Searching in small-world networks

Alessandro P. S. de Moura*
Instituto de Fisica, Universidade de São Paulo, Caixa Postal 66318, 05315-970 São Paulo, São Paulo, Brazil

Adilson E. Motter
Max Plank Institute for the Physics Complex Systems, Nöthnitzer Strasse 38, 01187 Dresden, Germany.

Celso Grebogi
Instituto de Física, Universidade de São Paulo, Caixa Postal 66318, 05315-970 São Paulo, São Paulo, Brazil
(Received 19 September 2002; published 11 September 2003)

We study the average time it takes to find a desired node in the Watts-Strogatz family of networks. We consider the case when the *look-up time* can be neglected and when it is important, where the look-up time is the time needed to choose one among all the neighboring nodes of a node at each step in the search. We show that in both cases, the search time is minimum in the small-world regime, when an appropriate distance between the nodes is defined. Through an analytical model, we show that the search time scales as $N^{1/D(D+1)}$ for small-world networks, where N is the number of nodes and D is the dimension of the underlying lattice. This model is shown to be in agreement with numerical simulations.

DOI: 10.1103/PhysRevE.68.036106 PACS number(s): 89.75.Hc, 84.35.+i, 87.23.Ge, 89.20.Hh

In recent years, the study of small-world networks has attracted much interest in many areas. Such networks are characterized by having a structure of links which is in between a completely regular and a totally random topology. A network is characterized by the number of nodes N and the average number of links per node κ [1]. One of the most important quantities for a network is its *length*, which is the average number of steps required to go from one node of the network to another, following the links. It is known [2] that for a given N and κ, a random network (i.e., whose links connect randomly chosen pairs of nodes) has a length L which is small and scales logarithmically with N. Regular networks, on the other hand, have large L, and L increases algebraically with the size N of the network. Watts and Strogatz [3] have shown that small-world networks, while presenting a structure very close to the regular ones, have lengths almost as small as those of equivalent random networks. The small-world phenomenon is extremely important, because many natural and artificial networks are found to be small worlds. Some examples are sociological networks [3–5], power transmission networks [3], biological networks [3], the Internet and the World-Wide Web [6], the structure of language [7], and many others. For reviews, see Refs. [8,9].

In this paper, we investigate the problem of *searches* in networks, which is of great relevance for many systems, in particular for communication networks. Previous works on this subject have centered on scale-free networks [10], lattices with long-range links connected according to a distance-dependent probability distribution [11], congestion-sensitive networks [12], models of the World-Wide Web [13], and social networks [5,14]. In these works, it was found that the topology of the network may have a large effect on the typical search time. Here we use the Watts-Strogatz network family [3] to study the search time, and how it is affected as the network's topology changes smoothly from regular to random. To be more specific, if we start at an initial node a_s of the network, how long does it take to get to a looked-for node a_f, by jumping from node to node through the network's links? The answer depends on how each node is chosen from the previous one. This in turn depends on what knowledge is available about the network's structure. We shall focus on networks on which a distance measure $d(a,b)$ is defined for pairs of nodes (a,b), which indicates how "close" the nodes a and b are. d contains the available knowledge about the network. We assume that closer nodes are more likely to be linked. For instance, in the case of a packet being routed through the Internet, one is searching for a specific IP address, and $d(a,b)$ would be some suitable measure of how close the two IP addresses a and b are. During the search process, one usually has only local information to decide the next step: if the current node in the search is a_i, we assume that all it is known is the neighborhood $\Gamma(a_i)$ of a_i, which is the set of all nodes a_i is linked to. In our model, the next node a_{i+1} in the search is chosen by taking from $\Gamma(a_i)$ the node which is the closest to the looked-for node a_f [15]. This procedure models how search works in many real situations. In the case of the IP address search mentioned above, at each router the packet goes on to the next available router whose IP address is closest to the desired one. The same process can also be used to model other searches, such as the classic letter-sending experiment by Milgram [4], subject search in the World-Wide Web, and, possibly, retrieval of information in associative memories.

