8

Walking and searching on networks

The navigation and exploration of complex networks are obviously affected by the underlying connectivity properties and represent a challenging scientific problem with a large number of practical applications. The most striking example is the World Wide Web (WWW), an immense data repository which acquires a practical interest and value only if it is possible to locate the particular information one is seeking. It is in this context and having in mind large-scale information and communication technologies (ICTs) that most of the models and studies are developed. Because information discovery and retrieval rely on the understanding of the properties of random walks and diffusion phenomena in complex networks, considerable activity has been devoted to the investigation of these basic processes.

In this chapter we will review the main strategies that can be used to explore and retrieve information from the vertices of a network. In particular, we want to expose how the topological properties of networks might affect search and retrieval strategies and how these strategies can effectively take advantage of the network's connectivity structure. We start from basic considerations on diffusion processes taking place on networks and naturally dive into more applied works that consider well-defined problems encountered in the ICT world.

8.1 Diffusion processes and random walks

The discovery process and the related algorithms in large-scale networks are clearly related to the properties of a random walk diffusing in a network of given topology. In other words, the simplest strategy to explore a network is to choose one node, to follow randomly one of the departing links to explore one of its neighbors, and to iterate this process until the required information is found or a satisfactory knowledge of the network connectivity pattern is acquired. It is clear that a random

https://doi.org/10.1017/CBO9780511791383.009 Published online by Cambridge University Press

walk lies underneath this simple strategy, and in order to improve the search and navigation a good understanding of this basic process is necessary.

As a starting point in the study of diffusion processes in complex networks, we first analyze the simple unbiased random walk in an undirected network with given connectivity pattern. The basic properties of this process can be understood as the random diffusion of W walkers (or particles) in a network made of N nodes. We can consider that each node i has an occupation number W_i counting the number of walkers present on it. The total number of walkers in the network is $W = \sum_i W_i$ and each walker diffuses along the edges with transition rates that depend on the node degree and other factors. In the case of a simple Markovian random walk, a walker located on a node i diffuses out of i along an edge (i, j) with a rate given by

$$d_{ij} = \frac{r}{k_i}, \tag{8.1}$$

where k_i is the degree of i. This relation simply defines a uniform rate of diffusion along any one of the edges departing from the node i, and corresponds to a total rate of escape $\sum_{j \in \mathcal{V}(i)} d_{ij}$ from any node equal to r.

The simplicity of random walk processes allows for the derivation of detailed analytical results. Before describing some of them, we take this opportunity to illustrate a general method of wide applicability in the context of heterogeneous networks. In Chapters 1 and 2, we have seen that it is convenient to group the nodes in degree classes for the statistical characterization of networks. The assumption of statistical equivalence of nodes with the same degree is in fact a very general tool that applies in the study of many dynamical processes, such as epidemics or opinion dynamics (Chapters 9 and 10). In this framework, we consider that nodes are only characterized by their degree and that their statistical properties are the same as long as their degree is the same. This assumption allows the use of degree block variables defined as

$$W_k = \frac{1}{N_k} \sum_{i | k_i = k} W_i \, , \tag{8.2}$$

where N_k is the number of nodes with degree k and the sum runs over all nodes i having degree k_i equal to k. The variable W_k represents the average number of walkers in nodes within the degree class k and conveniently allows the wide range of degrees present in the system to be taken into account. We now consider a network with degree distribution $P(k)$ in which the walkers move from a node with degree k to another with degree k' with a transition rate r/k. The dynamics of walkers is simply represented by a mean-field dynamical equation expressing the

https://doi.org/10.1017/CBO9780511791383.009 Published online by Cambridge University Press

variation in time of the walkers $W_k(t)$ in each degree class. This can easily be written as:

$$\partial_t W_k(t) = -r W_k(t) + k \sum_{k'} P(k'|k) \frac{r}{k'} W_{k'}(t). \tag{8.3}$$

The first term on the right-hand side of this equation just considers that walkers move out of the node with rate r. The second term accounts for the walkers diffusing from the neighbors into the node of degree k. This term is proportional to the number of links k times the average number of walkers coming from each neighbor. This is equivalent to an average over all possible degrees k' of the conditional probability $P(k'|k)$ of having a neighbor of degree k' times the fraction of walkers coming from that node, given by $W_{k'}(t)r/k'$. In the following we consider the case of uncorrelated networks in which the conditional probability does not depend on the originating node and for which it is possible to use the relation $P(k'|k) = k'P(k')/\langle k\rangle$ (see Chapter 1). The dynamical rate equation (8.3) for the subpopulation densities reads then as

$$\partial_t W_k(t) = -r W_k(t) + \frac{k}{\langle k\rangle} \sum_{k'} P(k') r W_{k'}(t). \tag{8.4}$$

The stationary condition $\partial_t W_k(t) = 0$ does not depend upon the diffusion rate r which just fixes the time scale at which the equilibrium is reached and has the solution

$$W_k = \frac{k}{\langle k\rangle} \frac{W}{N}, \tag{8.5}$$

where we have used that $\sum_k P(k)W_k(t) = W/N$ represents the average number of walkers per node that is constant. The above expression readily gives the probability $p_k = W_k/W$ to find a single diffusing walker in a node of degree k as

$$p_k = \frac{k}{\langle k\rangle} \frac{1}{N}. \tag{8.6}$$

This last equation is the stationary solution for the visiting probability of a random walker in an uncorrelated network with arbitrary degree distribution. As expected, the larger the degree of the nodes, the larger the probability of being visited by the walker. In other words the above equation shows in a simple way the effect of the diffusion process that brings walkers, with high probability, into well-connected nodes, thus showing the impact of a network's topological fluctuations on the diffusion process.

