie at the extreme diameter, which is D = 2(k – 1) hops
- Diameter
  - Grows logarithmic with size n because $k = O(\log_2(n))$
- Average path length
  - Also grows logarithmically, is proportional to its diameter

Entropy of Binary Tree Network

- A balanced binary tree network is regular, but its entropy is not zero
- Entropy is a function of the degree sequence distribution
- The degree sequence distribution for the binary tree is g’ = [53%; 7%; 40%]
- Entropy is calculated as I(G) = - Σ p $\log_2$ p

$$p_1 = \frac{n/2}{n} = \frac{n}{2n} = \frac{1}{2} \qquad \text{leaf node frequency}$$

$$p_2 = \frac{1}{n} \qquad \text{root node frequency}$$

$$p_3 = \frac{n - (n/2) - 1}{n} = \frac{n-2}{2n} \qquad \text{internal node frequency}$$

Entropy of Binary Tree Network

$$I(\text{balanced binary tree}) = -\left[\frac{1}{2}\log_2\frac{1}{2} + \frac{1}{n}\log_2\frac{1}{n} + \frac{n-2}{2n}\log_2\frac{n-2}{2n}\right]$$

$$= \frac{1}{2} + \frac{1}{n}\log_2 n + \left(\frac{n-2}{2n}\log_2\frac{2n}{n-2}\right)$$

$$I(\text{Balanced binary tree}) = 1 + \frac{\log_2(n)}{n}; \quad n \gg 1$$

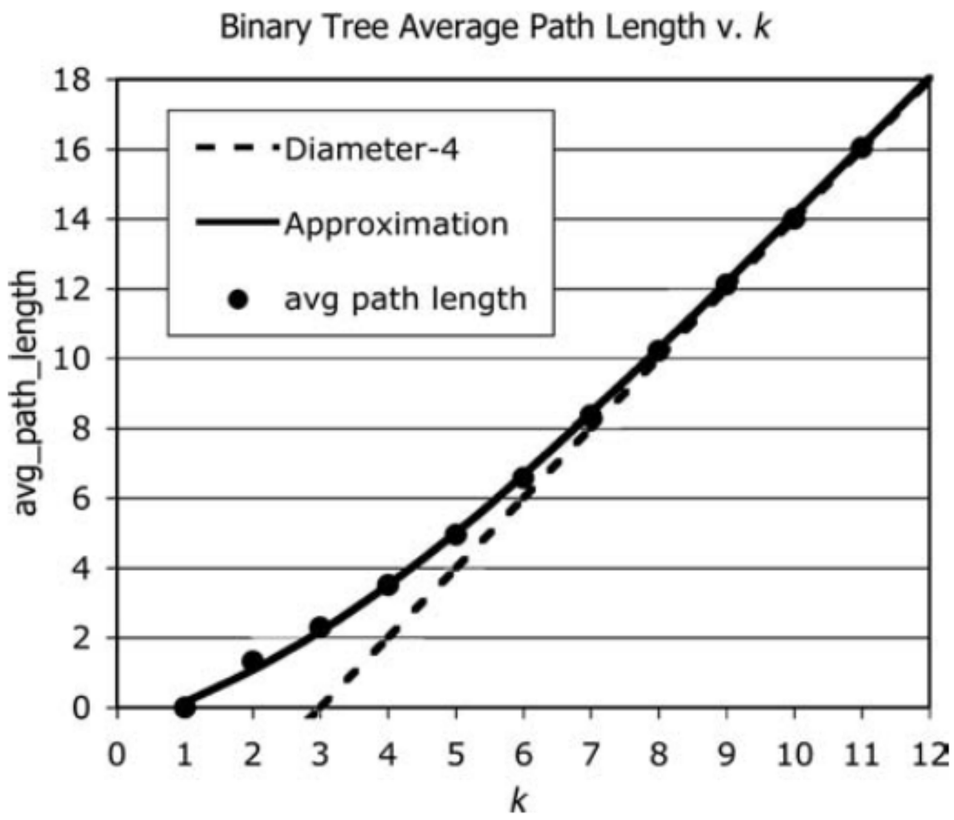


Figure Path length and $(D - 4)$ versus level k for a balanced binary tree with $n = 2^k - 1$ nodes, $m = n - 1$ links, and diameter $= D = 2(k - 1)$.

Average path length and $(D - 4)$ merge for high values of k. Thus, average path length is asymptotic to $(D - 4)$:

$$\text{avg\_path\_length(balanced binary tree)} = (D - 4);\ k \gg 1$$

$$D = 2(k - 1), \text{ so avg\_path\_length} = 2k - 6 = 2\log_2(n + 1) - 6$$

- For smaller values of *k*, *say*, *k* < *9*, the approximation breaks down
  - The nonlinear portion of the approximation diminishes exponentially as *k* increases — reaching zero as (D – 4) dominates:

$$\text{avg\_path\_length} = (D - 4) + \frac{A}{1 + \exp(B\,\text{k})}$$

where A =10.67, B = 0.45 gives the best fit.

  - Substituting *D* = 2(*k* – 1) and *k* =$\log_2(n+1)$

$$\text{avg\_path\_length} = 2\log_2(n+1) - 6 + \frac{10.67}{1 + \exp(0.45\log_2(n+1))}$$

Link Efficiency

- A balanced binary tree has m = n – 1 links
- Link efficiency of a "large" balanced binary tree is:

$$E(\text{balanced binary tree}) = 1 - \frac{D-4}{m} = 1 - \frac{(2k-1)-4}{n-1}; \quad k > 9$$

$$E = 1 - \frac{2\log_2(n+1) - 6}{n-1}, \text{ because } k = \log_2(n+1)$$

- Assuming $k >> 1$

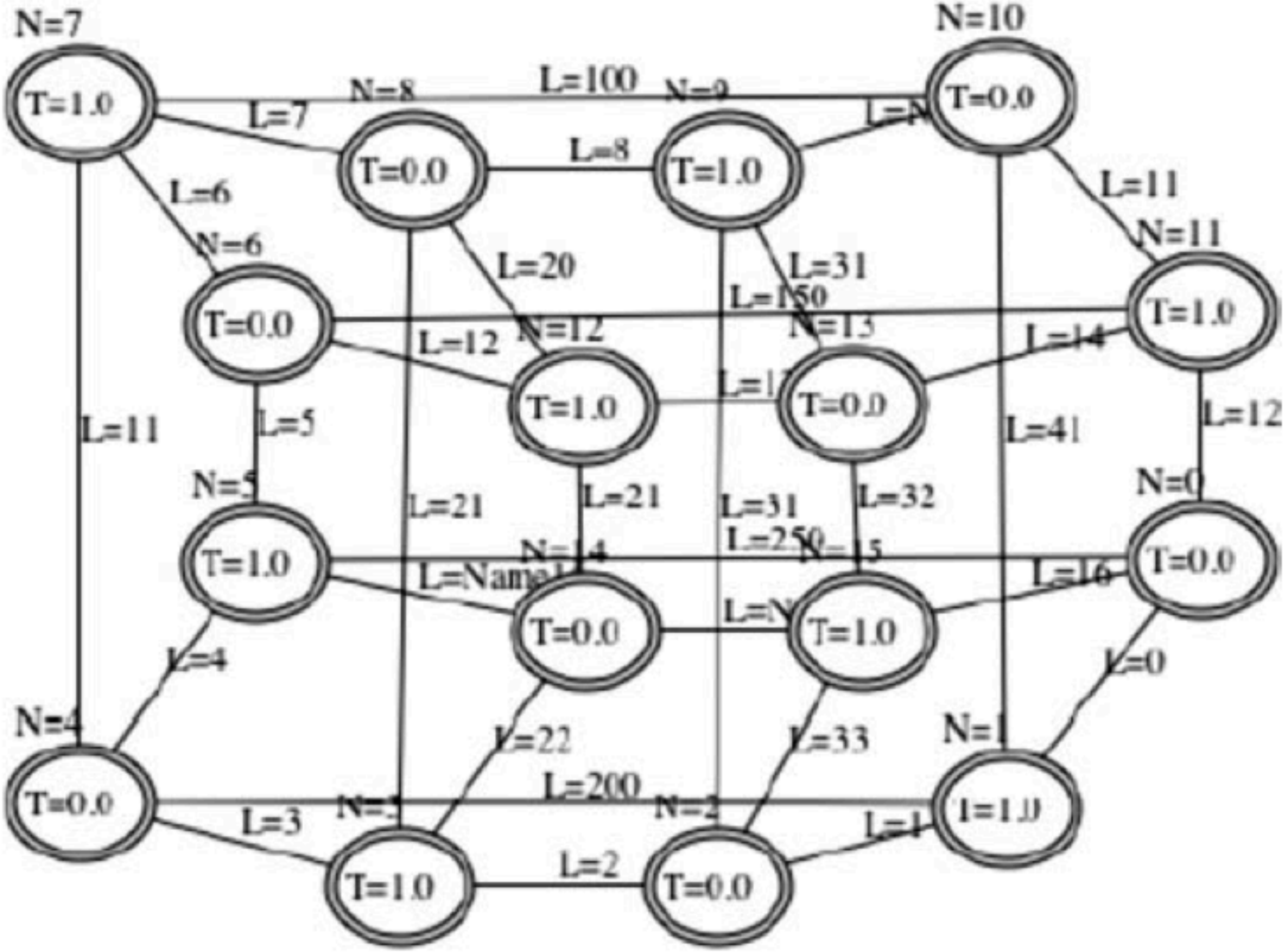
$$E(\text{balanced binary tree}) = 1 - \frac{2\log_2(n)}{n}; \quad k > 9$$

- Binary tree link efficiency approaches 100%, as n grows without bound

Toroidal Network

- Problem with a binary tree is the logarithmic growth in the distance from its root node to its leaf nodes
- Can we shorten paths without adding more links?
- How about a grid-like structure with wrap-around?