For concreteness, we first consider the Watts-Strogatz ring network [3] to study how the transition of the network from regular to random affects the average search time. In previous works it was assumed that the time it takes to look at each neighbor of the present node (we call it the *look-up time*) is negligible compared to the time it takes to jump to the next node. In some real situations, however, the look-up time can be of the same order as the hopping time, such as,

*Electronic address: amoura@if.usp.br

for instance, in a data structure used by a computer program. In this paper, the latter case is considered for the first time, to the best of our knowledge. We denote the corresponding search time by L_s^{LUT}. The usual definition of search time, which is the average number of steps it takes to reach a desired node, is denoted by L_s. The search procedure is as described above, and the distance function d is taken to be the distance along the ring between the two nodes. In particular, we are interested in how L_s and L_s^{LUT} behave in the small-world region situated in between the regular and random limits. It is important to notice that L_s (and also L_s^{LUT}) is different from the length L. They coincide only if the best path connecting any two nodes were always chosen in the search. However, since at each point in the search we have only local information, we do not know beforehand what the best path is. Therefore, we have $L_s \geqslant L$. We find that L_s reaches a minimum in the small-world region. The reason is that for the regular network, due to the absence of long-range links, L is large, and L_s is also large, since $L_s \geqslant L$. In the random limit, on the other hand, since the links connect randomly chosen nodes, the distance d is no longer useful in determining how close we are to the target node, and the search will jump from node to node in a random way, taking a long time to find the target node. Therefore, the search will be fastest for networks whose structures are in between regular and random; in other words, for small-world networks. L_s^{LUT} also attains its minimum in the small-world region. This is in sharp contrast to the behavior of the length L, which is minimum in the random limit. A very important issue is how L_s and L_s^{LUT} scale with the number of nodes N. We develop an analytical model to predict this, and we show that both L_s and L_s^{LUT} scale with N as $L_s \sim N^{1/2}$ in the small-world region. This means that in small-world networks, the search time increases slowly as the size grows, as opposed to regular and random networks, where it increases linearly with N. It is especially important that L_s^{LUT} also scales as $N^{1/2}$ in the small-world regime, since it is well known that the search time always scales linearly with N if no information on the structure of the network is available. In search methods such as the one used in Ref. [10], which operate without information on the network's structure, even though L_s may be dramatically decreased, L_s^{LUT} always grows linearly with N, whereas it only increases as $N^{1/2}$ in our case, because of the knowledge embodied in the distance function d about the network.

To illustrate our results, we now focus on the Watts-Strogatz ring network, which is built from a one-dimensional periodic lattice, where N nodes are placed on a ring and each node is linked to its $2k$ nearest neighbors (k neighbors on each side). Then each link is rewired randomly with probability β, with $0<\beta<1$ parametrizing the network. $\beta=0$ corresponds to the original ring network, which is completely regular, and $\beta=1$ gives a random network. By changing β, we move smoothly from regular to random networks. We assume a *sparse* network, with $k \ll N$.

We first investigate how L_s changes as β goes from 0 to 1. The behavior of L_s^{LUT} is studied below. L_s is evaluated by applying the search algorithm for pairs of randomly chosen initial and target points. We thus get L_s as a function of the parameter β, as shown in Fig. 1. In the same figure we also show $L(\beta)$ for comparison. We see that L_s drops sharply from the value of the regular network, as β is increased from 0. It reaches a minimum level at small values of the randomness parameter, where it stays practically unchanged for a wide interval of β. It then rises again, as β approaches 1, the random limit. This behavior can be understood as follows. For $\beta=0$, the search proceeds from the initial node by jumping to the nearest neighbor node closest to a_f, which is at a distance k away, and so on repeatedly. It is easily seen that in this case, $L_s \sim N/k$. For $k \ll N$, $L_s(0)$ is thus very large. As β increases, however, some links are rewired, and long-range links appear in the network, which can play the role of shortcuts in the search process. If β is only slightly larger than 0, most nodes are still linked only to their nearest neighbors, and the search proceeds for some time by jumping to nearest neighbors, as before. But eventually it gets to a node that has a long-range rewired link, through which we may get closer to a_f then by jumping to the k-nearest neighbor. Therefore, for $\beta>0$, but not too large, L_s decreases, due to the presence of long-range links in the network. We will show analytically that L_s scales as $N^{1/2}$ in this case. As β increases further and approaches 1, the links to nearest neighbors are almost all rewired into long-range links. Since the rewiring is random, nodes are linked in a fashion that is uncorrelated with their position in the ring. As a result, the distance function d no longer gives useful information for the search. For $\beta=1$, a search can be approximated as a random sequence of nodes, and it can readily be shown that L_s also scales linearly with N, such as in the regular case.