As previously noted, the simplicity of the diffusion process allows for much more detailed analysis and results. In particular, it is possible to identify specific

https://doi.org/10.1017/CBO9780511791383.009 Published online by Cambridge University Press

network structures where powerful analytical techniques can be exploited to find rigorous results on the random walk properties and other physical problems such as localization phenomena and phase transitions (see for instance the review by Burioni and Cassi [2005]). In the case of complex networks with arbitrary degree distribution, it is actually possible to write the master equation for the probability $p(i, t|i_0, 0)$ of a random walker visiting site i at time t (starting at i_0 at time $t = 0$) under the form

$$\partial_t p(i, t|i_0, 0) = -\left(\sum_j x_{ij} d_{ij}\right) p(i, t|i_0, 0) + \sum_j x_{ji} d_{ji} p(j, t|i_0, 0), \tag{8.7}$$

where x_{ij} is the adjacency matrix of the network. The first term on the right-hand side corresponds to the probability of moving out of node i along the edges connecting i to its neighbors, while the second term represents the flux of walkers arriving from the neighbors of i. With the choice of transition rates given by Equation (8.1), one can obtain explicitly the probability for a walker to be in node i in the stationary limit of large times (Noh and Rieger, 2004) as

$$p_i^\infty = \frac{k_i}{\langle k \rangle} \frac{1}{N}, \tag{8.8}$$

showing again the direct relation between the degree of a node and the probability of finding random walkers in it.

Another particularly important quantity for diffusion processes is the return probability $p_0(t)$ that a walker returns to its starting point after t steps, and is directly linked to the eigenvalue density $\rho(\lambda)$ (also called the spectral density) of the modified Laplacian operator associated with this process[1]

$$L'_{ij} = \delta_{ij} - \frac{x_{ij}}{k_j}. \tag{8.9}$$

As shown in Appendix 5, these two quantities are related by the equation

$$p_0(t) = \int_0^\infty \mathrm{d}\lambda \mathrm{e}^{-\lambda t} \rho(\lambda), \tag{8.10}$$

so that the behavior of $p_0(t)$ is connected to the spectral density, and in particular its long time limit is directly linked to the behavior of $\rho(\lambda)$ for $\lambda \to 0$. Notably, this problem is relevant not only to the long time behavior of random walks but also to many other processes including synchronization or signal propagation (Samukhin, Dorogovtsev and Mendes, 2008).

[1] We emphasize that this modified Laplacian is different from the Laplacian discussed in Chapter 7 and Appendix 4, each column being divided by k_j: $L_{ij} = k_j L'_{ij}$.

https://doi.org/10.1017/CBO9780511791383.009 Published online by Cambridge University Press

Before turning to the case of complex networks, let us first consider some simple examples. For a D-dimensional regular lattice, the low eigenvalue behavior of the spectral density is given by (Economou, 2006) $\rho_D(\lambda) \sim \lambda^{D/2-1}$, which, using Equation (8.10), leads to the well-known result

$$p_0(t) \sim t^{-D/2}. \tag{8.11}$$

For the case of a random Erdős–Rényi graph, Rodgers and Bray (1988) have demonstrated that for $\lambda \to 0$ the spectral density has the form $\rho_{\mathrm{ER}}(\lambda) \sim \mathrm{e}^{-c/\sqrt{\lambda}}$ which leads to the long time behavior of the form

$$p_0(t) \sim \mathrm{e}^{-at^{1/3}} \tag{8.12}$$

where a and c are constants depending on the specific network. In the case of a Watts–Strogatz network for which shortcuts are added to the regular lattice, Monasson (1999) has shown that the eigenvalue density can be written as the product of ρ_D and ρ_{ER} leading to

$$\rho_{\mathrm{WS}}(\lambda) \sim \lambda^{D/2-1}\mathrm{e}^{-p/\sqrt{\lambda}}, \tag{8.13}$$

where p is the density of shortcuts. This form implies the following behavior for the return probability

$$p_0(t) - p_0(\infty) \sim \begin{cases} t^{-D/2} & t \ll t_1 \\ \exp(-(p^2t)^{1/3}) & t_1 \ll t \end{cases}, \tag{8.14}$$

where $p_0(\infty) = 1/N$ and the crossover time is $t_1 \sim 1/p^2$. In the first regime, the diffusion is the same as on a D-dimensional lattice with a typical behavior scaling as $1/t^{D/2}$. Accordingly, the number of distinct nodes visited is $N_{\mathrm{cov}} \sim t^{D/2}$ (Almaas, Kulkarni and Stroud, 2003). After a time of order t_1, the walkers start to feel the effect of the shortcuts and reach a regime typical of the Erdős–Rényi network with a stretched exponential behavior $\exp(-t^{1/3})$ and $N_{\mathrm{cov}} \sim t$. At very long times, the number of distinct nodes saturates at the size of the network $N_{\mathrm{cov}} \sim N$ (see Figure 8.1).

Interestingly, Samukhin *et al.* (2008) show that the spectral density for uncorrelated random networks with a local tree-like structure is rather insensitive to the degree distribution but depends mainly on the minimal degree in the network (in the infinite size limit). In particular, for uncorrelated random scale-free networks with minimum degree m equal to one or two, the spectral density is given by

$$\rho(\lambda) \sim \lambda^{-\xi}\mathrm{e}^{-a/\sqrt{\lambda}}, \tag{8.15}$$

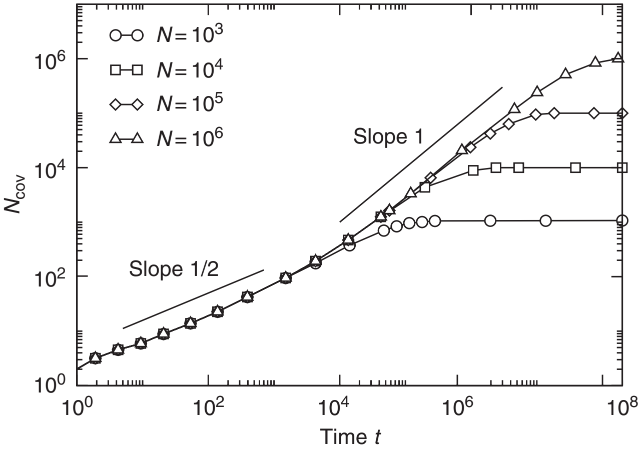

Fig. 8.1. Average number of visited distinct sites for a random walk on a (one-dimensional) Watts–Strogatz network with shortcut density $p = 0.01$. At short times, $N_{\mathrm{cov}} \sim t^{1/2}$, and for longer times, $N_{\mathrm{cov}} \sim t$. In the last regime, the coverage number saturates to the size N of the network. Data from Almaas *et al.* (2003).

where a is a constant, $\xi = 9/10$ for $m = 1$ and $\xi = 4/3$ for $m = 2$. The return probability on scale-free networks is thus different from the Watts–Strogatz case and turns out to be given by

$$p_0(t) \sim t^{\eta} \mathrm{e}^{-bt^{1/3}}, \tag{8.16}$$

with $\eta = -7/30$ for $m = 1$ and $\eta = 1/18$ for $m = 2$ (Samukhin *et al.*, 2008).