A Toroidal Network



Toroidal Networks

- How many links are there?
- What is the entropy of a toroidal network?
- The mapping function

$$f : v_i \sim v_{(i+1) \bmod \sqrt{n}}; \quad v_i \sim v_{(i+\sqrt{n}) \bmod n}$$

- The Average Path Length

TABLE 3.2 Results of Path Matrix Analysis of Toroidal Networks

Size, n	Toroid	Row Sum	Factored Row Sum	Average Path Length
4	2×2	4	2*2	$\frac{4}{3} = 1.33$
9	3×3	12	3*4	$\frac{12}{8} = 1.50$
16	4×4	32	4*8	$\frac{32}{15} = 2.13$
25	5×5	60	5*12	$\frac{60}{24} = 2.50$
36	6×6	108	6*18	$\frac{108}{35} = 3.09$
49	7×7	168	7*24	$\frac{168}{48} = 3.50$

Hypercube Networks

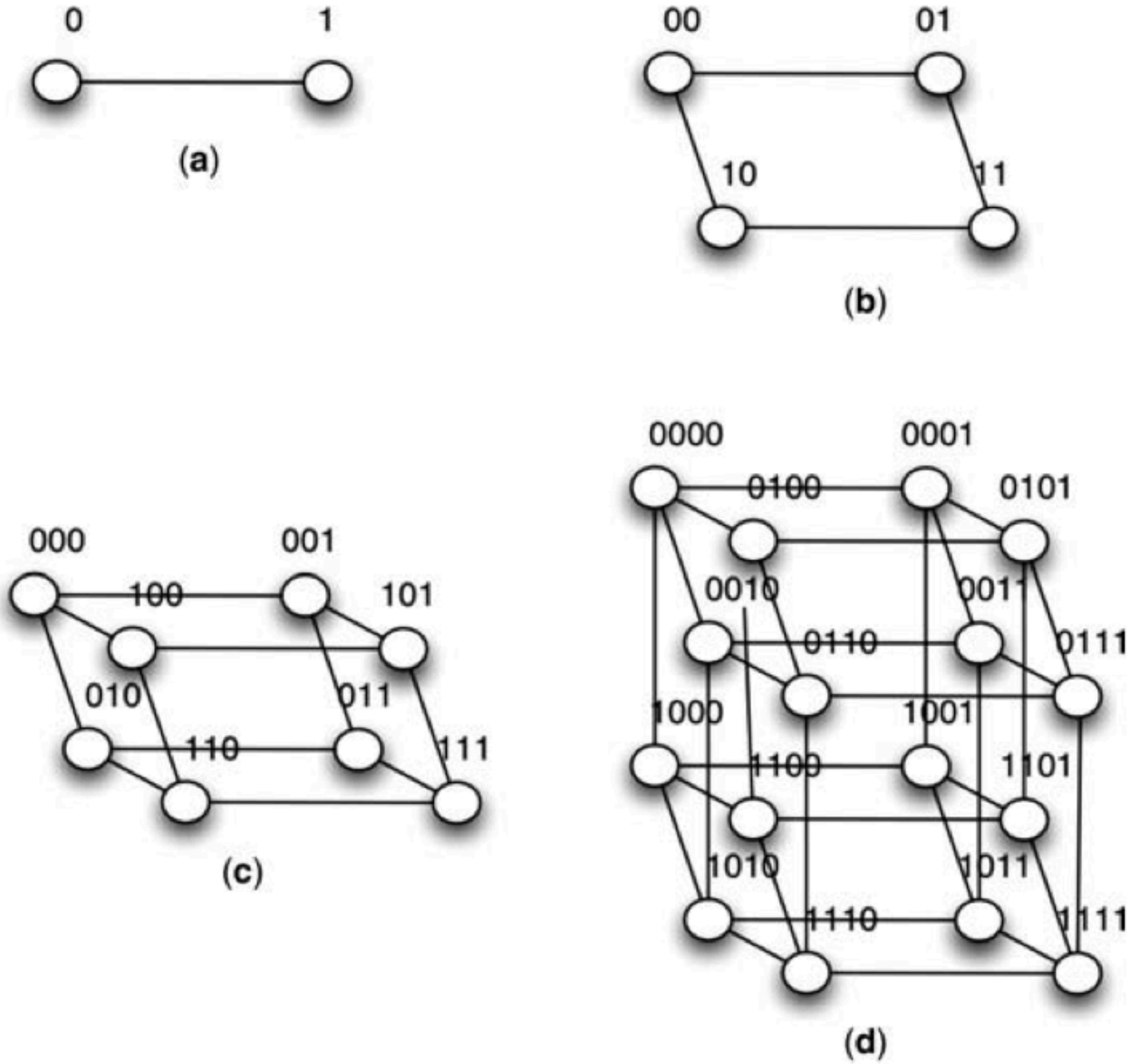


TABLE 3.3 Hamming Distance between Binary Numbers[a]

Hamming Distance	000	001	010	011	100	101	110	111
000	0	1	1	2	1	2	2	3
001	1	0	2	1	2	1	3	2
010	1	2	0	1	2	3	1	2
011	2	1	1	0	3	2	2	1
100	1	2	2	3	0	1	1	2

[a]Columns, $0 \cdots 7$; rows, $0 \cdots 4$.

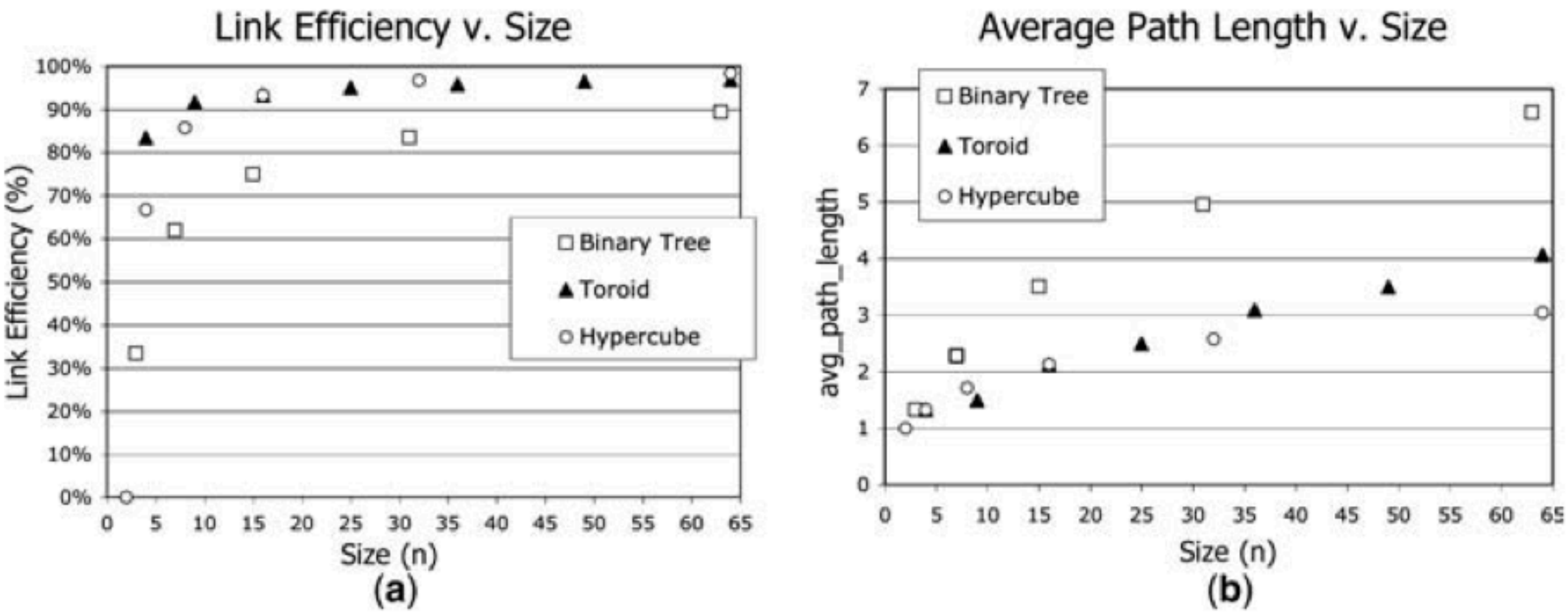


Figure 3.6 Comparison of actual link efficiency (a) and average path length (b) of binary tree, toroidal, and hypercube networks.

RANDOM NETWORKS

Generation

- Not as straightforward as it seems
  - The resulting network should be “random” (high-entropy)
  - Avoid node isolation, duplicate links, and loops
- Random network generation procedures
  - Gilbert Random Network
  - Erdos–Renyi (ER) Random Network
  - Anchored Random Network

Gilbert Random Network

- Idea
  - Select links with probability p from a complete graph with n nodes such that the resulting network ends up with $m = p[n(n-1)/2)]$ links on average
  - A Gilbert network has density p
  - A Gilbert network is one network from $C\binom{n}{2}$ possible networks with n nodes and m links
  - For example, if n=100, then

$$C\binom{100}{2} = \frac{100!}{2(98!)} = \frac{100(99)}{2} = 4950$$

a Gilbert network is one of 4950 possible networks, selected at random

Procedure

Given n and probability p, generate a Gilbert network by applying the following microrules:

1. Initially: generate n nodes and number them from 0 to $(n-1)$.
2. Set m: Let $m = n((n-1)/2)$ = the number of nodes in a complete graph.
3. Repeat for $i = 0, 1, \ldots, (m-1)$:
   a. Given (`Math.random()` $< p$), connect link i to a node pair; otherwise ignore.
   b. Count the number of links connected and compute the density:

$$\text{Density} = \frac{\text{number of connected links}}{m}$$

ER Random Network

- Proposed in 1959 by Paul Erdos and Alfred Renyi
- The standard method of random network generation today
- Idea
  - Fixes the number of links *m* and nodes *n* and does away with the probability variable p
  - Avoids loops and duplicate links, but it does not guarantee a strongly connected network

Procedure

Given n and m, construct an ER random network as follows:

1. Initially: Generate n nodes and number them from 0 to $(n - 1)$.
2. Initially: m given, and #links (number of links) $= 0$.
3. Repeat until $m =$ #links have been inserted:
   a. Select random node: tail $=$ `(Math.random())`n.
   b. Select random node: head $=$ `(Math.random())`n.
   c. Avoid loop: while (tail $==$ head) head $=$ `(Math.random())`n.
   d. Avoid duplicate: if (no duplicate) insert new link between tail and head and increment #links. Otherwise, do nothing.

Anchored Random Network

- Idea
  - A slight modification to the ER generative procedure guarantees that all nodes are connected to at least one other node
  - Visiting every node at least once (in round-robin style) and testing the degree of each node
    - If the degree is zero, the algorithm attaches the tail of the link to the solitary node
    - Otherwise, it selects a tail node at random

Procedure

Given n and m, construct an anchored random network as follows:

1. Initially: Generate n nodes and number them from 0 to $(n-1)$.
2. Initially: $m \geq (n/2)$ given, and #links=0.
3. Repeat until $m =$ #links have been inserted:
   a. Round robin: $i = 0,1,2,\ldots,(n-1); 0,1,2,\ldots$.
   b. Select tail: If (degree(i) $>$ 0) tail = `(Math.random()`n, else tail = i.
   c. Select random node: head = `(Math.random())`n.
   d. Avoid loop: While (tail == head) head = `(Math.random())`n.
   e. Avoid duplicate: If (no duplicate), insert new link between tail and head and increment #links. Otherwise, do nothing.