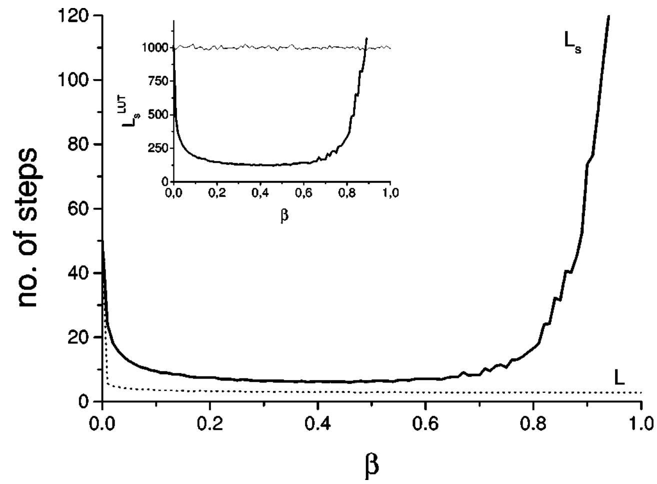

FIG. 1. Search time L_s as a function of the randomness parameter β (full line), for $N=2000$ and $k=10$. Each point corresponds to an average over 1000 pairs of nodes, for 20 independent realizations of the network. For comparison, the length L is also plotted (dashed line). The inset shows L_s^{LUT} as a function of β with our method (thick line) and the breath-first method (thin line).

Although we have used the Watts-Strogatz ring network for illustration, it is clear that the same qualitative reasoning can be applied to any network where a meaningful distance function can be defined, and where closer nodes are more likely to be connected. This assumption holds for many real

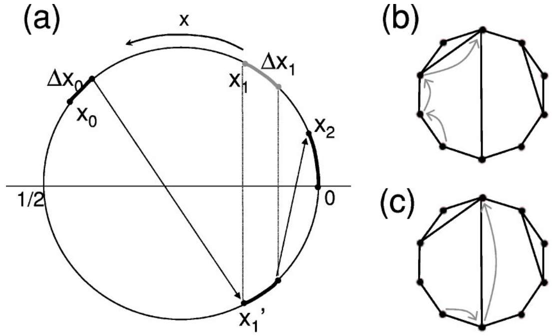

FIG. 2. (a) Illustration of the search on the Watts-Strogatz network. At $x_0-\Delta x_0$, the first shortcut is found, linking to x_1', which can be considered to be reflected through the horizontal axis, on x_1 (see text). (b) shows a possible search in a small network, and (c) shows the least path connecting the same pair of nodes. This exemplifies the difference between L and L_s.

networks. In general, search in small-world networks is expected to be faster than in regular or random networks.

We now present an analytical model to predict how L_s scales with N, k, and β, in the small-world region. We assume $N \gg 1$, $k \ll N$. We also assume $0<\beta \ll 1$, so that most links connect nearest neighbors, and the density of long-range links is low. We consider, however, that the total number of long-range links $M=Nk\beta$ is large, $M \gg 1$. To simplify the analysis, we take the ring on which the nodes are placed to have unit circumference, and each node is described by its position x in the circle, with $0 \leqslant x<1$. Since N is large, we can consider x to be a continuous variable. We take $x=0$ to correspond to the target node a_f, and x is measured counterclockwise (see Fig. 2).

