

The Optimal Path in a Random Network

Lidia A. Braunstein^{1,2}, Sergey V. Buldyrev¹, Sameet Sreenivasan¹, Reuven Cohen³, Shlomo Havlin^{1,3}, and H. Eugene Stanley¹

¹ Center for Polymer Studies and Department of Physics
Boston University, Boston, MA 02215, USA

² Departamento de Física, Facultad de Ciencias Exactas y Naturales
Univ. Nacional de Mar del Plata, Funes 3350, 7600 Mar del Plata, Argentina

³ Minerva Center and Department of Physics
Bar-Ilan University, 52900 Ramat-Gan, Israel

Abstract. We study the optimal distance ℓ_{opt} in random networks in the presence of disorder implemented by assigning random weights to the links. The optimal distance between two nodes is the length of the path for which the sum of weights along the path (“cost”) is a minimum. We study the case of strong disorder for which the distribution of weights is so broad that its sum along any path is dominated by the largest link weight in the path. We find that in random graphs, ℓ_{opt} scales as $N^{1/3}$, where N is the number of nodes in the network. Thus, ℓ_{opt} increases dramatically compared to the known small world result for the minimum distance ℓ , which scales as $\log N$. We also find the functional form for the probability distribution $P(\ell_{\text{opt}})$ of optimal paths. In addition we show how the problem of strong disorder on a random network can be mapped onto a percolation problem on the Cayley tree and using this mapping, obtain the probability distribution of the maximal weight on the optimal path.

Introduction

Much attention has been focused on the topic of complex networks characterizing many biological, social, and communication systems [1–3]. The networks can be visualized by nodes representing individuals, organizations, or computers and by links between them representing their interactions. The classical model for random networks is the Erdős-Rényi model where two nodes are chosen randomly from the total N nodes in the system and are connected by a link [4]. An important quantity characterizing networks is the minimum distance ℓ between two nodes in the network. For the Erdős-Rényi network, ℓ scales as $\log N$, consistent with the “six degrees of separation” concept (e.g., if $N = 10^6$, $\ell \approx 6$).

Here we study a more realistic problem in which all links are not assumed to be equivalent. Hence we assign to each link a weight or “cost.” For example, the cost could be the time required to transit the link, e.g., there are often many traffic routes from point A to point B with a set of delay times τ_i associated with each link along the path. The fastest (optimal) path is the one for which $\sum_i \tau_i$ is a minimum, and often the optimal path has more links than the shortest path.

If the distribution of weights is such that all the links have the same weight, the average length of the optimal path between any two nodes is the minimal length ℓ_{\min} . In that case it is well known that $\ell_{\min} \sim \log N$ [5] (or, for some scale free networks $\ell_{\min} \sim \log \log N$ [6]). If the distribution is narrow, the average length of the optimal path ℓ_{opt} , in general, is greater than ℓ_{\min} but scales the same as ℓ_{\min} [7,8]. If the random distribution is broad, in the limit of infinite broadness, the disorder is called “strong” and only the largest weight in the path dominates the sum. The strong disorder limit is implemented by assigning to each link a potential barrier ϵ_i so that τ_i is the waiting time to cross this barrier. Thus $\tau_i = e^{\beta \epsilon_i}$, and the optimal path corresponds to the minimum ($\sum_i \tau_i$) over all possible paths. When $\beta = 1/kT \rightarrow \infty$, only the largest τ_i dominates the sum. Thus $T \rightarrow 0$ (very low temperatures) corresponds to the strong disorder limit.

We focus here on the case of strong disorder. This is believed to be the case for many computer and traffic networks, since the slowest link in communication networks determines the connection speed. We study this problem both theoretically and numerically and find that for random graphs ℓ_{opt} , the average length of the optimal path, scales as $N^{1/3}$.

Theoretical arguments

To obtain the optimal path in the strong disorder limit, we present the following theoretical argument. It has been shown [9,10] that the optimal path for $\beta \rightarrow \infty$ between two nodes A and B on the network can be obtained by the following algorithm:

1. Sort the edges by descending weight.
2. If the removal of the highest weight edge will not disconnect A from B – remove it.
3. Go back to step 2 until all edges have been processed.