Degree Distribution

- Gilbert and ER generation procedures both obey a *Poisson distribution* ($n >> 1$)
- How to show it?
  - Show that random selection of node pairs follows a binomial distribution
  - Show that the binomial distribution transforms into the Poisson distribution as the number of links m grows large, thus eliminating m from the distribution equation
  - *Note:* Use the fact that $((1 - \lambda)/m)$ becomes $e^{-\lambda}$ as m grows without bound

Degree Distribution

- The degree distribution of *G* is binomial
  - Consider *G* with n = 10 nodes, m = 30 links, so the *avg_degree*(*G*) = λ = 2*m*/*n*) links
  - According to the ER generation procedure, each node receives an average of six connections in m = 30 timesteps
  - For a node *v*

$$\text{Prob}(v \text{ selected } k \text{ times in } m \text{ steps}) = \left(\frac{\lambda}{m}\right)^k \left(\frac{1-\lambda}{m}\right)^{m-k}$$

$$\text{Prob}(\text{degree}(v) = k) = B(m,k) = C\binom{m}{k}\left(\frac{\lambda}{m}\right)^k \left(\frac{1-\lambda}{m}\right)^{m-k}$$

Degree Distribution

- As *n* and *m* increase in size, *B*(*m*, *k*) is approximated by the *Poisson distribution*

$$\lim_{m\to\infty}\{B(m,k)\} = \lim_{m\to\infty} \underbrace{(m(m-1))\cdots\left(\frac{m-k+1}{m^k}\right)}_{T_1}\underbrace{\left(\frac{\lambda^k}{k!}\right)}_{T_2}\underbrace{\left(1-\frac{\lambda}{m}\right)^m}_{T_3}$$

$$\text{Lim}(m \rightarrow \text{infinity})\ T_1 = 1$$

$$\text{Lim}(m \rightarrow \text{infinity})\ T_2 = \frac{\lambda^k}{k!}$$

$$\text{Lim}(m \rightarrow \text{infinity})\ T_3 = e^{-\lambda}$$

$$\text{Lim}(m \rightarrow \text{infinity})\ B(m,k) = \frac{\lambda^k e^{-\lambda}}{k!}$$

  - This is a *Poisson distribution*

Example

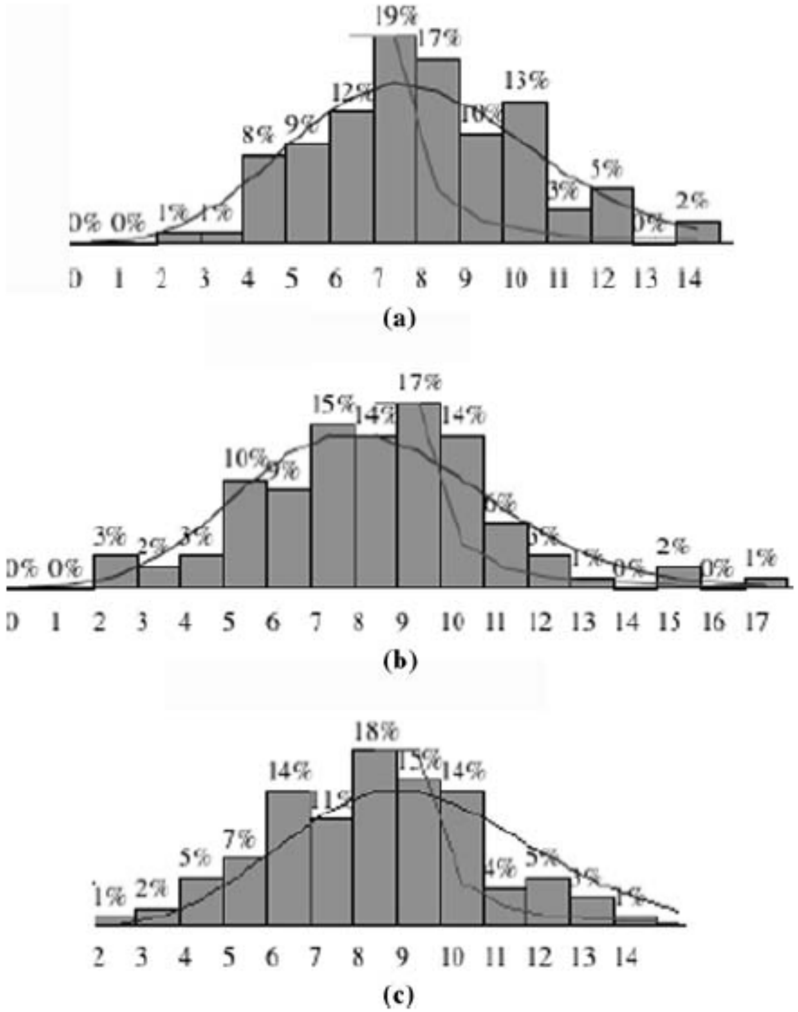


Figure Degree sequence distribution of random networks ($n = 100$, $m = 400$) generated by (a) Gilbert generation procedure, (b) ER generation procedure, and (c) anchored ER generation procedure.

Entropy

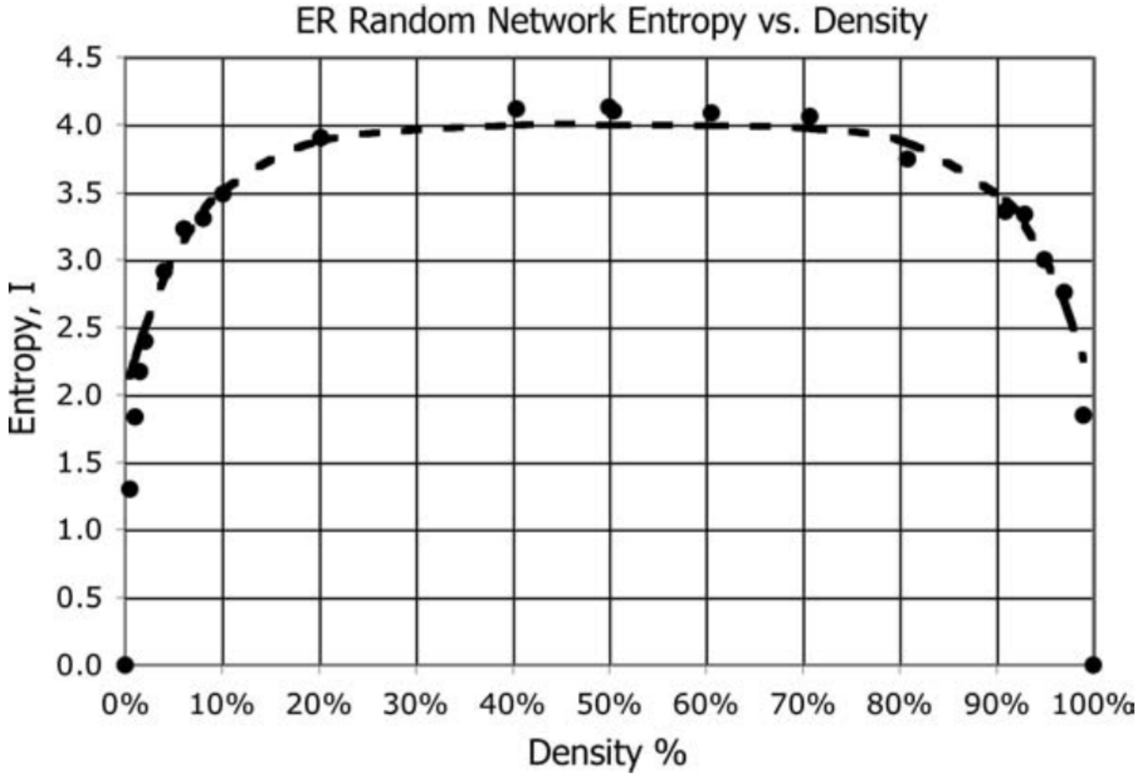


Figure Entropy of ER random network versus density as a percentage of fully connected (complete) network: $n = 100$, $m = 100$ to 4950.

A random network is fully random only in the middle of its range of density values

Model of Entropy

- In the left half, entropy rises exponentially and then flattens off near density = 50%, and similarly declines to zero over the right half
- The left half and right-half expressions model I(x) and I(1 — x), for 0 < x < 1:

$$\text{Left}(x) = A(1 - \exp(-Bx)); \text{ left half}$$

$$\text{Right}(x) = A(1 - \exp(-B(1 - x))); \text{ right half}$$

Combining the two halves, we have:

$$I(x) = 0.5[\text{left}(x) + \text{right}(x)] = 0.5[A(1 - \exp(-Bx)) + A(1 - \exp(-B(1 - x)))]$$

Average Path Length

- The average path length of a random network decreases as the number of links increases

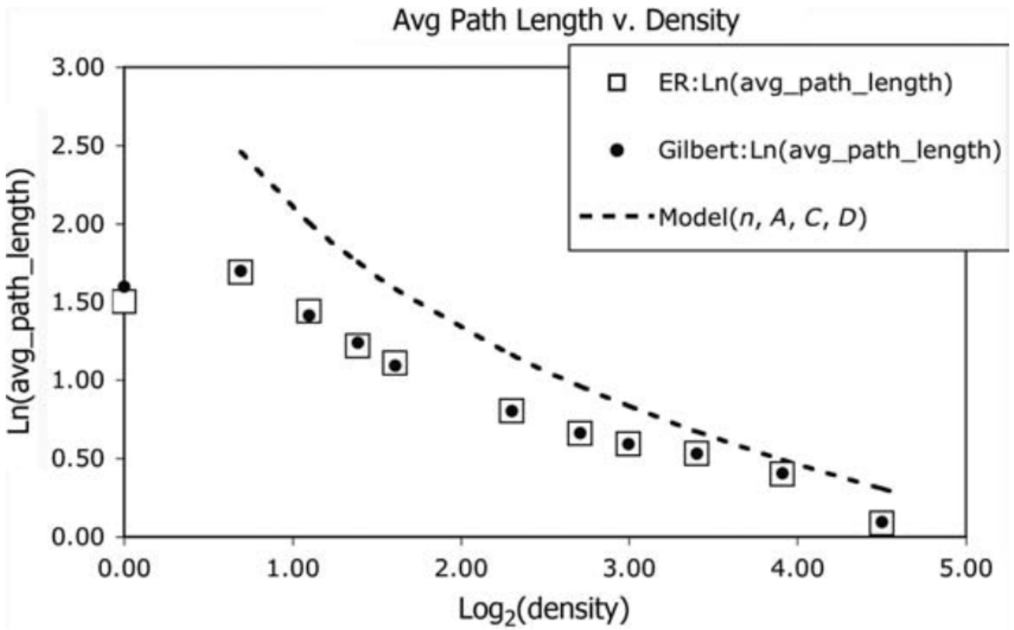


Figure Average path length versus density of links for ER, Gilbert networks, and model (dashed line), based on the modified theoretical approximation and base 2 logarithms: $n = 100,\ A = 1.32,\ C = 1.51,\ D = 0.$

$$\text{Model}(n, A, C, D) = \frac{A \log(n)}{\log(n(\text{density})C) + D}$$

Cluster Coefficient

- The cluster coefficient increases proportional to density

$$\text{cluster\_coefficent(random network)} = \frac{\lambda}{n}$$

where λ = mean degree = $(2m/n) = (n-1)\text{density} \sim n(\text{density}); n \gg 1$.