Many other quantities can be investigated both numerically and analytically, and we refer the interested reader to previous work (Gallos, 2004; Kozma, Hastings and Korniss, 2005; Sood, Redner and ben Avraham, 2005; Bollt and ben Avraham, 2005; Kozma, Hastings and Korniss, 2007). For example, the average time to visit or to return to a node is inversely proportional to its degree (Noh and Rieger, 2004; Baronchelli and Loreto, 2006). All these results confirm the importance of hubs in diffusion processes and show how scale-free topologies have an important impact on the dynamics, highlighting the relevance of topological heterogeneity. Finally, the diffusion process equations can also be generalized to the case of weighted networks and more complicated diffusion schemes (Wu *et al.*, 2007; Colizza and Vespignani, 2008), but in all cases the degree variability plays an important role and alters the stationary visiting or occupation probability, favoring high degree nodes. This has a noticeable impact on most of the algorithms aimed at navigating and searching large information networks, as we will show in the next sections.

https://doi.org/10.1017/CBO9780511791383.009 Published online by Cambridge University Press

8.2 Diffusion in directed networks and ranking algorithms

The basic considerations of the previous section are at the core of one of the most celebrated applications of the internet world: the PageRank algorithm. This algorithm has been the winning feature of the search engine “Google” and represented a revolution in the way we access information on the WWW.

In order to explore and index web pages, search engines rely on automatic programs called *web crawlers* that follow a list of links provided by a central server or follow recursively the links they find in the pages that they visit, according to a certain set of searching instructions. When a crawler finds a new web page in its search, it stores the data it contains and sends it to a central server. Afterwards, it follows the links present in the page to reach new websites. Web crawlings are repeated at periodic time intervals, to keep the index updated with new pages and links. The information retrieved by the crawlers is analyzed and used to create the index. The index stores information relative to the words present in the web pages found, such as their position and presentation, forming a database relating those words with the relevant hyperlinks to reach the pages in which they appear, plus the hyperlinks present in the pages themselves. The final element in a search engine is the user interface, a search software that accepts as an input words typed by the user, explores the index, and returns the web pages that contain the text introduced by the end user, and are considered as most relevant. In this process, important information is given by the *ranking* of the pages returned, i.e. the order in which they are presented after the query. Obviously, nobody is willing to visit dozens of uninteresting pages before discovering the one that contains the particular information that is sought. Therefore, the more relevant the first page returns are, the more successful and popular will the search engine be. The search engines available in the market make use of different ranking methods, based on several heuristics for the location and frequency of the words found in the index. Traditionally, these heuristics combine information about the position of the words in the page (the words in the HTML title or close to the top of the page are deemed more important than those near the bottom), the length of the pages and the meaning of the words they contain, the level of the directory in which the page is located, etc.

The PageRank algorithm is in this respect a major breakthrough based on the idea that a viable ranking depends on the topological structure of the network, and is provided by essentially simulating the random surfing process on the web graph. The most popular pages are simply those with the largest probability of being discovered if the web-surfer had an infinite time to explore the web. In other words “Google” defines the importance of each document by a combination of the probability that a random walker surfing the web will visit that document, and some heuristics based in the text disposition. The PageRank algorithm just gauges the

https://doi.org/10.1017/CBO9780511791383.009 Published online by Cambridge University Press

importance of each web page i by the PageRank value $P_R(i)$ which is the probability that a random walker surfing the web graph will visit the page i. According to the considerations of the previous section it is clear that such a diffusion process will provide a large P_R to pages with a large degree, as the random walker has a much higher probability of visiting these pages. On the other hand, in the web graph we have to take into account the directed nature of the hyperlinks. For this reason, the PageRank algorithm (Brin and Page, 1998) is defined as follows

$$P_R(i) = \frac{q}{N} + (1-q) \sum_j x_{ji} \frac{P_R(j)}{k_{\mathrm{out},j}}, \tag{8.17}$$

where x_{ij} is the adjacency matrix of the Web graph, $k_{\mathrm{out},j}$ is the out-degree of vertex j, N is the total number of pages of the web graph and q is the so-called *damping factor*. In the context of web surfing, the damping q is a crude modeling of the probability that a random surfer gets bored, stops following links, and proceeds to visit a randomly selected web page. The set of equations (8.17) can be solved iteratively and the stationary P_R can be thought of as the stationary probability of a random walk process with additional random jumps, as modulated by q. If $q = 0$, the PageRank algorithm is a simple diffusion process that just accumulates on the nodes with null out-degree (since the web is directed, random walkers cannot get out of such nodes) or in particular sets of nodes with no links towards the rest of the network (such as the out-component of the network, see Chapter 1). If $q \neq 0$ (applications usually implement a small $q \simeq 0.15$) the stationary process gives the PageRank value of each node.