Since the edge weights are random, so is the ordering. Therefore, in fact, one needs not even select edge weights to begin with. This “bombing” algorithm can be replaced by simply removing randomly chosen edges one at a time, where an edge is not removed if its removal will cause the connectivity between A and B to be lost. The final path left is the optimal path between A and B in the limit $\beta \rightarrow \infty$.

Since randomly removing links is a percolation process, the optimal path must be on the percolation backbone connecting A and B . Since the network is not embedded in space but has an infinite dimensionality, we expect from percolation theory that at criticality loops are not relevant and the random graph can be approximated by a Cayley tree with a Poisson degree distribution. Thus, the shortest path must be the same as the optimal path. It is also known from percolation theory that at criticality the mass S of the incipient infinite cluster scales as ℓ_{\min}^2 [11]. Since the spanning cluster S scales

at criticality as $N^{2/3}$ [12], it follows that

$$\ell_{\min} \sim \ell_{\text{opt}} \sim N^{\nu_{\text{opt}}}, \tag{1}$$

where $\nu_{\text{opt}} = 1/3$ [13].

To test Eq. (1), we apply two numerical approaches. The first approach is to find the optimal path (which minimizes the sum of weights) using the ultrametric approach described in Ref. [9]. The second approach is based on the ‘‘bombing’’ algorithm of Ref. [9].

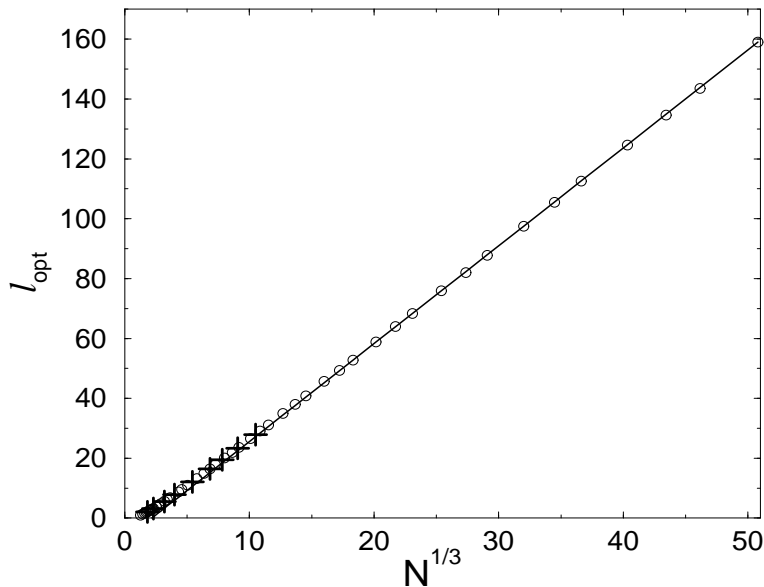


Fig. 1. ℓ_{opt} as a function of $N^{1/3}$ for the optimal path length in strong disorder using the two methods discussed in the text:(i) the results obtained using the ‘‘bombing’’ approach (\circ) and (ii) the results obtained using the ultrametric approach (\times).The dashed line is the linear fitting of the results showing the linear relation $\ell_{\text{opt}} = 3.27N^{1/3} - 7.11$. This result supports the theoretical value $1/3$.

Numerical analysis

Next we describe in detail the two numerical methods for computing ℓ_{opt} between any two nodes in strong disorder. We can assume that the energy spectra ϵ_i is discrete. We can make β so large that, even for the closest values of energy spectra, the waiting times $\tau_i = \exp[\beta\epsilon_i]$ differ by at least a factor of 2. In this limit, the sum is dominated by the maximum value $\exp[\beta\epsilon_{\text{max}}]$.

When all the links on the paths have different weights, the optimal path is the one that has the smallest maximal link weight between all the paths. In general, as a consequence of the existence of loops, there are links in common between different paths. Such a link might provide the maximum ϵ_i of both paths. In this case we compare the second highest weight and take the path with the lower value and so forth until the optimal path is determined. This procedure is equivalent to comparing integers written in binary codes and hence indeed minimizes $\sum \tau_i$ for $\beta \rightarrow \infty$.