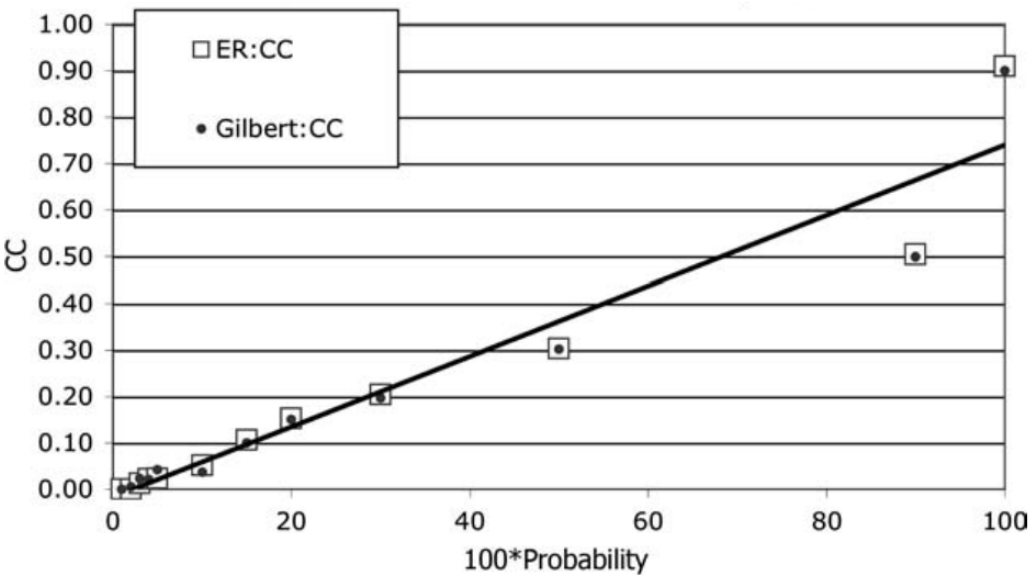


Figure Cluster coefficient of ER and Gilbert random networks versus link density (probability). A straight line approximates the relationship cluster_coefficient (random network) = O(density).

Link Efficiency

$$E(\text{random}) = \frac{m - \text{avg\_path\_length}(\text{random})}{m} = 1 - \frac{\text{avg\_path\_length}(\text{random})}{m}$$

Number of links m and density d are related by

$$d = \frac{2m}{n(n-1)}$$

so link efficiency can be expressed in terms of m or d:

$$E(\text{random}) = 1 - \frac{A \log(n)}{m \log(n(\text{density})C)}$$

Example

- When n = 100, d = 0.04 (m = 200, A = 1.32, C = 1.51):

$$E(\text{random}) = 1 - \frac{1.32 \log(100)}{200 \log((100)(0.04)(1.51))} = 1 - 1.32\frac{6.64}{200 \log(6.04)}$$
$$= 1 - \frac{8.77}{(200)(2.59)}$$
$$= 0.983$$

- Compare this with results of regular networks, random networks are highly efficient users of links because of the small-world effect
  - A small amount of randomness in any network injects a major drop in average path length
  - Randomness results in a large jump in link efficiency

Properties

- Diameter
  - The maximum-length path across all node-pairs
- Center
  - The minimum of the maximum length paths, from any node to any other node
- Radius
  - The longest path from a node *v* to all other nodes of a connected graph
- Closeness
  - The number of direct paths from all nodes to all other nodes that must pass through the node *v*

Diameter

- We model the decrease in diameter as density increases using a modified average path length model:

$$\text{Model}(n, A, C, D) = \frac{A \log(n)}{\log(n(\text{density})C) + D}$$

- Given the parameters *n* = 100, A = 2.0, C = 0.44,

  D = 1.0, density = 0.5:

$$\text{Diameter} = \frac{2 \log(100)}{\log((0.44)(100)(0.5)) + 1} = 2.43 \text{ hops}$$

$$\text{avg\_path\_length} = \frac{1.32 \log(100)}{\log((100)(0.5)(1.51))} = 1.41 \text{ hops}$$

Radius

- Random network centrality "shrinks" with increasing density:

$$\text{Radius(random)} = \frac{A\log(n)}{\log(n(\text{density})C) + D}$$

- Given the parameters *n* = 100, A = 1.59, C = 0.88, and D = 0.5, density = 0.5:

$$\text{Radius(random)} = \frac{1.59\log(100)}{\log((100)(0.5)(0.88)) + 0.5}$$

$$= 1.59\frac{6.64}{\log(44) + 0.5}$$

$$= 1.77\,\text{hops}$$

Closeness

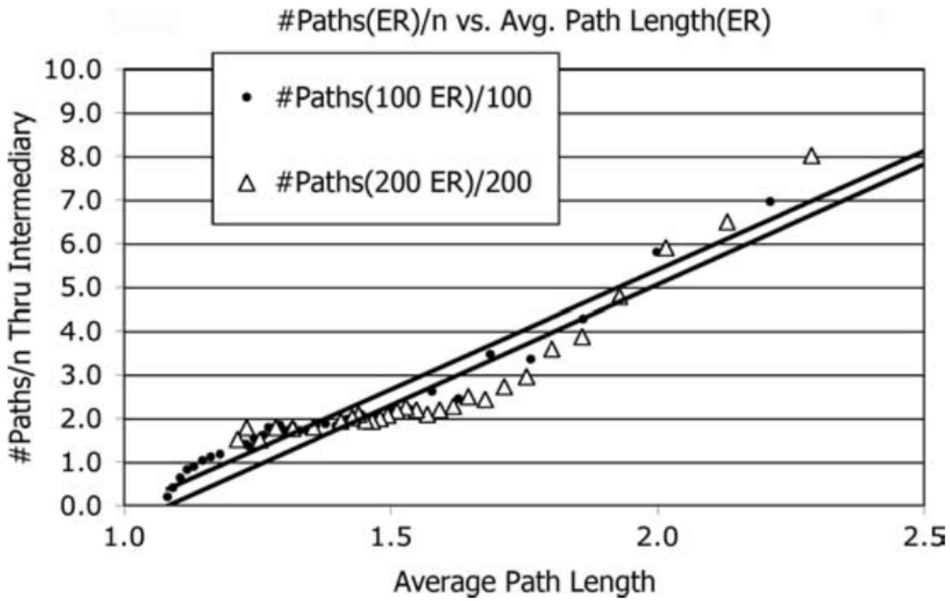


Figure Number of paths through the largest intermediary node versus average path length of a random network for $n = 100{,}200$.

There is a somewhat linear relationship between closeness and path length:

$$\frac{\text{Number of paths through intermediate node}}{n} = O(\text{avg\_path\_length})$$

$$\text{avg\_path\_length} = O\left(\frac{\log(n)}{\log(\lambda)}\right), \text{ and } \lambda = n(\text{density}):$$

$$\text{Number of paths through intermediate node} = O\left(\frac{n\log(n)}{\log(n(\text{density}))}\right)$$

Closeness

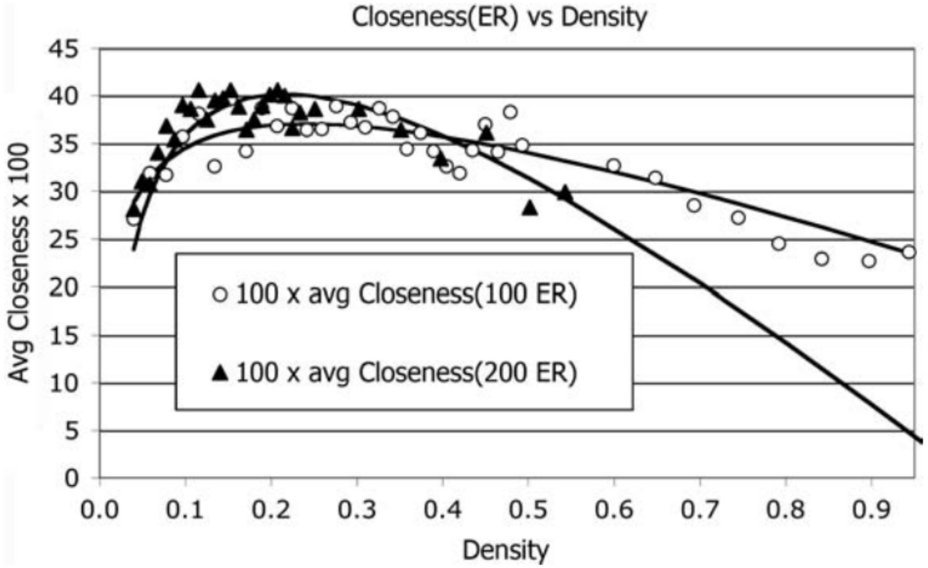


Figure Average closeness versus density for random networks of size $n = 100,200$. Closeness rises to a peak and then declines with increase in number of links.