While we know from the previous section that the visiting probability in an undirected network increases with the degree of the node, in the case of the directed web graph and the PageRank algorithm this fact is not completely intuitive as the P_R visiting probability depends on the global structure of the web and on the hyperlinks between pages. In particular the structure of the equations appears to give some role to the out-degree as well. It is possible, however, to show with a mean-field calculation that in uncorrelated networks the average PageRank value is just related to the in-degree of the nodes. Analogously to what has been shown for regular diffusion, let us define the PageRank for statistically equivalent nodes of degree class $\mathbf{k} = (k_{\mathrm{in}}, k_{\mathrm{out}})$ as

$$P_R(k_{\mathrm{in}}, k_{\mathrm{out}}) = \frac{1}{N_{\mathbf{k}}} \sum_{i\,|k_{\mathrm{in},i}=k_{\mathrm{in}};k_{\mathrm{out},i}=k_{\mathrm{out}}} P_R(i), \tag{8.18}$$

where $N_{\mathbf{k}}$ is the number of nodes with in- and out-degree k_{in} and k_{out}, respectively. In this case Fortunato *et al.* (2005) have shown that in the stationary limit

https://doi.org/10.1017/CBO9780511791383.009 Published online by Cambridge University Press

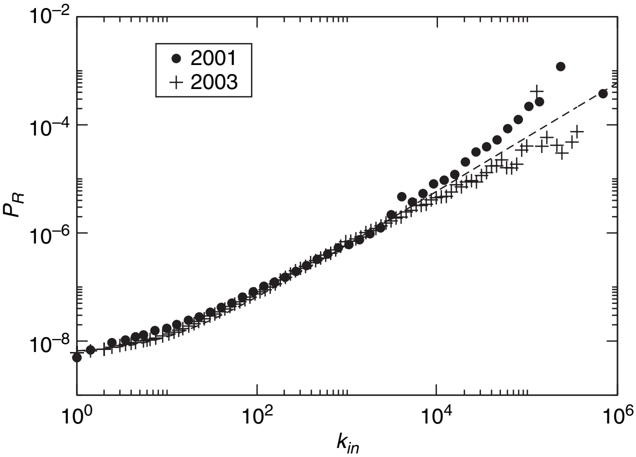

Fig. 8.2. Average PageRank as a function of in-degree for two samples of the Web graph obtained by two large crawls performed in 2001 and 2003 by the WebBase collaboration at Stanford (`http:// dbpubs.stanford.edu:8091/~testbed/doc2/WebBase/`). The 2001 crawl indexes 80, 571, 247 pages and 752, 527, 660 links and the 2003 crawl has 49, 296, 313 pages and 1,185,396,953 links. The PageRank values are obtained with a $q = 0.15$ and averaged over logarithmic bins of in-degree. The dashed line corresponds to a linear fit. Data courtesy of S. Fortunato.

$$P_R(k_{\mathrm{in}}, k_{\mathrm{out}}) = \frac{q}{N} + \frac{(1-q)}{N} \frac{k_{\mathrm{in}}}{\langle k_{\mathrm{in}} \rangle}, \tag{8.19}$$

where $\langle k_{\mathrm{in}} \rangle$ is the average in-degree in the web graph considered. In other words, the average PageRank of nodes just depends on the in-degree of the node. This expression is obtained in a statistical equivalence assumption and for uncorrelated networks, while the real web graph obviously does not satisfy these conditions. However, numerical inspection of the relationship between in-degree and P_R in real web graph samples shows that the linear behavior is a good approximation to the real behavior (see Figure 8.2 and Fortunato *et al.* [2006b]). The PageRank algorithm is therefore a measure of the popularity of nodes that is mostly due to the in-degree dependence of the diffusion process. This is, however, a mean-field result and important fluctuations appear: some nodes can have for example a large PageRank despite a modest in-degree, and one of the refinements of search engines consists in their ability to uncover such outliers. Overall, the striking point is that the effectiveness of the algorithm is due to the heterogeneous properties of the network which induce a discovery probability varying over orders of magnitude and truly discriminating the most popular pages. The self-organized complex structure of the web is therefore a key element in our capability of making use of it.

https://doi.org/10.1017/CBO9780511791383.009 Published online by Cambridge University Press

Interestingly, the use of PageRank has spilled out in other areas where the ranking and retrieval of information in large-scale directed networks is particularly relevant. A recent example is provided by the problem of measuring the impact of a scientific publication. The usual measure given by the simple number of citations can hide papers that appear as important only after some time, or overestimate some other papers because of temporary fads. Network tools prove useful in this context: scientific publications are considered as nodes, and each citation from a paper A to a paper B is drawn as a directed link from A to B. The number of citations of a paper is then given by its in-degree. Detailed analysis of the corresponding directed network for the *Physical Review* journals can be found in Redner (1998; 2005). The retrieval of scientific information often proceeds in this network in a way similar to what occurs on the web. Typically, a scientist starts from a given paper and then follows chains of citation links, discovering other papers. In this context, Chen *et al.* (2007) propose to use the PageRank of a paper, instead of its in-degree, to quantify its impact. The fundamental reasons why such a quantity is better suited are two-fold: (i) it takes into account the fact that being cited by papers that are themselves important is more important, and (ii) for PageRank, being cited by a paper with a small number of references (small out-degree) gives a larger contribution than being cited by a paper which contains hundreds of references (see Equation (8.17)). The analysis of the citation network of the *Physical Review* shows that the PageRank of a paper is correlated with its in-degree, as in the case of the World Wide Web. However, a number of outliers with moderate in-degree and large PageRank are found. For example, old papers having originated a fundamental idea may not be cited anymore because the idea has become folded into a field's common practice or literature, so that these old papers' in-degree remains moderate, but PageRank measures more accurately their impact over the years. Walker *et al.* (2007) explore these ideas further by defining the CiteRank index, which takes into account the fundamental differences between the web and a citation network: first, links between papers cannot be removed or updated; second, the time-arrow implies that papers age. The CiteRank algorithm therefore simulates the dynamics of scientists looking for information by starting from recent papers (the probability of starting from a random paper decays exponentially with its age) before following the directed links. In a different context – but still in the area of academic ranking – Schmidt and Chingos (2007) propose to rank academic doctoral programs based on their records of placing their graduates in faculty positions. In network terms, academic departments link to each other by hiring their faculty from each other (and from themselves) and the PageRank index can be computed on this directed network. Wissner-Gross (2006) also uses a weighting method similar to PageRank to generate customized reading lists based on the link structure of the online encyclopedia Wikipedia (`http://wikipedia.org`).

https://doi.org/10.1017/CBO9780511791383.009 Published online by Cambridge University Press

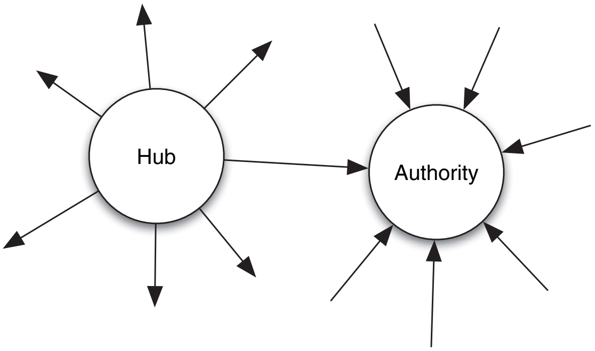

Fig. 8.3. Authorities and hubs in the web graph. Authorities are the most relevant web pages on a certain topic and hubs are pages containing a large number of links to authorities.