First, we describe the ultrametric algorithm [10]. We assign weights to all the links in the graph where the order of magnitude is taken from a uniform distribution. This is accomplished by taking a random, uniformly distributed, variable of the *logarithm* of the weight. In the limit of strong disorder the sum of the weights is dominated by the largest value along the path. Next, we start from one node (the origin) and visit all the other nodes connected to the origin using the Dijkstra algorithm [14]. If a node at distance ℓ_0 (from the origin) is being visited for the first time, this node will be assigned a list S_0 of weights τ_{0i} , $i = 1 \cdots \ell_0$ of the links by which we reach that node sorted in descending order,

$$S_0 = \{\tau_{01}, \tau_{02}, \tau_{03}, \dots, \tau_{0\ell_0}\}, \quad (2)$$

with $\tau_{0j} > \tau_{0j+1}$ for all j . If we reach a node for a second time by another path of length ℓ_1 , we define for this path a new list S_1 ,

$$S_1 = \{\tau_{11}, \tau_{12}, \tau_{13}, \dots, \tau_{1\ell_1}\}, \quad (3)$$

and compare it with a S_0 previously defined for this node.

Different sequences can have weights in common because some paths have links in common, so it is not enough to identify the sequence by its maximum weight; in this case it must also be compared with the second maximum, the third maximum, etc. We define $S_p < S_q$ if there exists a value m , $1 \leq m \leq \min(\ell_p, \ell_q)$ such that

$$\begin{aligned} \tau_{pj} = \tau_{qj} & \quad \text{for} \quad 1 \leq j < m & \quad \text{and} \\ \tau_{pj} < \tau_{qj} & \quad \text{for} \quad j = m, \end{aligned} \quad (4)$$

or if $\ell_q > \ell_p$ and $\tau_{pj} = \tau_{qj}$ for all $j \leq \ell_p$.

If $S_1 < S_0$, we replace S_0 by S_1 . The procedure continues until all paths have been explored and compared. At this point, $S_0 = S_{\text{opt}}$, where S_{opt} is the sequence of weights for the optimal path of length ℓ_{opt} . In reality it is highly inefficient to compare all possible paths. This is why we use the Dijkstra algorithm. The Dijkstra algorithm explores only a limited set of paths, guaranteeing that the optimal path belongs to this set. The algorithm is implemented as follows. At the beginning we assign to every node i except one that we choose as our ‘‘origin’’, a value $S_i = \{\infty\}$. The origin is assigned a value $S_0 = \{0\}$. The search for the optimal path follows a procedure akin to ‘‘burning’’ where the ‘‘fire’’ starts from our chosen origin. In the first step, we

burn all the neighbors of the origin and replace the values assigned to them by the weight of the link that connects them to the origin. For example, if node i is a neighbor of the origin connected to it through a link which carries weight 10, then $S_i = \{10\}$ after the first step. At this point all the neighbors of the origin form what we call the “burning set” and the origin is deemed “extinguished”. Now the algorithm proceeds as follows. That member of the burning set, which has the lowest value of S_i assigned to it, is deemed extinguished, and the same burning procedure starts with the node i as the origin. The only difference is that we burn a node j if and only if its weight sequence S_j is larger than $S_i \cup \tau_{ij}$, where τ_{ij} is the weight of the link connecting nodes i and j and \cup denotes conjunction. If node j is already burning, we do not include it into the burning set again, but just replace its S_j with $S_i \cup \tau_{ij}$. This procedure guarantees that a node, once extinguished, can never again become part of the “burning set”. Moreover the weight sequence S_i for an extinguished node yields weight of the optimal path connecting it with the origin. Once Dijkstra algorithm is completed, we have a minimal spanning tree[15] constructed on our graph. Dijkstra algorithm in the strong disorder limit is also equivalent to that of invasion percolation [16,17].