Consider length of average paths and number of direct paths (suppose n = 100):

$$100(\text{closeness}(\text{random})) = C_0(1 - \text{density})\lambda^r + C_1$$

$$r = \frac{A \log_2(n)}{\log_2(B\lambda) + C}$$

Weak Ties

The population of Pointville, Flatland is 129. The village is unusual in that everyone is equally likely to know everyone else, but because of their busy social life, Flatlanders have time to
become friendly with an average of only 24 other Flatlanders. What are the longest and shortest weak ties across Pointville?

- A *weak tie* is a chain of acquaintances that leads from person u to person *w*, for any node pair *u~w*, across the entire population
- The longest and shortest weak ties are equivalent to the *diameter* and *radius* of this population

Weak Ties

$$\text{Density} = d = \frac{2m}{n(n-1)} = \frac{2\lambda(n/2)}{n(n-1)} = \frac{\lambda}{n-1}$$

Substitution of density into the approximations for diameter and radius derived earlier (we assume that $n = 100$ and $n = 129$ give similar values of parameters A,C, and D):

$$\text{Diameter} = \frac{2\log(n)}{\log(0.44nd) + 1}$$

$$\text{Radius} = \frac{1.59\log(n)}{\log(nd0.88) + 0.5}$$

Now, let $n = 129$ and $\lambda = 24$:

$$d = \frac{24}{128} = 0.1875$$

$$\text{Diameter} = \frac{2\log(129)}{\log((0.44)(129)(0.1875)) + 1} = 3.18 \text{ or } 4\,\text{hops}$$

$$\text{Radius} = \frac{1.59\log(129)}{\log((129)(0.1875)(0.88)) + 0.5} = 2.77 \text{ or } 3\,\text{hops}$$

The longest and shortest weak ties differ by 1 hop. Every person in Pointville knows every other person through at most four intermediaries.

Randomization of Regular Networks

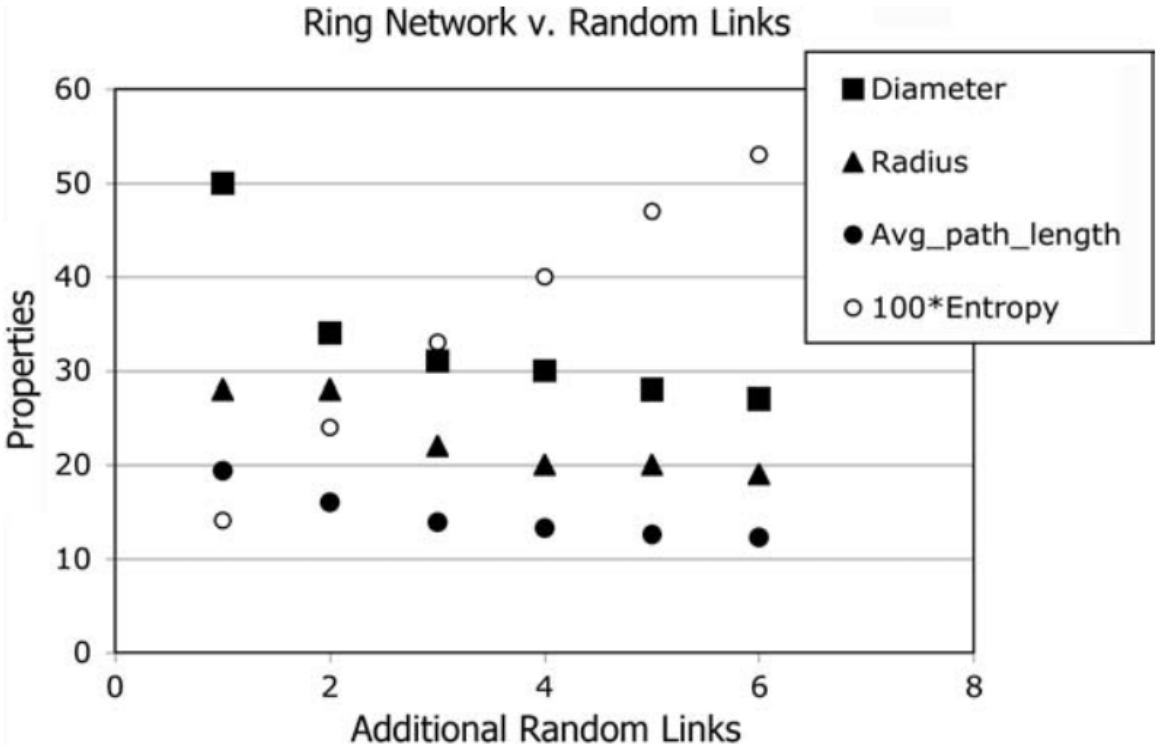


Figure Effect on diameter, radius, average path length, and 100× entropy of ring network as the number of additional random links increases.

SMALL-WORLD NETWORKS

Definition

- Small-world networks are sparse networks with high cluster coefficient, relatively short average path length, and scalable entropy
- Small-world effect is the rapid decline in average path length as a small number of random links are added to a (structured) network
- Random links tend to bisect a network, effectively dividing the distance between opposite halves of the network by 50%
- "six degrees of separation" $\rightarrow$ "50% elimination of separation"

Generation

- The Watts–Strogatz (WS) Procedure

1. Given n, rewiring probability p, and $k = 2$, generate a k-regular graph by connecting each of n nodes to their immediate neighbors, and neighbor's neighbors. This network has $m = 2n$ links [$\lambda = 4$, density = $(4/\mathrm{n})$].
2. For every link, $\mu = 1, 2, \ldots, m$, rewire μ with probability p, as follows. If (`Math.random()` $< p$), disconnect the head(μ), and rewire it to a different randomly selected node. Avoid ($v_{\text{random}} = \text{head}(\mu)$), and duplicate links. Otherwise, do nothing.

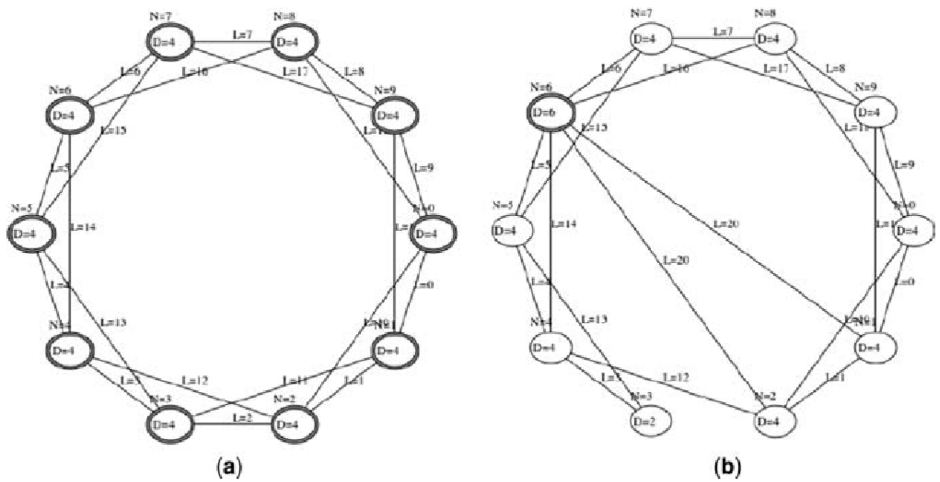


Figure WS small-world generation starts with (a) a 2-regular network and then evolves into (b) a semirandom and semistructured "slightly small-world" network: $n = 10$, $m = 20$.

Degree Sequence

- Small worlds are hybrids
  - Part k-regular and part random
- The topology of a small-world network falls somewhere between that of a k-regular network and random network

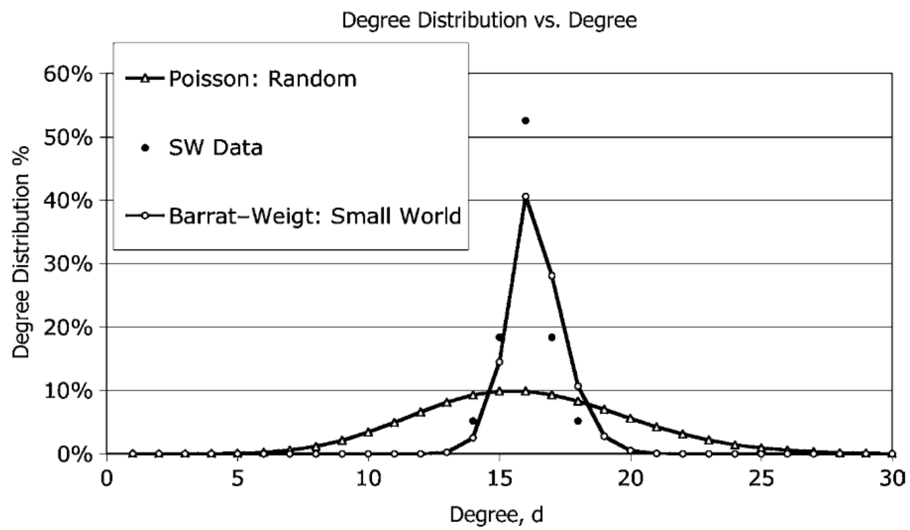


Figure Degree sequence distribution for random and small-world networks with $n = 50$, $m = 400$, and rewiring probability $p = 5\%$.

Closed-form Expression

1. The degree of a typical node u stays the same if none of its links are rewired, and increases if other node's links are redirected to u.
2. The probability that links are *not* rewired follows a binomial distribution, $B(k,i,(1-p))$, where $k = m/n$, $i =$ number of links not rewired, and $p =$ rewiring probability.
3. The probability that another node's links are redirected to node u follows a Poisson distribution, $P(\lambda_1, d-k-i)$; $d \geq k$, where $\lambda_1 = pk$ is the expected value of redirected links, $d =$ degree, and $i =$ number of links redirected to u.
4. The probability of increasing the degree of node u is equal to the joint probability, $B(k,i,(1-p))P(\lambda,d-k-i)$.
5. The degree distribution $h(d)$ is equal to the sum of joint probabilities over $i =$ 1, 2, 3, ... $\min\{d-k,k\}$ links.