Finally, let us mention that alternative methods in the same spirit of PageRank have been proposed in order to improve the ranking of search engines by considering both the in- and the out-degree distribution of the web graph and other directed graphs such as citation and authorship networks (Kleinberg, 1998). This method relies on the distinction between *authorities* and *hubs* (see Figure 8.3). Authorities are web pages that can be considered the most relevant source of information about a given topic. Given the large amount of knowledge that these kinds of pages encode, it is natural to assume that they have a large number of incoming links. Hubs, on the other hand, are pages dealing with a given topic, which are not authorities themselves but which contain a large number of outgoing links pointing to related authorities. In this situation, the set of hubs and authorities on a topic form a bipartite clique, in which all hubs point to all authorities. Therefore, by focusing on the detection of bipartite cliques, it should be possible to identify which are those authorities and rank them in the highest position. Following this approach, Kleinberg (1998) has proposed the Hyperlink-Induced Topic Search (HITS) algorithm, which has been the seed for several variations and improvements (Marendy, 2001).

8.3 Searching strategies in complex networks

The problem of searching in complex networks is a very important and practical one which arises in many contexts. Complex networks can often be seen as large reservoirs of information, which can be passed from one part of the network to another thanks to the existence of paths between nodes (see Chapter 1). As described in the previous section, the most obvious example is given by the World Wide Web, which is nowadays one of the most important sources of information

https://doi.org/10.1017/CBO9780511791383.009 Published online by Cambridge University Press

used on a daily basis. Peer-to-Peer (P2P) networks also allow for the retrieval of desired files, while the Internet is the support of such virtual networks and makes it possible to transmit messages in electronic form. The exploration of a network has therefore the typical goal of retrieving a particular item of information, of transmitting messages to a node (be it a computer or an individual), or of finding in a social network the right person to perform a given task.

In this respect, it is very important to consider the experiment elaborated by Milgram (1967) which has been largely celebrated as putting on solid ground the familiar concept of the small world (see also Chapter 2). Randomly selected individuals in the Midwest were asked to send a letter to a target person living in Boston. The participants knew the name and occupation of the target but were only allowed to pass the letter to a person they knew on a first-name basis. Strikingly, the letters that successfully reached the target had been passed only a small number of times, traveling on average through a chain of six intermediate hops. This experiment was duplicated and repeated in various environments, with similar results (Travers and Milgram, 1969; Korte and Milgram, 1970; Lundberg, 1975; Bochner, Buker and McLeod, 1976). Very recently, it was also performed and analyzed by Dodds, Muhamad and Watts (2003) in a more modern set-up: the participants were asked to use email instead of regular mail. Besides the small-world effect, another important and puzzling conclusion can be drawn from the outcome of the experiments of Milgram (1967) and Dodds *et al.* (2003), with more far-reaching implications: the existing short paths *can be found without global knowledge of the social network*. People receiving the message had to forward it to one of their neighbors in the social network, but were not given any information on the structure of the network (which indeed remains largely unknown). The standard algorithms for finding the shortest paths between pairs of nodes involve the exploration of the whole network. On the contrary, the participants in the experiments used only their local knowledge consisting of the identity and geographical location of their neighbors, so that the search for the final target was decentralized. Even if there is no guarantee that the true shortest paths were discovered during the experiment, it is quite striking that very short paths were found at all.

Obviously, the task of searching for a specific node or a specific piece of information will be more or less difficult depending on the amount of information available to each node. The most favorable case arises when the whole structure of nodes and edges is known to all vertices. The straightforward strategy is then to take advantage of this global knowledge to follow one of the shortest paths between the starting node and the target vertex. Clearly, such a case of extensive knowledge is seldom encountered and other strategies have to be used, as we will review in the following (see also Kleinberg [2006] for a recent review on this topic).

https://doi.org/10.1017/CBO9780511791383.009 Published online by Cambridge University Press

8.3.1 Search strategies

The task of searching can correspond to various situations: finding a particular node (possibly given some information on its geographical position, as in Milgram's experiment) in order to transmit a message, or finding particular information without knowing a priori in which node it is stored (as is the case in Peer-to-Peer networks). In all cases, search strategies or algorithms will typically consist of message-passing procedures: each node, starting from the initial node, will transmit one or more messages to one or more neighbors on the network in order to find a certain target. This iterative process stops when the desired information is found and sent to the source of the request.

As already mentioned, when each vertex has exhaustive information about the other vertices and the connectivity structure of the network, messages can be passed along the shortest path of length ℓ_{st} from the source s to the target t. The delivery time T_N and number of messages exchanged (i.e. the traffic involved) are also given by ℓ_{st} which, in most real-world complex networks, scales on average only as $\log N$ (or slower) with the size N of the network (Chapter 2). Such low times and traffic are, however, obtained at the expense of storing in each node a potentially enormous amount of information: for example, in the case of the WWW, each server should keep the address and content of all existing web pages, a clearly impossible configuration. Another concern regards nodes with large betweenness centrality, through which many shortest paths transit, and which will therefore receive high traffic if many different queries are sent simultaneously. We will see in Chapter 11 how congestion phenomena can then arise and how various routing strategies can alleviate such problems.