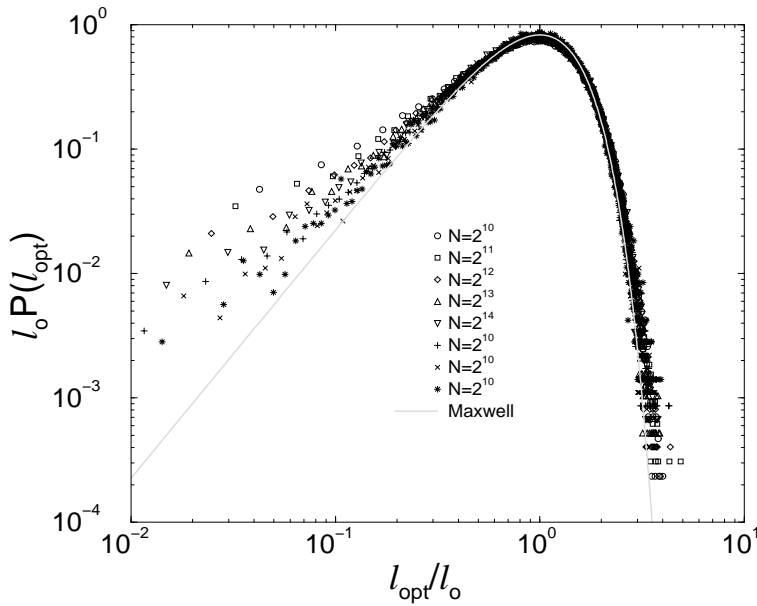


Fig. 2. Scaled curve for the probability distribution $P(\ell_{\text{opt}})$ of optimal path lengths for network sizes $N = 1024, 2048, 4096, 8192, 32768, 65536$. The gray curve represents Maxwellian fit given by Eq. (5).

Using this method, we obtain systems of sizes up to 4000 nodes, typically 10^5 realizations of disorder. We compute ℓ_{opt} by averaging the length of the optimal path for all the nodes of the configuration and over all realizations.

An alternative method of obtaining the optimal path in strong disorder is called the “bombing” algorithm [9]. We first choose a pair of nodes on the graph and begin removing links randomly, making sure that the connectivity between the two chosen nodes is not destroyed as each link is removed. The last path remaining is equivalent to the optimal path obtained by the ultrametric algorithm.

The bombing algorithm is slow, as one must test the connectivity after removal of each link. To improve the speed, we first find the minimal path in the network and then select links in random order. We remove the selected link from the graph. If the removed link belongs to the minimal path, we check if the connectivity between the two nodes is still present and recompute the new minimal path. If the connectivity between the two nodes is destroyed, we restore the link.

The advantage of this procedure is that one has to test for connectivity only if the selected link appears to belong to the minimal path. Since checking the connectivity is the most time consuming part in the original “bombing” algorithm, we could reach systems of sizes up to 2^{16} nodes with 10^5 realizations of weight disorder. Fig. 1 demonstrate that both algorithms yield very similar results, supporting the theoretical result $\langle \ell_{\text{opt}} \rangle \sim N^{1/3}$.

In Fig. 1, using the theoretical result $\nu_{\text{opt}} = 1/3$ we show numerical values of $\langle \ell_{\text{opt}} \rangle$ averaged over 10^5 realizations of disorder as a function of $N^{1/3}$. The linear behavior supports the theoretical value $\nu_{\text{opt}} = 1/3$.

We also study the probability distribution $P(l_{\text{opt}})$ of optimal path lengths on the network. The scaled curve for $P(l_{\text{opt}})$ for different network sizes is shown in Fig. 2 on a log-log plot. We find that there are two regimes in this distribution, the first one being a power law $P(l_{\text{opt}}) \sim (l_{\text{opt}})^\alpha$ which is evident from the figure, with $\alpha \approx 2$. The second regime is a stretched exponential $P(l_{\text{opt}}) \sim e^{-Cl_{\text{opt}}^\theta}$ where C is a constant and θ is around 2. This leads us to the expectation that the distribution may have a Maxwellian functional form:

$$P(l_{\text{opt}}) = \frac{4\ell_{\text{opt}}^2 e^{-(\ell_{\text{opt}}/l_o)^2}}{\sqrt{\pi}l_o^3}, \quad (5)$$

Where $l_o = \sqrt{\pi}\langle \ell_{\text{opt}} \rangle/2$ is the most probable value of ℓ_{opt} . The solid line in the figure is the plot of this function and as seen it agrees with our numerical results.