The distribution of nodes with d links after rewiring is the sum

$$h(\mathrm{d}) = \sum_{i=1}^{\min\{d-k,k\}} B(k,\, i,\, (1-p))P(\lambda_1,\, d-k-i);\;\; d \geq k$$

Example

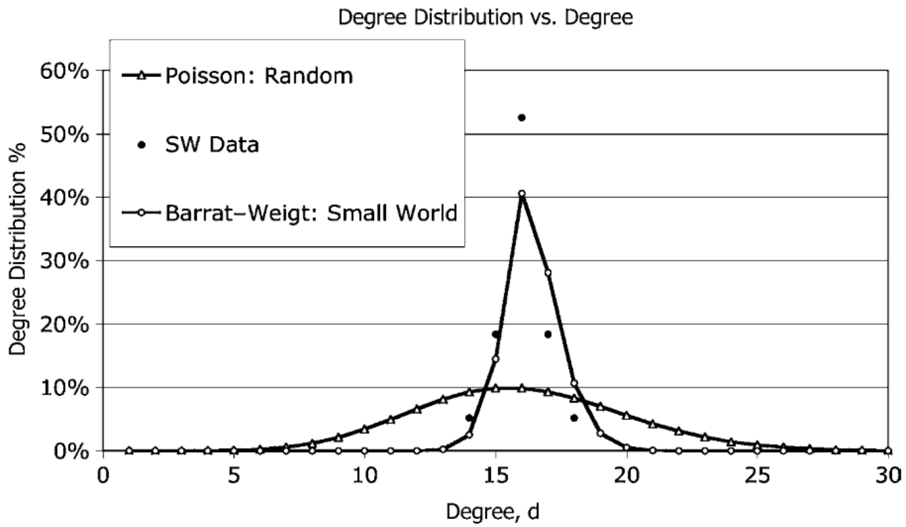


Figure Degree sequence distribution for random and small-world networks with $n = 50$, $m = 400$, and rewiring probability $p = 5\%$.

$$d = 15$$

$$p = 0.05$$

$$k = \frac{m}{n} = \frac{400}{50} = 8$$

$$\min\{d - k, k\} = \min\{15 - 8, 8\} = 7$$

$$\lambda_1 = pk = 0.05(8) = 0.4$$

$$h(15) = \sum_{i=1}^{7} \{B(8, i, (0.95))P(0.4, 15 - 8 - i)\}$$

$$B(8, i, 0.95) = C\left(\begin{smallmatrix} i \\ 8 \end{smallmatrix}\right)(0.95)^i(0.05)^{8-i}$$

$$P(0.4, 15 - 8 - i) = (0.4)^{15-8-i} \exp\frac{-0.4}{(15 - 8 - i)!}$$

$$h(15) = \sum_{i=1}^{7} \left\{ C\left(\begin{smallmatrix} i \\ 8 \end{smallmatrix}\right)(0.95)^i(0.05)^{8-i}(0.4)^{7-i} \exp\frac{-0.4}{(7 - i)!} \right\}$$

TABLE Summation Terms for $h(d)$

d	min (d-k, k)	$h(d)$	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$	$i = 6$	$i = 7$	$i = 8$
10	2	0.0%	0%	0%	0	0	0	0	0	0
11	3	0.0%	0%	0%	0%	0	0	0	0	0
12	4	0.0%	0%	0%	0%	0.02%	0	0	0	0
13	5	0.3%	0%	0%	0%	0.01%	0.24%	0	0	0
14	6	2.5%	0%	0%	0%	0.01%	0.19%	2.31%	0	0
15	7	14.5%	0%	0%	0%	0.00%	0.08%	1.85%	12.55%	0
16	8	40.6%	0%	0%	0%	0.00%	0.02%	0.74%	10.04%	29.81%
17	8	28.1%	0%	0%	0%	0.00%	0.00%	0.20%	4.02%	23.85%
18	8	10.7%	0%	0%	0%	0.00%	0.00%	0.04%	1.07%	9.54%
19	8	2.8%	0%	0%	0%	0.00%	0.00%	0.01%	0.21%	2.54%
20	8	0.5%	0%	0%	0%	0.00%	0.00%	0.00%	0.03%	0.51%

Properties

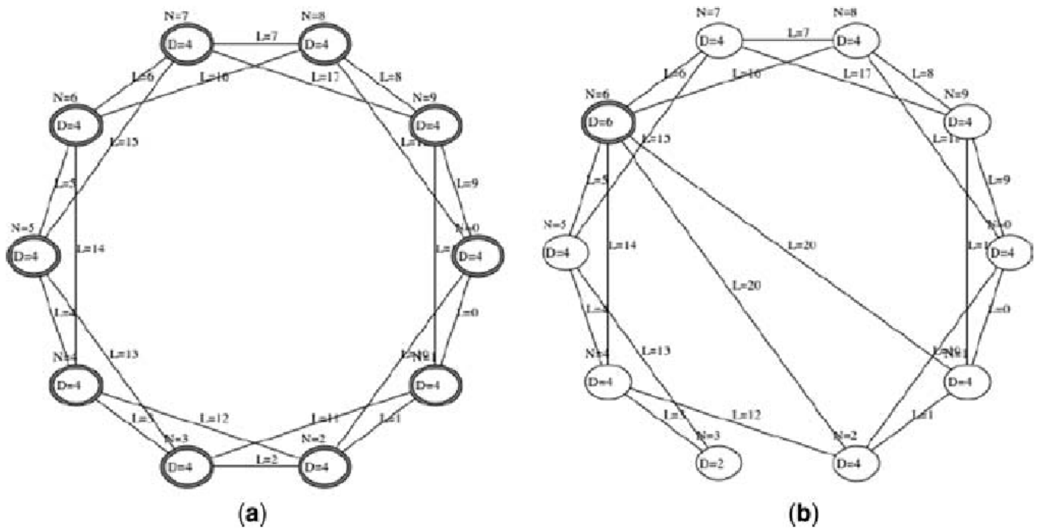


Figure WS small-world generation starts with (a) a 2-regular network and then evolves into (b) a semirandom and semistructured "slightly small-world" network: $n = 10$, $m = 20$.

TABLE WS Small-World Networks Generated by Rewiring a 2-Regular Network versus a Toroidal Network

Property	2-Regular	WS Small World	Toroidal Network	Toroid→ SW	Random
avg_path_length	3.5	2.87	2.5	2.37	2.33
Cluster coefficient *CC*	0.500	0.363	0	0.055	0.169
Entropy	0	2.83	0	3.61	5.82

Entropy versus Rewiring Probability

- Explore the relationship between each small-world property and the rewiring probability and density of the network
  - Hold the size and density of a 2-regular WS network constant while varying the rewiring probability p
  - Hold the size and rewiring probability constant while varying density.
  - Do this by starting with a k-regular network $k = m/n$, and noting that density = $2k/(n-1)$

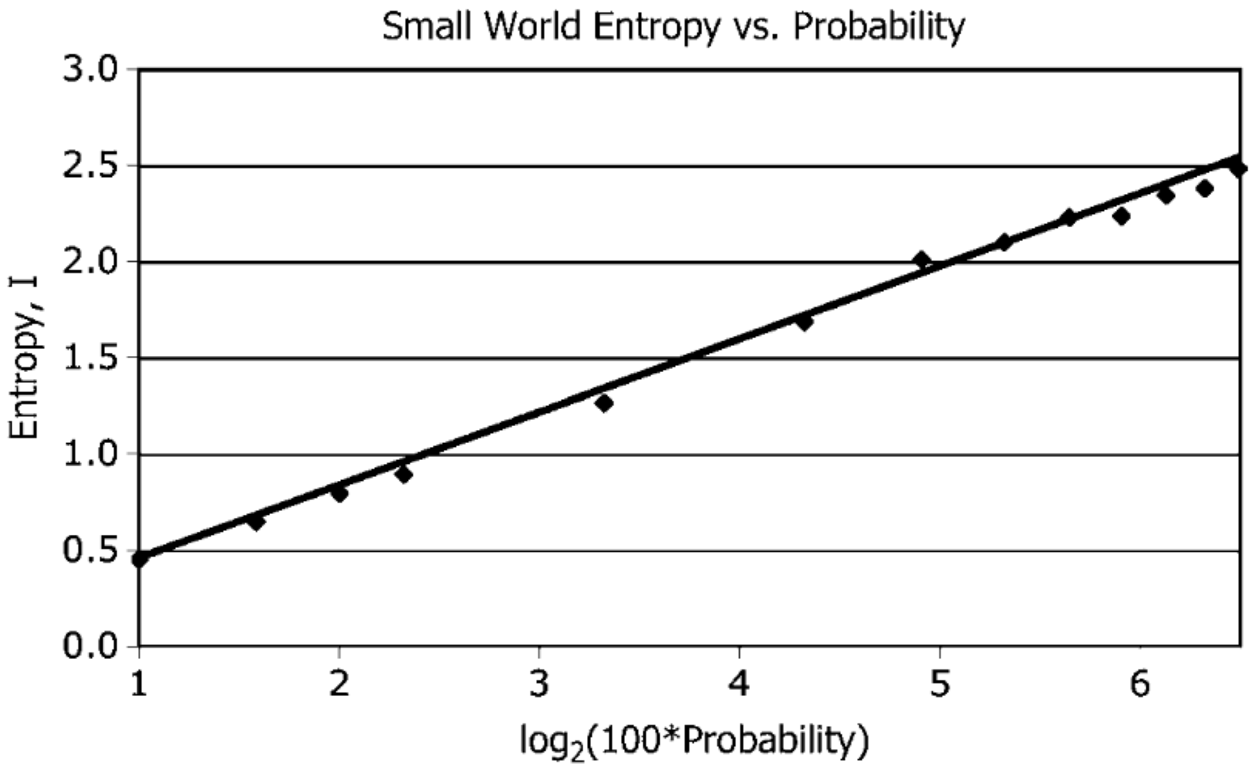


Figure Entropy of a WS small world versus logarithm of rewiring probability p; $n = 100$, $m = 200$. Rewiring probability ranges from 1% to 100%.

$$\text{Entropy}_{\text{WS}(p)} = I_{\text{WS}(p)} = A \ \log_2(100p); \ \ p = 0.01, 0.02, \ldots, 1.0$$