In the opposite case, when no information is available on the position of the requested information, nor on the network's structure, the simplest strategy is the so-called *broadcast search* (see Figure 8.4): the source node sends a message to *all* its neighbors. If none of them has the requested information (or file), they iterate this process by forwarding the message to all their own neighbors (except the source node). This process is repeated, flooding the network until the information is retrieved. In order to avoid excessive traffic and a possibly never-ending process, a corresponding time-to-live (TTL) is assigned from the start to the message. At each transmission the TTL is decreased by one until it reaches 0 and the message is no longer transmitted. When the target vertex is reached by the message, it sends back to the source the requested item. Since the other nodes cannot be easily informed of the success of the query, the broadcast process continues in other parts of the network. The broadcast algorithm, akin to the breadth-first algorithm commonly used to find shortest paths, proceeds in parallel and is clearly able to

https://doi.org/10.1017/CBO9780511791383.009 Published online by Cambridge University Press

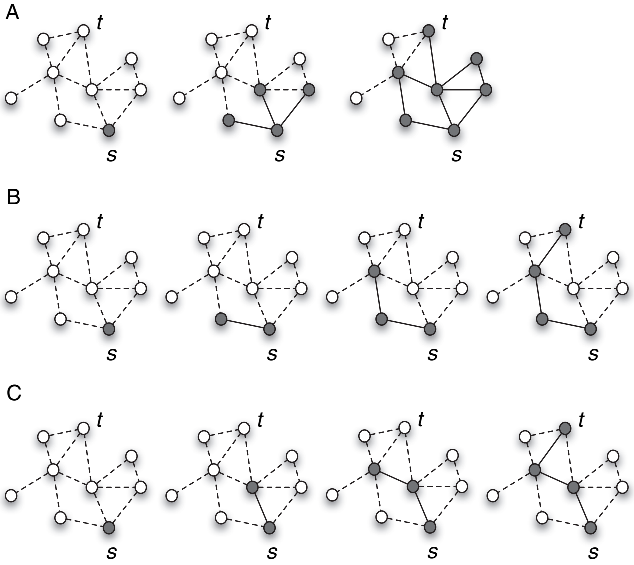

Fig. 8.4. Schematic comparison of various searching strategies to find the target vertex t, starting from the source s. A, Broadcast search; B, Random walk; C, Degree-biased strategy. The broadcast search finds the shortest path, at the expense of high traffic.

explore the entire network very rapidly. In particular, the desired target is reached after a number of steps equal to the shortest path from the source, and the delivery time is therefore equal to that obtained with full knowledge of the network. The obvious drawback consists in the large amount of traffic generated, since at each request all nodes within a shortest path distance ℓ of the source are visited, where ℓ is given by the TTL. The number of such nodes typically grows exponentially with ℓ and a large fraction of (or the whole) network receives a message at each search process. The traffic thus grows linearly with the size N of the network. Refined approaches have been put forward, in particular with the aim of obtaining efficient Peer-to-Peer search methods, such as the replication of information at various nodes, or iterative deepening, which consists of starting with a small TTL and increasing it by one unit at a time only if the search is not successful (see, for example, Yang and Garcia-Molina [2002]; Lv *et al.* [2002]). The amount of traffic can then be sensibly reduced but will still remain high.

Intermediate situations are generally found, in which each node possesses a limited knowledge about the network, such as the information stored in each of its neighbors. A straightforward and economical strategy is then given by the *random walk search*, illustrated in Figure 8.4. In this case, the source node starts by checking if any of its neighbors has the requested information. If not, it sends a

https://doi.org/10.1017/CBO9780511791383.009 Published online by Cambridge University Press

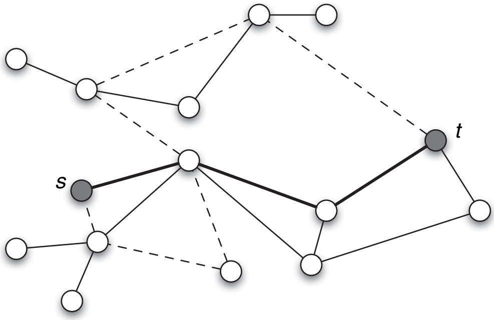

Fig. 8.5. Difference between the shortest path between nodes s and t (continuous thick line), of length 3, and an instance of a random walk (dashed line), here of length 6.

message to one randomly chosen neighbor,[2] which iterates the process (at each step, the node holding the message does not send it back to the node from which it was received), until a neighbor of the target is reached: the message is then passed directly to the target. Clearly, the delivery time T_N will be larger than for a broadcast strategy, since the random walk does not follow the shortest path from the source to the target (Figure 8.5). In fact, the random walk tends to visit the same nodes several times, and the probability of following exactly the shortest path is very low (Sneppen, Trusina and Rosvall, 2005; Rosvall, Minnhagen and Sneppen, 2005; Rosvall *et al.*, 2005). In generalized random networks with a scale-free degree distribution $P(k) \propto k^{-\gamma}$, with exponent $\gamma = 2.1$ corresponding to the value of Peer-to-Peer networks, Adamic *et al.* (2001) obtain for the random walk search $T_N \sim N^{0.79}$. This power-law behavior denotes a much worse behavior than the broadcast strategy in terms of delivery time. The traffic generated, however, is equal to T_N (since only one walker is generated at each request), and therefore remains smaller than the linear growth of the broadcast search.

We will see in the next sections how various strategies can be devised, depending on the network's structure and on the various levels of (local) knowledge available to the nodes.

8.3.2 Search in a small world

As illustrated by the Watts–Strogatz model (see Chapter 3), small-world networks are characterized by "shortcuts" which act as bridges linking together "far away" parts of the network and allow for a strong decrease of the network's diameter.