Finally, we repeated our simulations for the case in which disorder weights are associated with the nodes of the graph, and obtained the same scaling laws as for the disordered links case.

It should be pointed out that the above results concerning ℓ_{opt} practically do not depend on the average degree $\langle k \rangle$ of the random graph for large

$\langle k \rangle \gg 2$, and are the same even for the complete graph with $k = N - 1$. However, as we will see in the next section, the distribution of maximal weight τ_{\max} drastically depends on $\langle k \rangle$.

Probability distribution of the maximal weight on the optimal path

Now, we address another aspect of the problem, which is the probability distribution of the maximal weight τ_{\max} or equivalently the maximal random number ϵ_{\max} along the optimal path in a strongly disordered random network. As we mentioned earlier, the problem of the optimal path on a random graph in the strong disorder limit can be mapped onto a percolation problem on a Cayley tree with a degree distribution identical to the random graph and with a fraction p of its edges conducting. In order to further develop this analogy, we will show that the distribution of the maximal random number ϵ_{\max} along the optimal path can be expressed in terms of the order parameter $P_{\infty}(p)$ in the percolation problem on the Cayley tree, where $P_{\infty}(p)$ is the probability that randomly chosen site on the Cayley tree belongs to the infinite cluster. The motivation for the mapping on the Cayley tree comes from the following. Suppose A and B are two nodes in the random graph. Now, if we start our search for the optimal path between nodes A and B beginning at node A , then in the limit of the graph being of infinite size, the probability that we will visit a previously visited node after a finite number of steps tends to zero. Hence, we can assume that our search is equivalent to the search on a Cayley tree. If the original graph has a degree distribution p_k , the probability that we reach a node with a degree k by following a randomly chosen link on the graph, is equal to $kp_k/\langle k \rangle$, where $\langle k \rangle$ is the average degree. This is because the probability of reaching a given node by following a randomly chosen link is proportional to the number of links or the degree k of that node. Also, if we arrive at a node with degree k , the total number of outgoing branches is $k - 1$. Therefore, from the point of view of the Cayley tree, the probability to arrive at a node with $k - 1$ outgoing branches by following a randomly chosen link is $kp_k/\langle k \rangle$.

In the asymptotic limit, where the optimal path between the two points is very long, the probability distribution for the maximal weight link can be obtained from the following analysis. Let us assume that the probability of *not* reaching n generations starting from a given node of the Cayley tree whose edges conduct with a probability p , is Q_n . Suppose we are at a node whose outgoing degree is 2. Then the probability that starting from this node, we will not reach n generations of its descendants is the sum of three terms:

1. The probability that both the outgoing nodes are not conducting : $(1-p)^2$
2. The probability that both outgoing edges conduct, but the nodes reached by following them, do not have $n - 1$ generations of descendants : $p^2 Q_{n-1}^2$

3. The probability that one of the two outgoing edges conduct but the node reached by following the conducting edge does not have $n - 1$ generations of descendants : $2(1 - p)pQ_{n-1}$

Therefore, in this case[20]

$$Q_n(p) = (1 - p)^2 + p^2Q_{n-1}^2 + 2(1 - p)pQ_{n-1} \quad (6)$$

which on simplification becomes

$$Q_n(p) = ((1 - p) + pQ_{n-1})^2 \quad (7)$$

Following this argument for the case where a node has m outgoing edges, the probability that starting from this node, we can not reach n generations, is

$$Q_n(p) = ((1 - p) + pQ_{n-1})^m \quad (8)$$

Now in the case of a Cayley tree with a variable degree such as ours, we also have to incorporate a factor which accounts for the probability that the node under consideration has a given number of outgoing edges. Thus for a node on the Cayley tree, the probability that it does not have descendants in generation n can be obtained by applying a recursion relation

$$Q_l(p) = \sum_{k=1}^{\infty} p_k k ((1 - p) + pQ_{l-1})^{k-1} / \langle k \rangle \quad (9)$$

for $l = 1, 2, \dots, n$ with the initial condition $Q_0 = 0$, which indicates that a given node is always present in generation zero of its descendants.