Entropy versus Density

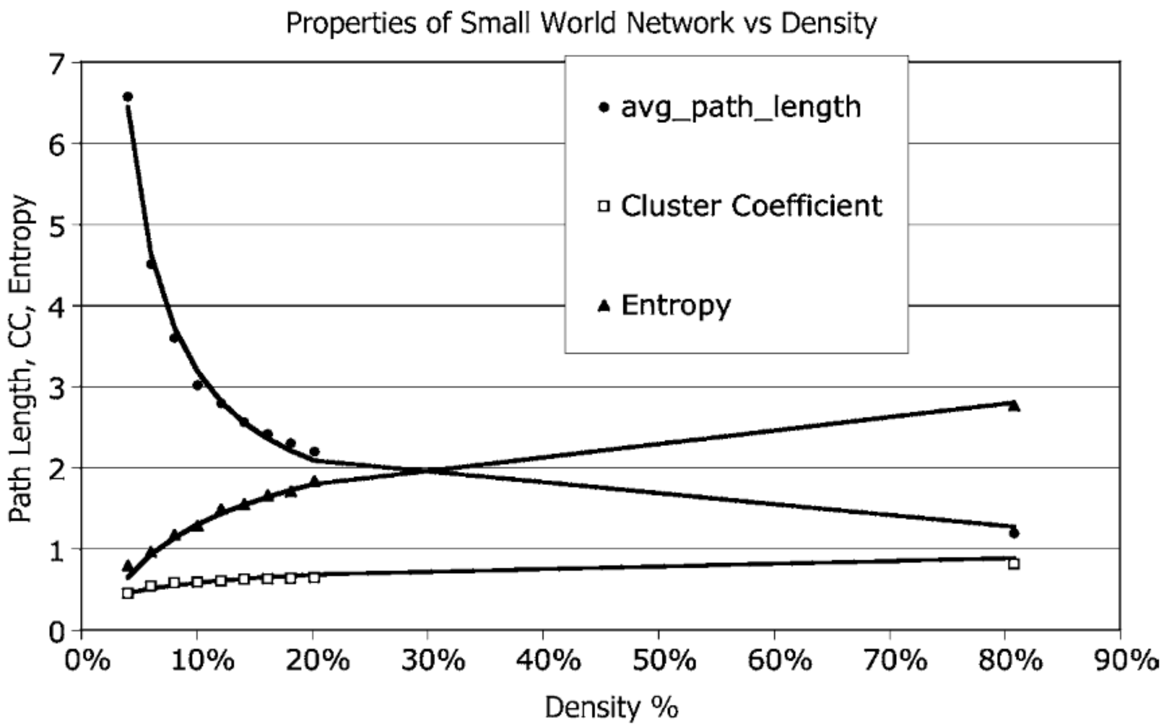


Figure Properties of small-world network with rewiring probability $p = 4\%$, $n = 100$, and $4\% \leq$ density $\leq 80\%$; [*Note*: kn (density/2)].

$$\text{Density}(k\text{-regular}) = 2\frac{k}{n};\ \ k = n\frac{\text{density}}{2};\ \ n \gg 1$$

Entropy rises very slowly with density according to the logarithmic function $I_{\text{WS(density)}} = A\ \log_2(B(\text{density})) - C$.

Path Length

- Deriving average path length

    1. When rewiring is nonexistent, and the initial network is 2-regular, average path length is $n/4k$.
    2. For very small rewiring probability p, average path length begins a rapid decline, after $p \geq (1/m)$.
    3. At some (early) point p^*, rewiring is sufficiently large that the network transitions from mostly regular, to mostly random. This is known as the *crossover point* p^*, and signifies a *phase transition* in the network. The value of p^* is of interest to physicists, who associate it with phase transitions in materials.

- Average path length declines from an initial (p = 0) value of $n/4k$ according to a scaling function f (r), r is 2 times the average number of rewired links (r = 2pm) (Newman–Moore–Watts expression)

$$f(r) = 4\frac{\tanh^{-1}(r/\beta)}{\beta}; \text{ where } \beta = \sqrt{(r^2 + 4r)}$$

$$\text{avg\_path\_length(SW)} = n\frac{f(r)}{2k} = \frac{2n}{\beta k}\tanh^{-1}\left(\frac{r}{\beta}\right)$$

Example

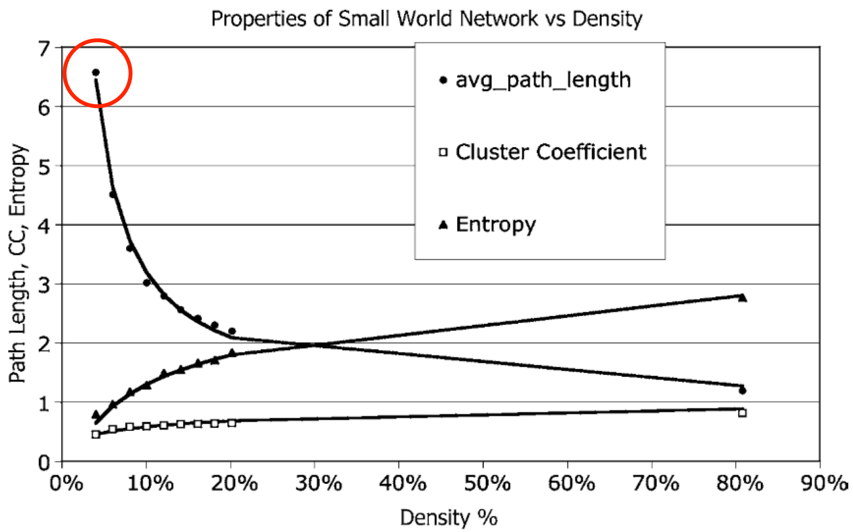


Figure Properties of small-world network with rewiring probability $p = 4\%$, $n = 100$, and $4\% \leq \text{density} \leq 80\%$; [*Note*: kn (density/2)].

$$n = 100,\ m = 200,\ k = 2,\ \text{density} = 0.04,\ \text{and}\ p = 0.04$$

$$r = (2)(0.01)(400) = 8, \quad \beta = \sqrt{(64 + 32)} = 9.8$$

$$\text{arc\_tanh}\left(\frac{8}{9.8}\right) = 1.15$$

$$\frac{2n}{\beta k} = \frac{200}{2(9.8)} = 10.2$$

$$\text{avg\_path\_length (SW)} = (10.2)(1.15) = 11.7 \text{ hops}$$

The Newman–Moore–Watts expression overestimates average path length

Cluster Coefficient

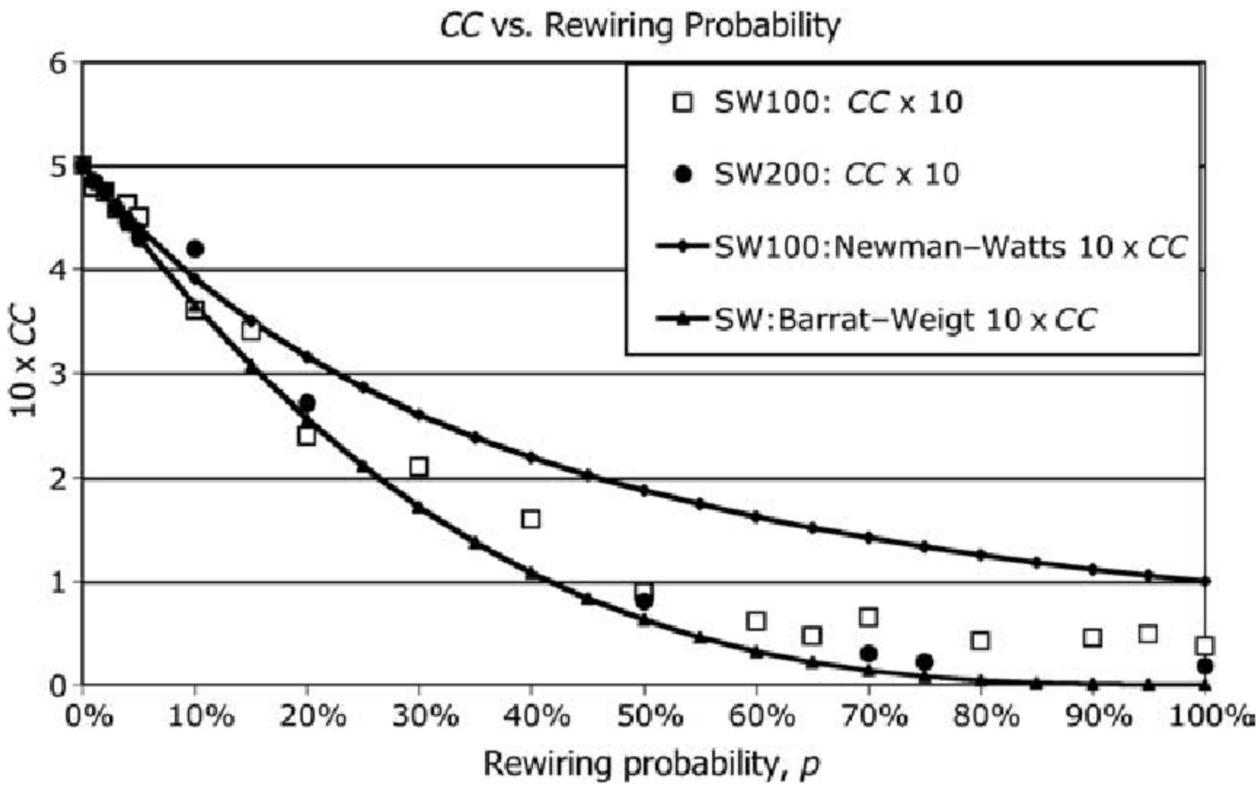


Figure Cluster coefficient of WS small-world versus rewiring probability for $n = 100$, 200, and $m = 200$, 400 ($\lambda = 4$, $k = 2$).