[2] The efficiency of the search process can be increased by the use of several random walkers in parallel, although this generates additional costs in terms of traffic (Lv *et al.*, 2002).

https://doi.org/10.1017/CBO9780511791383.009 Published online by Cambridge University Press

The concept of "far away" here refers to an underlying space (e.g. geographical) in which the nodes are located, as in Milgram's original experiment. In such cases, it is natural to try to use these shortcuts for the searching process. Such an approach is not as easy as it might seem: for example, in the Watts–Strogatz model, shortcuts are totally random and of arbitrary length, so that it is difficult to select the correct shortcut. Such bridges must therefore encode some information about the underlying structure in order to be used. Kleinberg (2000a) has formalized the conditions under which a Watts–Strogatz-like small-world network can be efficiently navigated (see also Kleinberg [2000b]). In his model, shortcuts are added to a D-dimensional hypercubic lattice in the following way: a link is added to node i, and the probability for this link to connect i to a vertex j at geographical distance r_{ij} is proportional to $r_{ij}^{-\alpha}$, where α is a parameter. Each node knows its own position and the geographical location of all its neighbors. The search process used is the simplest greedy one: a message has to be sent to a certain target node t whose geographical position is known. A node i receiving the message forwards it to the neighbor node j (either on the lattice or using the long-range link) that is geographically closest to the target, i.e., which minimizes r_{jt}. The idea is thus to send the message at each step closer and closer to the target. Kleinberg (2000a) shows a very interesting result: if $\alpha = D$, the delivery time scales as $\log^2(N)$ with the size N of the network. Various generalizations of this result can be found in the literature. For example, modified versions of the algorithm consider that nodes may consult some nearby nodes before choosing to which neighbor it forwards the message (see Lebhar and Schabanel [2004]; Manku, Naor and Wieder [2004] and Kleinberg [2006] for a review). On the contrary, as soon as the exponent α deviates from the space dimension D, the delivery time increases as a power of N, i.e., much faster (Figure 8.6).

The intuitive interpretation of this result is as follows: when α is too large, shortcuts connect nodes that are geographically not very distant, and therefore do not

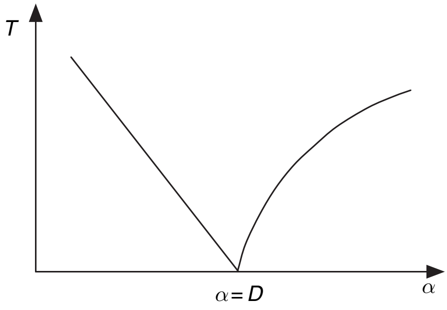

Fig. 8.6. Schematic evolution of the delivery time as a function of α in Kleinberg's model of a small-world network. Figure adapted from Kleinberg (2000a).

https://doi.org/10.1017/CBO9780511791383.009 Published online by Cambridge University Press

shorten efficiently the distances. When α is too small on the other hand, long-range shortcuts are available, but they are typically very long and are not useful to find a specific target. Only at $\alpha = D$ does one obtain shortcuts at all length scales so that the target can be found efficiently through the greedy algorithm.

The situation depicted by Kleinberg's result makes Milgram's experiment even more striking and intriguing. It means indeed that only very particular small-world networks can be searched in short times: the ones with the adequate distribution of shortcuts. A first objection that can come to mind is that we do not live on a hypercubic lattice and that the local population density is strongly heterogeneous. In order to take this aspect into account, Liben-Nowell *et al.* (2005) propose a rank-based friendship linking model, in which, for each node (individual) i, the geographical neighbors are ranked according to their distance to i, and a long-range link is established between i and j with probability inversely proportional to j's rank. The real length of such links will therefore depend on the local density, being shorter in more densely populated areas. Such a generalization of Kleinberg's model, which is in fact realized in the online community called Live-Journal (Liben-Nowell *et al.*, 2005), turns out to support the greedy search algorithm efficiently. A particular linking rule (using the inverse of the rank) is needed, however, just as a particular value of the exponent α is crucial in Kleinberg's result.

A very important point to understand how searching effectively works in social networks is that several distances may coexist, such as geographical but also based on social criteria or affinity: the creation of a link between two individuals may occur not only as a function of geographical distance (e.g. because two persons live in the same neighborhood), but also because they are members of the same group, because they share the same interests or hobby, or because they have the same profession.[3] Interestingly, Killworth and Bernard (1978) have shown that, in Milgram's experiment, the choice of the neighbor to whom the message was forwarded was made mostly on the basis of two criteria: geography and occupation. The participants were therefore taking advantage of the potential proximity of their contacts to the target in both geographical and social space. While geography can be represented by an embedding in a D-dimensional space ($D = 2$), a natural approximation of the space of occupations is a hierarchical structure. Following this idea, Watts, Dodds and Newman (2002) put forward a model in which each node exists at the same time in different hierarchies, each hierarchy corresponding to some dimension of social space. In each hierarchy, a network is constructed, favoring links between nearby nodes, and the union of these networks forms the global social network. The effective distance between nodes is then the

[3] Menczer (2002) also generalizes the idea of an underlying distance to the WWW, by using a semantic or lexical distance between web pages.

https://doi.org/10.1017/CBO9780511791383.009 Published online by Cambridge University Press

minimum possible distance over all the hierarchies. Numerical simulations reveal that the greedy algorithm is efficient if there exists a small number of different ways to measure proximity, and if the network's creation favors homophily, i.e. links exist in each social dimension preferentially among close nodes (see also Kleinberg [2001] for a rigorous approach on similar concepts of group membership). The more recent study of Adamic and Adar (2005) on a real social network (obtained through the email network of a firm) has also revealed that using the hierarchical structure increases the search efficiency (see also Şimşek and Jensen [2005]). Finally, Boguñá, Krioukov and claffy (2007) consider networks in which nodes reside in a hidden metric space, and study the greedy routing in this hidden space. They show that scale-free networks with small exponents for the degree distribution and large clustering are then highly navigable. Such approaches allow us to rationalize the apparent contrast between the success of Milgram's experiment and the restrictive result of Kleinberg.