For a random graph, a randomly chosen node the randomly chosen node has k outgoing edges with the original probability p_k . Thus it has a slightly different probability \tilde{Q}_n of not having descendants in its n th generation:

$$\tilde{Q}_n(p) = \sum_{k=1}^{\infty} p_k ((1 - p) + pQ_{n-1})^k \quad (10)$$

If we denote by $f_n(p)$, the probability that starting at a randomly chosen node we can reach, or survive up to, the n th generation, then

$$f_n = 1 - Q_n \quad (11)$$

and hence,

$$\tilde{f}_n = 1 - \sum_{k=1}^{\infty} p_k (1 - pf_{n-1})^k \quad (12)$$

while for $1 \leq l < n$

$$f_l = 1 - \sum_{k=1}^{\infty} p_k k (1 - pf_{l-1})^{k-1} / \langle k \rangle \quad (13)$$

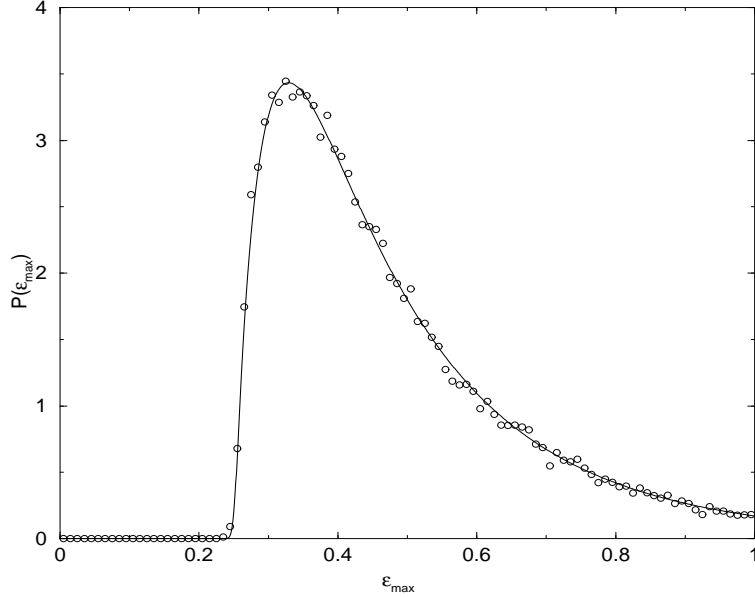


Fig. 3. The probability distribution of the maximal random number ϵ_{\max} along the optimal path obtained using simulations on a random graph with $\langle k \rangle = 4$ and using the analytical method on a Cayley tree with Poisson degree distribution and $\langle k \rangle = 4$. The simulations involve 100000 network realizations and are carried out on a network of 65536 nodes. The values of l_{opt} for this network lie in the range $40 < l_{\text{opt}} < 120$.

and $f_0 = 1$.

If n goes to infinity, this formula converges exponentially to the probability $f_\infty = P_\infty$ for a node to be connected to infinity for any p except for $p_c = \langle k \rangle / \sum_{k=1}^\infty p_k k(k-1)$, where the convergence is a power law. In the asymptotic limit of the optimal path problem, we have a pair of nodes separated by a very long path l_{opt} . The probability $\Pi(p)$, that they will be connected at given p , is in fact can be approximated by the probability that both of them are connected to the infinity and hence

$$\Pi(p) = P_\infty^2. \tag{14}$$

For the Poisson degree distribution $p_k = x^k e^{-x} / k!$, $P_\infty(p)$ satisfies