Suppose the search starts at a node a_s, corresponding to $x=x_0$. Since the density of long-range links is small, it is likely that the search proceeds by jumping to the kth-nearest neighbors a few times before finding a shortcut. This part of the search can be represented as a continuous segment in the circle, of size Δx_0. Since $M \gg 1$, we have $\Delta x_0 \ll 1$ on average. At some point, the search finds a shortcut, and jumps to x_1, and then again goes a distance Δx_1 by jumping to the kth nearest neighbors, finds another shortcut, and so on (see Fig. 2). In this analysis, we are only concerned about the *distance* from x to the origin 0, where the target node is. This allows us to use only half of the circle $0 \leqslant x \leqslant 1/2$, and consider points on the other half to be reflected by the diameter through the origin, as shown in Fig. 2. Thus, the search can be represented as a sequence of pieces on the segment S given by $0 \leqslant x \leqslant 1/2$, which represents directly the distance to the target node.

Let us first consider the jump from $x_i-\Delta x_i$ to x_{i+1}. Since a long-range link is random, it may be connected with any other node with uniform probability. To work as a shortcut for the search, the link must be connected to a node closer to a_f than the node corresponding to $x_i-\Delta x_i$. Therefore, x_{i+1} has a uniform probability distribution on $0 \leqslant x < x_i-\Delta x_i$. We use the average for x_{i+1}, and neglect the term $\Delta x_i/2$ as being much smaller than the size of the jump. We then have $x_{i+1} \approx x_i/2$. So on average, x is halved at each shortcut. Since $x_0=1/4$, on average, the expected value for x_i is

$$x_i=2^{-i-2}, \tag{1}$$

where x_i is the distance from the target after the ith shortcut found. The search ends for a value n such that x_n gets close enough to the origin, so that

$$\Delta x_n \geqslant x_n. \tag{2}$$

When this condition is satisfied, the search reaches $x=0$, and finds the target node.

To calculate Δx_i, we start by noticing that the probability of finding a long-range link on a piece $dx \ll 1$ of the segment L is proportional to the number of shortcuts M and to dx. However, not all long-range links help the search: to do so, they must link to a node with coordinate smaller than the current node's coordinate x. Since the rewired links are connected according to a uniform probability distribution on the circle, the probability that a long-range link *is* a shortcut for the search is proportional to the distance x to the target node. Finally, since most of the search is done in jumps of size k (except at shortcuts and at nodes whose link to the kth neighbor has been rewired), the probability of finding a shortcut in the length dx of S is inversely proportional to k. From the above, the probability dP of finding a shortcut in a length dx of S is

$$dP=2CN\beta x|dx|=CN\beta|dx^2|, \tag{3}$$

where $C>0$ is a constant and we use $M=Nk\beta$. This expression is valid for an infinitesimal displacement dx. For a finite displacement starting at x and ending at x_b, with $x_b<x$, we get from Eq. (3) the probability P of finding a shortcut:

$$P(x,x_b)=1-\exp[-CN\beta(x^2-x_b^2)]. \tag{4}$$

From this equation, we can roughly estimate the average value x_b at which a shortcut will be found by putting $CN\beta(x^2-x_b^2)=1$, which yields to first order $x_b \approx x-\Delta x$, with $\Delta x=1/2CN\beta x$, where we assume $N\beta \gg 1$. Substituting Eq. (1) into this expression, we get the following for Δx_i:

$$\Delta x_i=2^{i+1}/CN\beta. \tag{5}$$

Now, by substituting Eqs. (1) and (5) into Eq. (2), we get an expression of the total number n of shortcuts crossed in the search:

$$2^{2n}=CN\beta/8. \tag{6}$$

The total length l_s crossed in the ring throughout the search is thus given by $l_s=\Sigma_{i=0}^{n}\Delta x_i$. Using Eq. (5) and using $x_0=1/4$ as before, we get

$$l_s=\frac{2}{CN\beta}\sum_{i=0}^{n}2^i \approx \frac{4}{CN\beta}2^n, \tag{7}$$

where we assume $n \gg 1$. Substituting Eq. (6) into this expression, we get $l_s=K(N\beta)^{-1/2}$, where $K=(2C)^{-1/2}$ is a constant. l_s gives the search length on the unit circle. We now need the search time L_s in terms of the number of nodes. They are related by $L_s=Nl_s/k$. So our final result is