Newman-Watts: $CC(k\text{-regular}) = 3\dfrac{k(k-1)}{2k(2k-1) + 8pk^2 + 4p^2k^2}$, p = rewiring probability

Barrat-Weigt: $CC(k\text{-regular}) = CC(0)(1-p)^3$, where $CC(0) = 3\dfrac{k-1}{2(2k-1)}$, $k = 2, 3, \ldots$

Closeness

- Closeness increases with increasing density—up to a point—and then dips

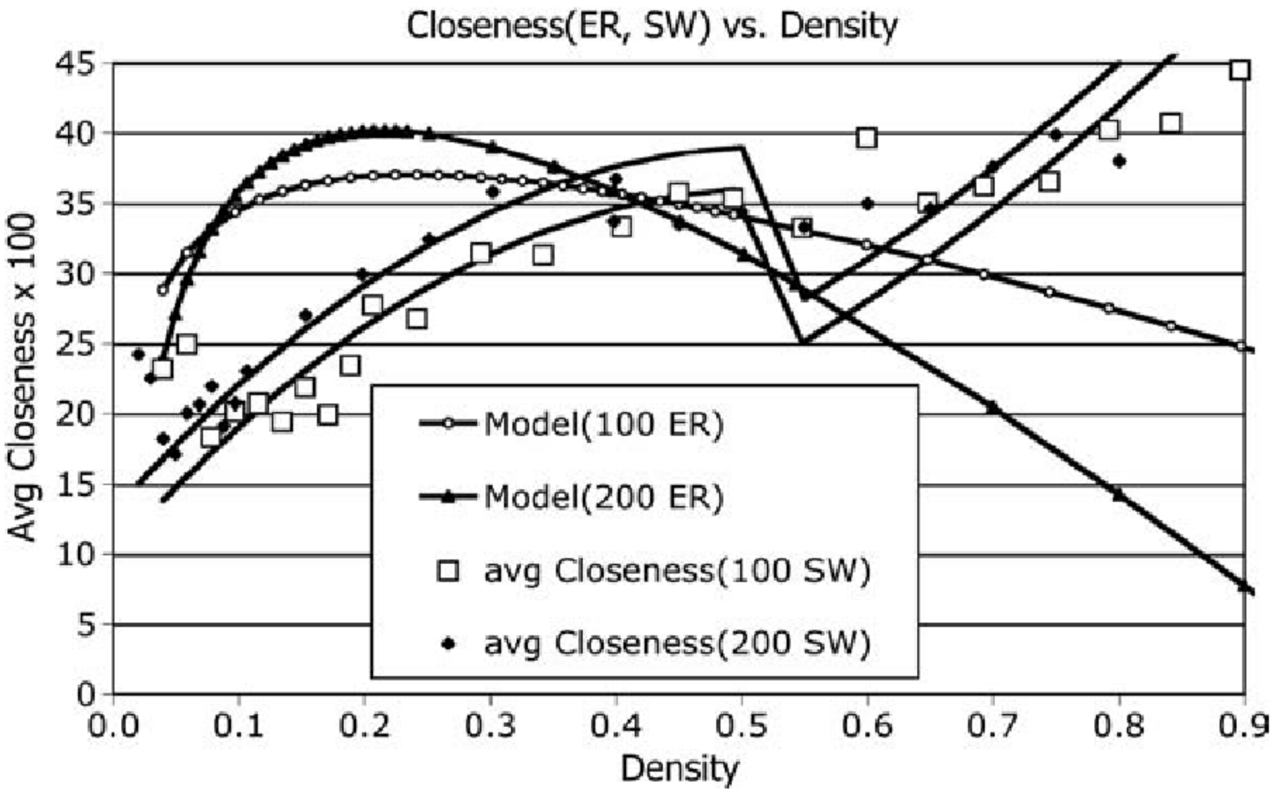


Figure Average closeness versus density for small worlds of size $n = 100, 200$, compared with equivalent random networks of the same size.

Model Approximation

1. Modify the mean-field approximation obtained in the previous chapter, closeness(random) $= O((1 - \text{density})z)$, to accommodate the impact of k-regularity on the network, where $z = \lambda^r$, $\lambda =$ mean degree, and $r(\text{random}) = O(\log(n)/\log(\lambda))$:
   a. Replace (1-density) with density because small worlds increase closeness as density increases—up to a point (50%),
   b. Note that the transformation from small-world to k-regular network occurs around $r = 1$.
   c. Use the fact that a *direct* path of length 1 contributes zero closeness because of the way closeness is defined.
2. Estimate curve-fit parameters C_1 and C_2, from data points collected by simulation.

$$\text{closeness(small world, } p) = C_1(\text{density})z + C_2$$

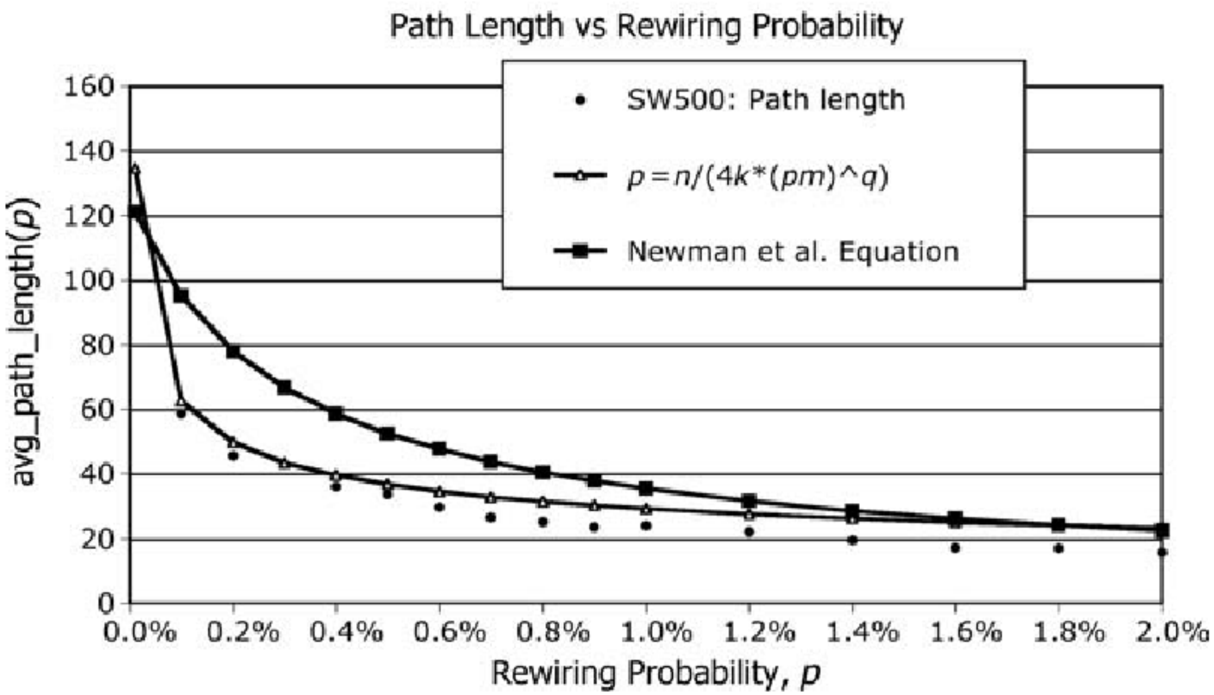
$$z = \begin{cases} \lambda^r & \text{if} \quad r \geq 1 \\ \dfrac{\lambda}{2} & \text{otherwise} \end{cases}$$

$$r = \frac{A \log_2(n - (1-p)\lambda)}{\log_2(\lambda)}$$

Phase Transition

- Phase transition in the physical world occurs when matter changes from a solid state to a liquid, from liquid to gas, and so on
- The idea is related to the sudden transition from a 2-regular to random network as rewiring probability increases
- Phase transition is a particular property of small worlds

Phase Transition



The transition from "mostly vertical" to "mostly horizontal" line in the figure is considered a *phase transition point*, and the rewiring probability corresponding to the transition is considered the *crossover point p**

Phase Transition

Define p^* as the point where the slope of the avg_path_length L versus p equals (-45°). This corresponds with a rate of change $\delta L/\delta r = -1$:

$$L = \frac{n/4k}{(r)^q}$$

$$\frac{\delta L}{\delta r} = -1 = -\frac{nq}{4kr^{q+1}}$$

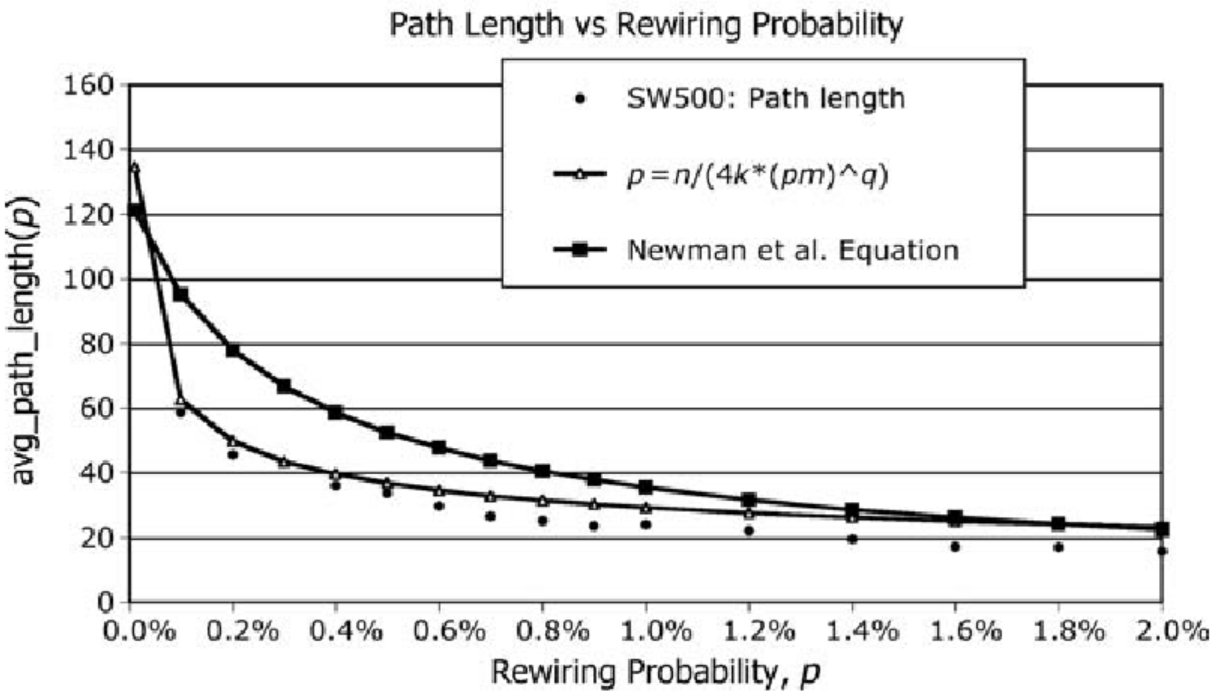
Solving for r, we obtain

$$r = \left(\frac{nq}{4k}\right)^{1/(q+1)}$$

$$= pm, \text{ so } p^* = \frac{r}{m} \text{ at crossover point}$$

$$p^* = \frac{r}{m} = \left(\frac{nq/m}{4k}\right)^{1/(q+1)}$$

Example



$q = \frac{1}{3}$, $1/(q+1) = 0.75, n = 500, m = 1000,$ and $k = 2$:

$$r = \left(\frac{nq}{4k}\right)^{0.75} = \left(\frac{500\left(\frac{1}{3}\right)}{4(2)}\right)^{0.75} = 9.75$$

$$p^* = \frac{r}{m} = \frac{9.75}{1000} = 0.00975 = 0.975\%$$

Therefore, the crossover point is approximately 1%.