8.3.3 Taking advantage of complexity

In Milgram's experiment, the identity and location of the target node were known. Such information is not always available, for example in networks without geographical or hierarchical structure. Moreover, when searching for a precise item, one may not even know the identity of the target node. In P2P applications for instance, requests for a specific file are sent without knowing which peers may hold it. In such cases, the greedy algorithm previously described cannot be applied, but the idea of using the shortcuts can still be exploited. In particular, the small-world character of many real-world networks is due to the presence of hubs that connect together many different parts of the network. While the probability of reaching a given target by a random walk is reduced when going through a hub, because of the large choice for the direction of the next step (Rosvall *et al.*, 2005), the fact that typically many shortest paths go through hubs makes them easily accessible (Dall'Asta *et al.*, 2005; Sneppen *et al.*, 2005). Search processes can thus take advantage of this property with a strategy that biases the routing of messages towards the nodes with large degree. This *degree-biased* searching approach (see Figure 8.4C) has been studied by Adamic *et al.* (2001) and Kim *et al.* (2002b), assuming that each node has information about the identity and contents of its neighbors, and on their degree. Using this purely local information, a node holding the message (search request) forwards it either to the target node if it is found among its neighbors, or to its most connected neighbor (see also Adamic, Lukose and Huberman [2003]). Numerical simulations allow comparing the efficiency of this method with the random-walk search. For example, on generalized random networks with a scale-free degree distribution $P(k) \sim k^{-\gamma}$ with $\gamma = 2.1$, Adamic

https://doi.org/10.1017/CBO9780511791383.009 Published online by Cambridge University Press

et al. (2001) obtain a delivery time $T_N \sim N^{0.7}$ smaller than for the random walk. Such increased performance is confirmed by analytical calculations and can be understood by the following intuitive argument: if each node has a knowledge of the information stored in all its neighbors, hubs will naturally have more knowledge than low degree nodes, and it seems therefore a sensible choice to send them the request. Starting from a random node on the network, a message which follows the degree-biased strategy will clearly reach the most-connected vertex after a few steps. From this position, it will start in fact to explore the network by visiting the nodes in decreasing order of their degree. Interestingly, the deterministic character of the rule turns out to be crucial, and any amount of randomness (such as sending the message to a neighbor chosen with a probability proportional to its degree) yields much larger delivery times (Kim *et al.*, 2002b). Moreover, the strategy turns out to be quite inefficient in homogeneous networks such as Erdős–Rényi or Watts–Strogatz models where no hubs are present (Adamic *et al.*, 2001; Kim *et al.*, 2002b; Adamic *et al.*, 2003; Adamic and Adar, 2005). The clear drawback of the degree-biased strategy lies in the potentially high traffic imposed on the hubs (see Chapter 11 for a detailed discussion on traffic issues).

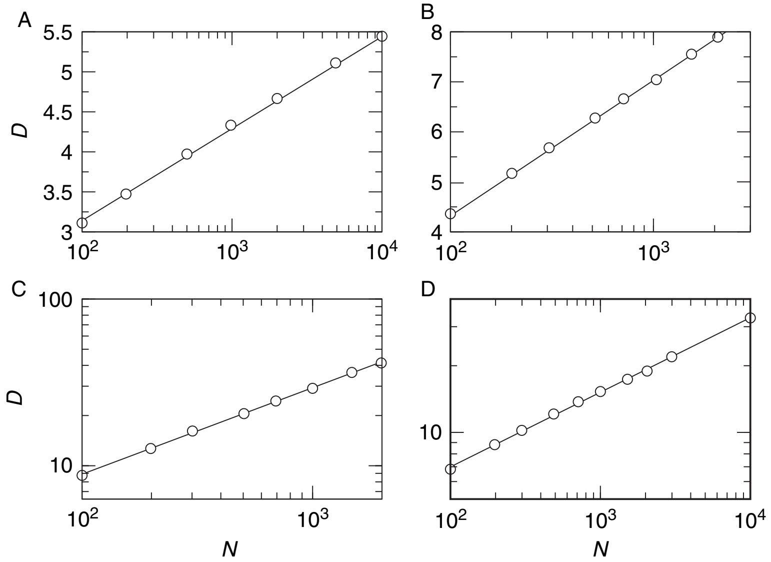

Fig. 8.7. Network effective diameter D as a function of size in a Barabási–Albert network for (A) shortest paths, (B) deterministic degree-biased strategy, (C) random walk strategy and (D) preferential degree-biased strategy (neighbor chosen with probability proportional to its degree). D scales logarithmically for A and B, and as a power-law for C and D. Data from Kim *et al.* (2002b).

https://doi.org/10.1017/CBO9780511791383.009 Published online by Cambridge University Press

Search strategies may also be used to find (short) paths between nodes, that can be subsequently re-used in future communications (Kim *et al.*, 2002b; Adamic *et al.*, 2003). With respect to the outcome of the search strategy, loops and possible backward steps are removed, yielding a path between the source and the target which is shorter than the real path followed by the message. The average path length obtained by selecting random source and target nodes defines an effective diameter of the network which depends on the search strategy. Numerical simulations on Barabási–Albert networks show that this effective diameter scales in very different ways with the network size N for the various strategies, from logarithmically for the degree-biased case to power laws for random walks (see Figure 8.7). Interestingly, the degree-biased strategy, even if it yields delivery times scaling as a power of the size, allows very short paths to be found once the loops have been taken out.

Various studies have recently aimed at improving the simple degree-biased search strategy. Thadakamalla, Albert and Kumara (2005) in particular consider the case of weighted networks in which a cost may be associated to the transmission of a message through an edge (see also Jeong and Berman [2007]). A *local betweenness centrality* can then be defined as the number of shortest (weighted) paths in the subnetwork formed by the neighbors and the neighbors' neighbors of a node (see also Trusina, Rosvall and Sneppen [2005]). Sending the message to the neighbor with largest local betweenness centrality then proves to be quite efficient, since it connects in an effective way information on the degree and weights. In general, complex networks are searchable thanks to their degree heterogeneity, and the combination of information on the degrees of neighboring node with knowledge of other attributes such as geography (Thadakamalla, Albert and Kumara, 2007), weights (Thadakamalla *et al.*, 2005; Jeong and Berman, 2007), social attributes (Şimşek and Jensen, 2005), information on the shortest path to the nearest hub (Carmi, Cohen and Dolev, 2006), or even the use of adaptative schemes for P2P applications (Yang and Garcia-Molina, 2002), allow the building of search algorithms that perform rather efficiently.

https://doi.org/10.1017/CBO9780511791383.009 Published online by Cambridge University Press