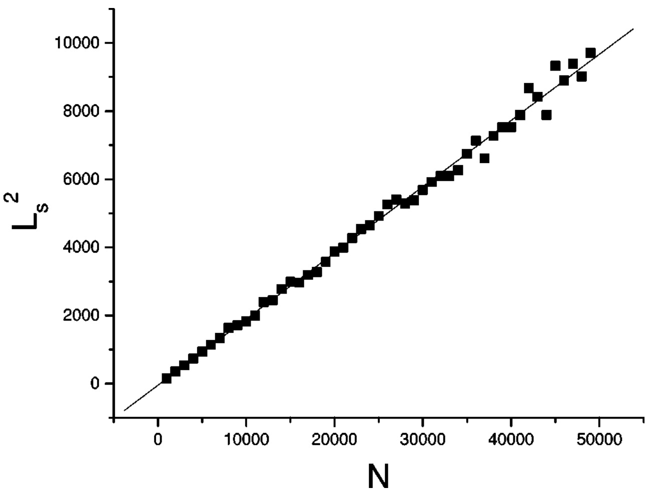

FIG. 3. Square of the search time L_s as a function of the network size N, for $\beta = 0.02$ and $k = 10$. Each point is an average over 1000 pairs of nodes, for 20 different realizations of the network.

$$L_s = K\left(\frac{N}{k^2\beta}\right)^{1/2}. \tag{8}$$

Equation (8) predicts that L_s scales as $N^{1/2}$. This means that the average search time for small-world networks increases very slowly with the network size, in comparison with regular and random networks, and also with blind search methods. To verify this scaling law, we calculate L_s for networks with several different values of N. The result for $k = 10$ and $\beta = 0.02$ is shown in Fig. 3. We see that the scaling $L_s \sim N^{1/2}$ is indeed confirmed by the simulation. The same is true for other values of k and β, provided that $k \ll N$, $0 < \beta \ll 1$, and $N\beta \gg 1$. The scalings with respect to k and β predicted by Eq. (8) were also verified numerically.

From the point of view of design, small worlds are the best choice for searching in large networks, and they become increasingly better as size increases. Indeed, the ratio $L_s^{sw}/L_s^{reg,rand}$ between the search times for small-world and regular or random networks (or blind search methods) scales with $N^{-1/2}$.

L_s measures the search time if the process of looking up each neighbor at each step takes a much shorter time than hopping from one node to the next. In some cases, however, we have to consider the "look-up time" in the calculation of the search time. We denote the corresponding search time by L_s^{LUT}. We now calculate L_s^{LUT}, assuming that the look-up time is the same as the "hopping time." Since the number of links at each node is independent of its distance from the target node, L_s^{LUT} is just equal to L_s multiplied by the average number of links per node $2k$ [16]. From Eq. (8) then we have

$$L_s^{\mathrm{LUT}} \approx 2K(N/\beta)^{1/2}. \tag{9}$$

This means that L_s^{LUT} also scales with the size as $N^{1/2}$, and small-world networks are also efficient for searches where the look-up time cannot be neglected. This is not true if there is no information about the network's structure. To directly verify this, we calculated numerically $L_s^{\mathrm{LUT}}(\beta)$ for our search method, and for the breath-first method, which is a "blind" method. The results are shown in the inset of Fig. 1. We see that L_s^{LUT} has indeed a minimum in the small-world region with our method, whereas it stays constant for the breath-first method. Although in Ref. [10] the authors show that in scale-free networks L_s depends logarithmically on N, we emphasize that if the look-up times are considered, the search time scales linearly with N. This happens because even though one goes through only a few nodes to reach the target, these nodes are the ones that are highly connected, and there are more links to look up than average.

We notice that, from Fig. 1, it is clear that both L_s and L_s^{LUT} stay near the minimum for a wide range of β. This means that our search method has a good performance for a wide range of network topologies.