$$P_\infty = 1 - e^{-x p P_\infty}, \tag{15}$$

and

$$\Pi(p) = (1 - e^{-x p P_\infty})^2 / (1 - e^{-x})^2. \tag{16}$$

In the bombing algorithm, the largest random number on the path, ϵ_{\max} , is equal to the fraction of remaining bonds at which, for the first time the

connectivity would be lost, if we remove bonds in the descending order of ϵ . Thus the probability $P(\epsilon_{\max} \leq p) = \Pi(p)$, which is the probability that connectivity is not lost when a fraction p of bonds is removed. The probability density of the maximum random number ϵ on this path is thus equal to the derivative of this function with respect to p :

$$P(\epsilon_{\max}) = \frac{d}{dp} \Pi(p)|_{p=\epsilon_{\max}} \quad (17)$$

In Fig. 3 we plot two curves. The first curve is the true probability distribution of ϵ_{\max} in a strongly disordered random graph with $\langle k \rangle = 4$. The second curve shows the same probability distribution obtained using the analytical method described above using a Cayley tree approximation of the graph with a Poisson degree distribution and $\langle k \rangle = 4$. The curves coincide very well, indicating the excellent agreement between the theoretical analysis and simulation.

Acknowledgments We thank A.-L. Barabási for helpful discussions, and ONR for financial support.

References

1. R. Albert and A.-L. Barabási, *Rev. Mod. Phys.* **74** (2002) 47–97.
2. J. F. F. Mendes, S. N. Dorogovtsev, and A. F. Ioffe, *Evolution of Networks: From Biological Nets to the Internet and the WWW* (Oxford University Press, Oxford, 2003).
3. R. Pastor-Satorras and A. Vespignani, *Evolution and Structure of the Internet: A Statistical Physics Approach* (Cambridge University Press, in press).
4. P. Erdős and A. Rényi, *Publicationes Mathematicae* **6** (1959) 290–297.
5. B. Bollobás, *Random Graphs* (Academic Press, London, 1985).
6. R. Cohen and S. Havlin, *Phys. Rev. Lett.* **90** (2003) 058701.
7. I. Smaller, J. Machta, and S. Redner, *Phys. Rev. E* **47** (1993) 262.
8. P. van der Hofstad, G. Hooghiemstra, and P. van Mieghem, *Prob. Eng. Inf. Sciences* **15** (2001) 225.
9. M. Cieplak, A. Maritan, and J. R. Banavar, *Phys. Rev. Lett.* **72** (1994) 2320–2323; **76** (1996) 3754–3757.
10. L. A. Braunstein, S. V. Buldyrev, S. Havlin, and H. E. Stanley *Phys. Rev. E* **65** (2001) 056128.
11. A. Bunde and S. Havlin, eds., *Fractals and Disordered Systems* (Springer, New York, 1996).
12. P. Erdős and A. Rényi, *Publications of the Mathematical Institute of the Hungarian Academy of Sciences* **5** (1960) 17–61.
13. L. A. Braunstein, S. V. Buldyrev, R. Cohen, S. Havlin, and H. E. Stanley *Phys. Rev. Lett.* **91** (2003) 168701.
14. T. H. Cormen et al., *Introduction to Algorithms* (MIT Press, Cambridge MA, 1990).

15. R. Dobrin and P. M. Duxbury, *Phys. Rev. Lett.* **86** (2001) 5076.
16. A.-L. Barabási, *Phys. Rev. Lett.* **76**, 3750 (1996);
17. N. Schwartz, A. Nazaryev, and S. Havlin, *Phys. Rev. E* **58**, 7642 (1998);
18. R. Cohen, S. Havlin, and D. ben-Avraham, in *Handbook of Graphs and Networks*, edited by S. Bornholdt and H. G. Shuster (Wiley-VCH, New York, 2002), Chapt. 4.
19. R. Cohen, K. Erez, D. ben-Avraham and S. Havlin, *Phys. Rev. Lett.* **85** (2000) 4626.
20. A.-L. Barabási, S. V. Buldyrev, H. E. Stanley, and B. Suki, *Phys. Rev. Lett.* **76** (1996) 2192–2195.