In the model presented above, we considered the Watts-Strogatz ring network. However, it is clear that the search in general small-world networks will also be sped up. In particular, we have extended our theory to higher-dimensional networks of the Watts-Strogatz family. We found that the search time for both L_s and L_s^{LUT} in a D-dimensional small-world network scales as $N^{1/D(D+1)}$, which should be compared with $N^{1/D}$ for regular networks, and linear scaling for random ones. For $D = 2$ the scaling is $N^{-1/6}$, which is potentially relevant to understand the shortness of the letters' paths in the classical Milgram experiment [4,5]. We observe that the corresponding scaling in the Kleinberg model is $N^{-1/3}$ [11].

Summarizing, we have shown that small-world networks optimize search times. We have also predicted analytically for small worlds how the average search time scales with the various quantities characterizing the network. For networks embedded in a D-dimensional space, our mean-field theory predicts that the search time scales as $N^{1/D(D+1)}$, valid also for the cases in which the look-up time cannot be neglected. In particular, for the Watts-Strogatz one-dimensional model, the search time scales as $N^{1/2}$. These predictions were confirmed by the direct simulation of the search. Although we have used a particular family of networks in the analysis, in general networks with small-world topologies minimize search times. Since search is a problem of extreme importance for so many natural and artificial networks, we expect our results to be relevant for many areas of knowledge.

We conclude with some thoughts on the meaning of our results for learning. From the standpoint of retrieval of information in an associative memory, the small-world property of the network represents a maximization of efficiency because similar pieces of information are stored together, which makes searching by association possible, and because even very different pieces of information are never separated by more than a few links, which guarantees a fast search. If we assume that the human memory is associative, we may speculate that the dendritic wiring in learning processes is mainly driven by the minimization of the search time in the underlying network. This in turn may be related to the topology of the neural network itself, which has been argued to be a homogeneous small-world network [17].

This work was supported by FAPESP and CNPq.

[1] B. Bollobás, *Modern Graph Theory* (Springer-Verlag, New York, 1991).
[2] D.J. Watts, *Small Worlds* (Princeton University Press, Princeton, NJ, 1999).
[3] D.J. Watts and S.H. Strogatz, Nature (London) **393**, 440 (1998).
[4] S. Milgram, Psychol. Today **1**, 61 (1967).
[5] D.J. Watts, P.S. Dodds, and M.E.J. Newman, Science **296**, 1302 (2002).
[6] R. Albert, H. Jeong, and A.-L. Barabási, Nature (London) **401**, 130 (1999).
[7] A.E. Motter, A.P.S. de Moura, Y.-C. Lai, and P. Dasgupta, Phys. Rev. E **65**, 065102 (2002).
[8] S.H. Strogatz, Nature (London) **410**, 268 (2001).
[9] R. Albert and A.-L. Barabasi, Rev. Mod. Phys. **74**, 47 (2002).
[10] B.J. Kim, C.N. Yoon, S.K. Han, and H. Jeong, Phys. Rev. E **65**, 027103 (2002); L.A. Adamic, R.M. Lukose, A.R. Puniyani, and B.A. Huberman, *ibid.* **64**, 046135 (2001).
[11] J.M. Kleinberg, Nature (London) **406**, 845 (2000).
[12] R. Guimerà, A. Díias-Guilera, F. Vega-Redondo, A. Cabrales, and A. Arenas, Phys. Rev. Lett. **89**, 248701 (2002).
[13] F. Menczer, Proc. Natl. Acad. Sci. U.S.A. **99**, 14014 (2002).
[14] D.R. White and M. Houseman, Complexity **8**, 72 (2003).
[15] It is possible that the search gets ‘‘stuck’’ at some point; for instance, if the closest node in $\Gamma(a_{i+1})$ is a_i, we would keep jumping between the nodes a_i and a_{i+1} forever. To prevent this, we establish that if the same link in the network is about to be crossed twice, the next node is chosen randomly in the network, and then it proceeds as before.
[16] The average number of links in the nodes visited by the searching process is actually slightly larger than $2k$, because nodes with larger number of links are more likely to be visited. We have neglected this difference, which is very small in the case of the Watts-Strogatz network.
[17] T. Nishikawa, A. E. Motter, Y.-C. Lai, and F. C. Hoppensteadt, Phys. Rev. Lett. **91**, 014101 (2003).